Masters of Ceremony & Editors: Polo Chau Jilles Vreeken Matthijs van Leeuwen Dafna Shahaf Christos Faloutsos

Proceedings of the
ACM SIGKDD 2016 Full-day Workshop on

Interactive Data Exploration and Analytics

IDEA 2016

San Francisco, CA, USA
August 14, 2016
poloclub.gatech.edu/idea2016

Proceedings of the
ACM SIGKDD Workshop on
Interactive Data Exploration and Analytics

These proceedings are not included in the ACM Digital Library.

IDEA'16, August 14, 2016, San Francisco, California, USA.

ACM SIGKDD Workshop on Interactive Data Exploration and Analytics

General Chairs

Duen Horng (Polo) Chau (Georgia Tech)
Jilles Vreeken (Max Planck Institute for Informatics and Saarland University)
Matthijs van Leeuwen (Universiteit Leiden)
Dafna Shahaf (Hebrew University of Jerusalem)
Christos Faloutsos (Carnegie Mellon University)

Program Committee

Acar Tamersoy (Symantec & Georgia Tech, USA)
Alex Endert (Georgia Tech, USA)
Antti Oulasvirta (Aalto U., Finland)
Antti Ukkonen (Finnish Institute of Occupational Health, Finland)
Bahador Saket (Georgia Tech, USA)
Danai Koutra (U. Michigan, USA)
Edith Law (U. Waterloo, Canada)
Esther Galbrun (Loria, France)
Geoff Webb (Monash U., Australia)
Hanghang Tong (Arizona State University, USA)
Hannah Kim (Georgia Tech, USA)
Jaakko Hollmen (Aalto U., Finland)
Jaegul Choo (Georgia Tech)
James Abello (Rutgers, USA)
Jefrey Lijffijt (U. Ghent, Belgium)
Kai Puolamäki (Finnish Institute of Occupational Health, Finland)
Kevin Roundy (Symantec, USA)
Mario Boley (MPI-INF, MMCI, UdS, FHI Berlin & Frauenhofer IAIS, Germany)
Marti Hearst (UC Berkeley, USA)
Michael Berthold (U. Konstanz, Germany)
Minsuk (Brian) Kahng (Georgia Tech, USA)
Nan Cao (NYU Tandon School of Engineering, USA)
Nikolaj Tatti (Aalto University, Finland)
Pauli Mietinnen (Max Planck Institute for Informatics, Germany)
Saleema Amershi (Microsoft Research, USA)
Siegfried Nijssen (Leiden U., the Netherlands)
Stefan Kramer (U. Mainz, Germany)
Steffen Koch (U. Stuttgart, Germany)
Sucheta Soundarajam (Syracuse U., USA)
Thomas Gärtner (U. Nottingham, UK)
Thomas Seidl (LMU Munich, Germany)
Tijl De Bie (University of Bristol, UK)
Tim (Jia-Yu) Pan (Google, USA)
U Kang (Seoul National U., South Korea)
Wouter Duivesteijn (U. Ghent, Belgium)
Zhicheng 'Leo' Liu (Adobe Research, USA)

Preface

Data, data everywhere; massive datasets of previously unthinkable sizes, surpassing terabytes and petabytes, have quickly become commonplace. They arise in numerous settings in science, government, and enterprises. While technology exists by which we can collect and store such massive amounts of information, making sense of these data remains a fundamental challenge. In particular, we lack the means to explanatorily analyze databases of this scale. Currently, surprisingly few technologies allow us to freely "wander" around the data, and make discoveries by following our intuition, or serendipity. While standard data mining aims at finding highly interesting results, it is typically computationally demanding and time consuming, thus may not be well-suited for interactive exploration of large datasets.

Interactive data mining techniques that aptly integrate human intuition, by means of visualization and intuitive **human-computer interaction** techniques, and **machine computation** support have been shown to help people gain significant insights into a wide range of problems. However, as datasets are being generated in larger volumes, higher velocity, and greater variety, creating effective interactive data mining techniques becomes an increasingly harder task.

It is exactly this research, experiences and practices that we aim to discuss at IDEA, the workshop on Interactive Data Exploration and Analytics. In a nutshell, IDEA addresses the development of data mining techniques that allow users to interactively explore their data. We focus and emphasize on **interactivity** and effective **integration** of techniques from **data mining**, **visualization** and **human-computer interaction**. In other words, we explore how the best of these different but related domains can be combined such that the *sum is greater than the parts*.

Following the great success of IDEA at KDD 2013, 2014, and 2015, the main program of IDEA'16 consists of seventeen papers that cover various aspects of interactive data exploration and analytics. In addition there was one invited demonstration, and four keynotes. Seven papers were presented orally, and ten were presented during the interactive poster and demo session. These papers were selected from a total of 29 submissions after a thorough reviewing process. We sincerely thank the authors of the submissions and the attendees of the workshop. We wish to thank the members of our program committee for their help in selecting a set of high-quality papers. Furthermore, we are very grateful to Jerome H. Friedman, Jeffrey Heer, Eamonn Keogh, and Saleema Amershi for engaging keynote presentations on the fundamental aspects of interactive data exploration, analysis, and visualization.

Polo Chau & Jilles Vreeken & Matthijs van Leeuwen & Dafna Shahaf & Christos Faloutsos

Saarbrücken, July 2016

Table of Contents

Invited Talks

Regression Location and Scale Estimation with Application to Censoring
Jerome H. Friedman 8

Predictive Interaction
Jeffrey Heer 9

At Last! Time Series Joins, Motifs, Discords and Shapelets at Interactive Speeds
Eamonn Keogh 10

Towards Usable Machine Learning
Saleema Amershi 11

Research Papers

Expressive Query Construction through Direct Manipulation of Nested Relational Results
Eirik Bakke & David Karger 12

On the Intuitiveness of Common Discretization Methods
Mario Boley & Ankit Kariryaa 22

ReVACNN: Real-Time Visual Analytics for Convolutional Neural Network
Sunghyo Chung & Sangho Suh & Cheonbok Park & Kyeongpil Kang & Jaegul Choo & Bum Chul Kwon 30

Clustrophile: A Tool for Visual Clustering Analysis
Çağatay Demiralp 37

A Visual Approach for Interactive Co-Training
Qi Han & Weimeng Zhu & Florian Heimerl & Steffen Koch & Thomas Ertl 46

Visual Quality Assessment of Subspace Clusterings
Michael Hund & Ines Färber & Michael Behrisch & Andrada Tatu & Tobias Schreck & Daniel A. Keim & Thomas Seidl 53

Interacting with Predictions:
Visual Inspection of Black-box Machine Learning Models
Josua Krause & Adam Perer & Kenney Ng 63

Interactive Exploration for Domain Discovery on the Web
Yamuna Krishnamurthy & Kein Pham & Aécio Santos & Juliana Freire 64

Peekquence: Visual Analytics for Event Sequence Data
Bum Chul Kwon & Janu Verma & Adam Perer. 72

Human-guided Flood Mapping on Satellite Images
Jiongqian Liang & Peter Jacobs & Srinivasan Parthasararthy 76

SIDE: A Web App for Interactive Visual Data Exploration with Subjective Feedback
Jefrey Lijffijt & Bo Kang & Kai Puolamäki & Tijl De Bie 86

Interactive Constrained Boolean Matrix Factorization
Nelson Mukuze & Pauli Miettinen 96

"Why Should I Trust You?" Explaining the Predictions of Any Classifier
Marco Tulio Ribeiro & Sameer Sing & Carlos Guestrin 105

Direct-Manipulation Visualization of Deep Networks
Daniel Smilkov & Shan Carter & D. Sculley & Fernanda Viegas & Martin Watternberg . . 115

Clustering with a Reject Option: Interactive Clustering as Bayesian Prior Elicitation
Akash Srivastava & James Zou & Charles Sutton 120

Interacting with Massive Behavioral Data
Shih-Chieh Su 127

VIT-PLA: Visual Interactive Tool for Process Log Analysis
Sen Yang & Xin Dong & Moliang Zhou & Xinyu Li & Shuhong Chen & Rachel Webman & Aleksandra Sarcevic & Ivan Marsic & Randall Burd 130

Invited Talk

Regression Location and Scale Estimation with Application to Censoring

Jerome H. Friedman
Department of Statistics
Stanford University
jhf@stanford.edu

Abstract

The aim of regression analysis in machine learning is to estimate the location of the distribution of an outcome variable y, given the joint values of a set of predictor variables x. This location estimate is then used as a prediction for the value of y at x. The accuracy of this prediction depends on the scale of the distribution of y at x, which in turn, usually depends on x (heteroscedasticity). A robust procedure is presented for jointly estimating both the location and scale of the distribution of y given x, as functions of x, under no assumptions concerning the relationship between the two functions. The scale function can then be used to access the accuracy of individual predictions, as well as to improve accuracy especially in the presence of censoring.

Bio

Jerome H. Friedman is Professor Emeritus of Statistics, Stanford University. He received both bachelor's and Ph. D degrees in physics from the University of California, Berkeley. He was leader of the Computation Research Group at the Stanford Linear Accelerator Center from 1972 through 2006. He was Professor of Statistics, Stanford University, from 1982 through 2006, and served as Department Chair from 1988 through 1991. His primary interests center on machine learning and data mining. He has authored or coauthored over 100 papers in major statistical journals as well as three books on Data Mining, and has invented or co-invented several widely used machine learning and data mining procedures.

Invited Talk

Predictive Interaction

Jeffrey Heer
Department of Computer Science and Engineering
University of Washington
jheer@uw.edu

Abstract

How might we architect interactive systems that have better models of the tasks we're trying to perform, learn over time, help refine ambiguous user intents, and scale to large or repetitive workloads? In this talk I will present Predictive Interaction, a framework for interactive systems that shifts some of the burden of specification from users to algorithms, while preserving human guidance and expressive power. The central idea is to imbue software with domain-specific models of user tasks, which in turn power predictive methods to suggest a variety of possible actions. I will illustrate these concepts with examples drawn from widely-deployed systems for data transformation and visualization (with reported order-of-magnitude productivity gains) and then discuss associated design considerations and future research directions.

Bio

Jeffrey Heer is an Associate Professor of Computer Science & Engineering at the University of Washington, where he directs the Interactive Data Lab and conducts research on data visualization, human-computer interaction and social computing. The visualization tools developed by his lab (D3.js, Vega, Protovis, Prefuse) are used by researchers, companies and thousands of data enthusiasts around the world. His group's research papers have received awards at the premier venues in Human-Computer Interaction and Information Visualization (ACM CHI, ACM UIST, IEEE InfoVis, IEEE VAST, EuroVis). Other awards include MIT Technology Review's TR35 (2009), a Sloan Foundation Research Fellowship (2012), and a Moore Foundation Data-Driven Discovery Investigator award (2014). Jeff holds BS, MS and PhD degrees in Computer Science from UC Berkeley, whom he then betrayed to go teach at Stanford from 2009 to 2013. Jeff is also a co-founder of Trifacta, a provider of interactive tools for scalable data transformation.

Invited Talk

At Last! Time Series Joins, Motifs, Discords and Shapelets at Interactive Speeds

Eamonn Keogh
Computer Science and Engineering Department
University of California - Riverside
eamonn@cs.ucr.edu

Abstract

Given the ubiquity of time series, the last decade has seen a flurry of activity in time series data mining. Some of the most useful and frequently used primitives "reason" about the shapes of subsequences found in longer time series. Examples include Time Series Joins, Motifs, Discords and Shapelets. These primitives have found significant adoption, however they are all run in batch mode. For most non-trivial datasets, you start the process; you go to lunch (or on a short vacation!) and examine the results when you get back. What if you could solve such problems in interactive time? Well, now you can! With a new data structure call the Matrix Profile, interactive data mining of large datasets has become possible for the first time, and as we shall demonstrate, it is a game changer.

Bio

Eamonn Keogh is a Full Professor at the Computer Science & Engineering Department of University of California - Riverside. His research areas include data mining, machine learning and information retrieval, specializing in techniques for solving similarity and indexing problems in time-series datasets. He has authored more than 120 papers. He received the IEEE ICDM 2007 best paper award, SIGMOD 2001 best paper award, and runner up best paper award in KDD 1997. He has given over two dozen well received tutorials in the premier conferences in data mining and databases.

Invited Talk

Towards Usable Machine Learning

Saleema Amershi
Machine Teaching Group
Microsoft Research
samershi@microsoft.com

Abstract

It is widely believed that machine learning based applications will broadly impact human productivity and decision making. However, experts caution that the complexity of machine learning can also lead to unintended consequences ranging from irrelevant content recommendations to damaging financial and political decisions made on the bases of inaccurate predictions. While advances continue in improving the accuracy and efficiency of machine learning algorithms, a complementary strategy is needed in improving the capabilities of the people who are fundamentally involved in using and building these machine learning based systems. In this talk, I will present tools and techniques we have been developing in the Machine Teaching Group at Microsoft Research to support the people involved in the machine learning process. I will then discuss some of the open challenges and opportunities in working towards more usable machine learning.

Bio

Saleema Amershi is a Researcher in the Machine Teaching Group at Microsoft Research. Her research lies at the intersection of human-computer interaction and machine learning. In particular, her work involves designing and developing tools to support both end-user and practitioner interaction with interactive machine learning systems. Saleema received her Ph.D. in computer science from the University of Washington's Computer Science & Engineering department in 2012.

Expressive Query Construction through Direct Manipulation of Nested Relational Results

Eirik Bakke
MIT CSAIL
ebakke@csail.mit.edu

David R. Karger
MIT CSAIL
karger@csail.mit.edu

ABSTRACT

Despite extensive research on visual query systems, the standard way to interact with relational databases remains to be through SQL queries and tailored form interfaces. We consider three requirements to be essential to a successful alternative: (1) query specification through direct manipulation of results, (2) the ability to view and modify any part of the current query without departing from the direct manipulation interface, and (3) SQL-like expressiveness. This paper presents the first visual query system to meet all three requirements in a single design. By directly manipulating nested relational results, and using spreadsheet idioms such as formulas and filters, the user can express a relationally complete set of query operators plus calculation, aggregation, outer joins, sorting, and nesting, while always remaining able to track and modify the state of the complete query. Our prototype gives the user an experience of responsive, incremental query building while pushing all actual query processing to the database layer. We evaluate our system with formative and controlled user studies on 28 spreadsheet users; the controlled study shows our system significantly outperforming Microsoft Access on the System Usability Scale.

This work was previously published at SIGMOD 2016.

1. INTRODUCTION

Four decades after Query by Example [51], the broad problem of Making Database Systems Usable [25] remains open. Technical users still interact with relational data through hand-coded SQL, while non-technical users rely on restrictive form- and report-based interfaces tailored, at great cost, for their specific database schema [32, 27, 4]. Queries that involve "complex aggregates, nesting, correlation, and several other features remain on a tall pedestal approachable only by the initiated" [23]. Simple report queries traversing one-to-many relationships in the database schema, such as retrieving "a list of parts, and for each part a list of suppliers and a list of open orders", are painful to define for programmers and largely inaccessible to end users.

Meanwhile, users from a wide range of backgrounds seem happy, indeed eager, to interact with their data if it is served to them in spreadsheet form. "Export to Excel", the joke goes, "is the third most common button in data and business intelligence apps... after OK and Cancel"[1]. Spreadsheets lack basic database functionality such as joins and views, but demonstrate the great value of usable, general-purpose data manipulation tools [4].

Shneiderman [43] attributes the usability of the spreadsheet to its nature as a *direct manipulation* interface. The properties of such an interface include "visibility of the object of interest", "rapid, reversible, incremental actions", and "replacement of complex command language syntax by direct manipulation of the object of interest". Shneiderman paraphrases Harold Thimbleby: "The display should indicate a complete image of what the current status is, what errors have occurred, and what actions are appropriate."

We agree with Liu and Jagadish [35] that a successful solution to the visual query language problem must come in the form of a spreadsheet-like direct manipulation interface. In particular, we consider three requirements that have yet to be met in a single user interface design:

R1. *Query specification through direct manipulation of results.* The user should build queries incrementally through a sequence of operations performed directly on the data in the database, as seen through the result of each intermediate query [35]. In Shneiderman's terms, the *object of interest* is not the query, but the data, as when working with a spreadsheet.

R2. *The ability to view and modify any part of the current query, including operations performed many steps earlier, without redoing subsequent steps or departing from the direct manipulation interface.* This is tricky in light of R1, because the user will be looking at and manipulating the *result* of a query rather than an actual query expression. The mapping between the two is not obvious. [35]

R3. *SQL-like expressiveness from within the direct manipulation interface.* R1 and R2 can be trivially met if only simple queries are allowed. For example, Excel's *filter* feature works by direct manipulation of results, and allows its complete state to be viewed and modified from within the same interface, but supports only basic selection queries. To compete with SQL, a visual query system should allow the user to express any query commonly supported by SQL implementations, including arbitrary (multi-block) combinations of operations such as joins, calculations, and aggregations.

In this paper, we present SIEUFERD (pronounced *soy-fird*), the first visual query system to meet all of the requirements above in

Short version prepared for the 2016 KDD IDEA Workshop.
To cite, please refer to our full SIGMOD paper [5].

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16). August 14th, 2016, San Francisco, CA, USA.

[1] http://www.powerpivotpro.com/2012/03/the-3rd-most-common-button-in-data-apps-is

Direct Manip.	*Query Representation*	*Year*	*System*		*R1*	*R2*	*R3*	*Unrestricted Nested Results*
Yes	Overlaid on Result	2014	GBXT	[2]	X	X		X
		2012	DataPlay	[1]	X	X		X
		2006	Tabulator	[8]	X	X		X
		2002	Polaris/Tableau	[46]	X	X		
	Spreadsheet Formulas	2016	Object Spreads.	[37]	X	X		X
		2010	Spreads. as DB	[48]	X	X		
		2005	A1	[30]	X	X		X
		1997	OOF Spreads.	[15]	X	X		X
		1994	Forms/3	[10]	X	X		
	Exposed Algebraic	2013	Mashroom	[20]	X		X	X
		2011	Wrangler	[29]	X		X	
		1991	TableTalk	[18]	X		X	X
	Hidden Algebraic	2016	Gneiss	[13]	X		X	
		2013	GestureDB	[38]	X		X	
		2010	CRIUS	[41]	X			X
		2009	SheetMusiq	[35]	X			
		2008	AppForge	[50]	X			X
		1989	R^2	[22]	X		X	X
No	Diagram-based	2014	VisualTPL	[14]				X
		2009	App2You	[31]				X
		2005	QBB	[40]				
		2002	QURSED	[39]				X
		1990	QBD	[3]				
	Form-based	2008	Form Cust.	[28]				
		1998	QBEN	[36]				X
		1997	ESCHER	[49]				X
		1989	PERPLEX	[44]				
		1977	QBE	[51]				

Table 1: Summary of related systems, evaluated as visual query interfaces. R1 is indicated where some class of queries can be initially specified by direct manipulation of results. R2 is indicated where all parts of such queries can subsequently be modified through similar means. R3 is indicated where the same class of queries is relationally complete and supports aggregation in arbitrary multi-block queries.

a single user interface design. The key insight is that given a suitable data model for results, the complete structure of a query can be encoded in the schema of the query's own result. This in turn allows the user interface to display the query and its result in a single visual representation, which can then be manipulated directly to modify any part of the query. Specifically, we allow queries to produce results from the nested relational data model [24, 33], and display results using a nested table layout [6]. In our visual representation, the header area of the result's nested table layout encodes the structure of the query, which can then be manipulated using spreadsheet idioms such as formulas and filters. The use of nested results affords a natural visualization of operations such as joins and aggregation, and allows the user to see, in context, intermediate tuples produced in any part of the query.

Using our system, the user can express a relationally complete [16] set of query operators plus calculation, aggregation, outer joins, sorting, and nesting [5, Appendix A]. This covers the full set of query operators generally considered as the minimum to model SQL [7, 21], and expresses, for example, all `SELECT` statements valid in SQL-92.

In an initial formative user study, 14 participants were able to solve complex query tasks with a minimal amount of training, with many expressing strong levels of satisfaction with the tool. In a second, controlled study, another 14 participants rated both SIEUFERD and the query designer found in Microsoft Access on the System Usability Scale (SUS) [9] after doing a series of tasks on each. Users rated SIEUFERD 18 points higher on average than Access. This corresponds to a 46 percentage point difference on a percentile scale of other studies in the Business Software category.

This work was previously published at SIGMOD 2016 [5].

2. RELATED WORK

Visual query systems have been surveyed by Catarci et al. [11] and, recently, El-Mahgary and Soisalon-Soininen [17]. Systems discussed in this section include, in particular, those that employ direct manipulation, nested results, or optimizations for traversing relationships in the database. Table 1 categorizes systems by query representation style, and provides an assessment of each system against the requirements set forth in the introduction.

Besides our core requirements, Table 1 also indicates which systems support nested results, i.e. a graphical equivalent of a hierarchical data model such as XML, JSON, or nested relations. This handles report-style queries that encode multiple parallel one-to-many relationships in a single result, as when retrieving "a list of parts, and for each part a list of suppliers and a list of open orders" [6]. Systems that base their result representation on a single flat table of primitive values, such as Tableau [46], are unable to express such queries. The same tends to hold for any system that takes its input from a single joined SQL query, since multivalued dependencies [19] in the flattened result (PARTS↠SUPPLIERS and PARTS↠ORDERS in the preceding example) would interact to produce a pathological number of tuples for even small inputs. Some systems, like Tableau and Gneiss [13], support a restricted form of nesting, where an otherwise flat result table can be grouped into a single-branch hierarchy, or a finite set of such (a *dashboard* in Tableau, or a set of *hierarchical tables* in Gneiss). This still does not handle PARTS↠SUPPLIERS/ORDERS-type queries from the example above. Tableau, as well as other systems based on the pivot table concept, produce cross-tabulated rather than nested results; these concepts are orthogonal.

We first discuss visual query systems that do not fall in the direct manipulation category. *Form-based* systems originated with Query by Example (QBE) [51], where the user populates a set of empty *skeleton tables* with conditions, variables (*examples*), and output indications. ESCHER [49] and QBEN [36] extend QBE to support nested results, while PERPLEX [44] supports general-purpose logic programming. The ubiquitous search forms of commercial database applications can be seen as restricted versions of QBE tailored for a specific schema; Form Customization [28] generalizes such forms by considering the form designer as part of the query system. In *diagram-based* systems, the user manipulates queries for example through a schema tree or schema diagram, as in Query by Diagram (QBD) [3], Query by Browsing (QBB) [40], QURSED [39], and App2You [31], or through a diagrammatic query plan, as in VisualTPL [14]. The diagram-based query building style is common in commercial tools–Microsoft Access, Navicat, pgAdmin, dbForge, Alteryx etc. The general problem with both form-based and diagram-based interfaces is that users must manipulate queries through an abstract query representation that is divorced from the actual data that is being retrieved. To construct and understand queries, the user must look back and forth between the query representation on one side of the screen and a separate result representation on the other. Thus we do not consider these systems to be direct manipulation interfaces (requirement R1).

In the direct manipulation category, we now consider *algebraic* user interfaces. In such systems, the user builds queries by selecting, one step at a time, a series of operations to be applied to the currently displayed result. Each operation is applied to the result of all previous operations. Formal expressiveness is easy to achieve in algebraic interfaces, since the relevant relational operators can simply be exposed to the user directly. The main problem with algebraic interfaces is that the user has no direct way to, in the words of Liu and Jagadish, "modify an operation specified many steps ear-

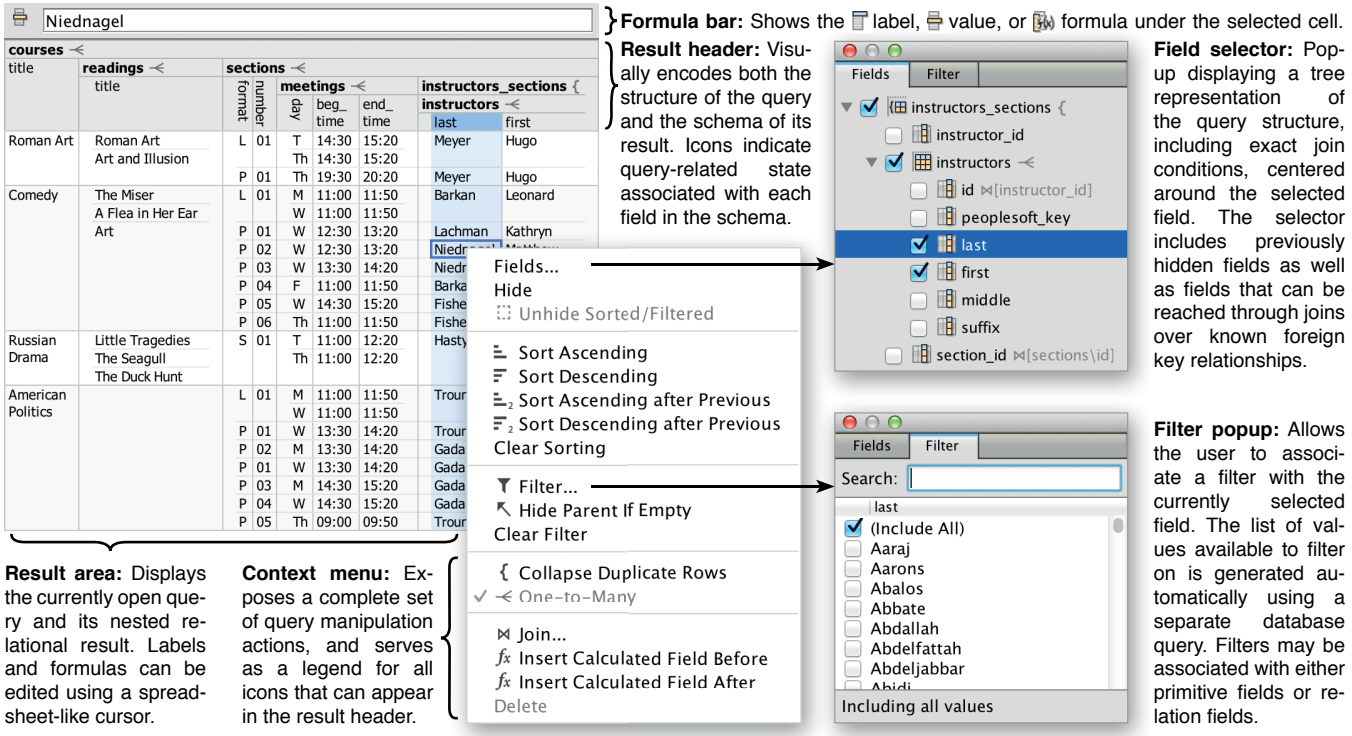


Figure 1: The SIEUFERD query interface. To create queries, users start from a simple tabular view of a table in the database and add filters, formulas, and nested relations. The integrated result and query representation is displayed continuously as the user interacts with the data. The particular query above instantiates six database tables (one per nested relation), contains five joins (each child relation against its parent), and is evaluated using five generated SQL queries (one for each one-to-many relationship ≺). This query was constructed purely by checking off the appropriate fields and foreign key relationships in the field selector.

lier without redoing the steps afterwards" [35] (requirement R2). For example, in GestureDB [38], the user has no way to modify a filter on a column that was subsequently used in an aggregation or removed with a projection. Similar problems exist in R^2 [22], AppForge [50], CRIUS [41], and Gneiss [13]. SheetMusiq [35] provides a partial solution by using an algebra where certain operators can commute out of a complex expression for subsequent modification; however, the technique breaks down for expressions enclosed in binary operators such as joins, set union, or set difference. In other systems, the underlying algebraic expression is exposed directly, as in the procedural *data manipulation scripts* of Wrangler [29], the XQuery-like *mashup scripts* of Mashroom [20], or the diagram-based representation in TableTalk [18]. Thus, only the initial query specification can be done through direct manipulation; tweaking and examination of existing queries must be done with a separate, indirect interface.

With clever use of formulas, Tyszkiewicz [48] shows that existing spreadsheet products can be considered expressive enough to formulate arbitrary SQL queries. If we consider Excel as a query system, however, only a subset of such queries could be said to be constructible by direct manipulation. Heavy reliance on set-based formula functions such as INDEX, MATCH, and SUMPRODUCT means that spreadsheet formulas soon take the role of a text-based query language, with a vocabulary far removed from that of typical query tasks. This would also be the case for spreadsheet programming systems such as Forms/3 [10], Object Oriented Functional Spreadsheets [15], A1 [30], and Object Spreadsheets [37].

Last, we consider direct manipulation systems that *overlay* their query representation on the result of the same query, with the structure of the query reflecting the visual structure of the result. This solves the mapping problem of requirement R2. The problem is that current such representations are not expressive enough to support arbitrary queries (requirement R3). For example, the direct manipulation interfaces of Tabulator [8] and GBXT [2] support filters and joins over schema relationships, but are unable to express calculation, aggregation, general-purpose joins, or other binary operators. In DataPlay [1], direct manipulation is used only to choose between universal and existential qualifiers. Tableau [46] allows a large class of two-dimensional visualizations to be created and manipulated through direct manipulation of table headers and corresponding axis *shelves*; however, queries involving calculations or binary operators must be configured using a separate interface rather than through direct manipulation. Our own system is the first to achieve SQL-like expressiveness from within a direct manipulation interface based on an overlaid query/result representation.

3. SYSTEM DESCRIPTION

3.1 Overview

Our core query building interface is shown in Figure 1. All user interactions are initiated from the *result area*, which shows the current query's nested relational result, formatted using a nested table layout. In a nested table layout, the table's *header* area visually encodes the schema of the nested result, including which fields are nested under others in the hierarchical schema. Because our system maps all query-related state to specific fields in the result schema, the result's table header simultaneously becomes a visual representation of the query that generated it. A set of icons, carefully designed to allow every aspect of the query state to be represented in the header, is used to augment the information that can be derived from the names and positions of fields.

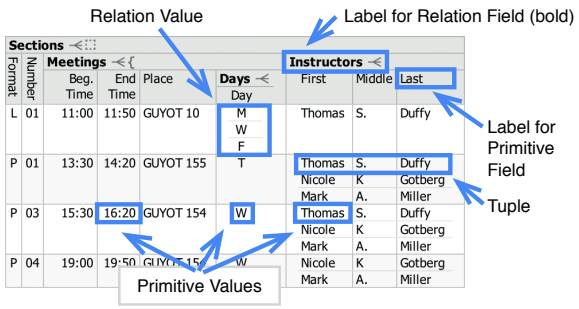


Figure 2: Terminology of the nested relational data model, illustrated on a nested table layout.

Starting from any selection of fields (columns) in the result area, the user may open a *context menu* of query-related actions, which also serves as a legend for icons that may appear in the result header. Query actions modify the query state, not the data in the database. Whenever a visual query is modified, the system generates and executes one or more corresponding SQL queries to evaluate it, merges the returned flat results into a single nested result, and displays the latter to the user. At the same time, the fields and iconography in the new result's header reflect the updated state of the modified query.

To keep the result layout compact, several aspects of the query state are indicated with icons in the header but are not displayed in full until the user requests it. In these cases we leverage well-established spreadsheet idioms to expose the underlying state. A filter icon (▼) next to a field label indicates the presence of a filter on that field, which can be manipulated by opening the *filter popup* from the context menu. A formula icon (f_x) indicates that the primitive field in question is a calculated field with an associated spreadsheet-style formula. The actual formula can be edited using the *formula bar* above the result area, or directly in any non-header cell belonging to the field's column. Finally, as in a spreadsheet, our system allows fields (columns) to be hidden from view and later recalled for inspection. If the hidden field was used for filtering or sorting, or is referenced from a formula, a dashed cell icon (⬚) is shown for the relevant dependent field to indicate that the visible result depends on a hidden portion of the query. Hidden fields can be recalled using the *field selector* popup, which shows an expandable list of available fields, centered around the field it was opened for. The field selector also serves to suggest new joins over known foreign key relationships, modeled as pre-existing hidden fields, and to display exact join conditions.

For the remainder of this paper, we will use the following terminology when referring to concepts in the nested relational data model: A *value* is either a *primitive* or a *relation*, where a relation is defined as a set of *tuples*, each containing a set of *fields* identified by *labels*, each containing a value, recursively. The *schema* of a value either defines the value to be a primitive, or defines the value to be a relation, with schemas further specified for each of the latter's fields, recursively. See Figure 2.

3.2 Query Model

We now discuss the specific structure of queries in our system. A visual query is modeled as a nested relational schema that has been annotated with query- and presentation-related properties on each field. We refer to the annotated schema as the SIEUFERD *query model*. When SQL queries are generated from a visual query and flat result sets have been assembled into a nested relational result, the schema of the nested result is identical to the schema in the query model. This correspondence makes it straightforward to translate high-level user interactions on the visualized query result to concrete modifications on the underlying query model, and conversely, to indicate the state of the query model in the table header of the visualized result.

Table instantiation. As a basic rule, each relation in the query model gets to retrieve data from one concrete table in the underlying database; that relation is said to *instantiate* the database table. The following is a simple query that instantiates the table called COURSES and displays a selection of its fields:

courses

id	area_id	title	may_pdf	may_audit	exam_type
56	2	Roman Art	N	Y	Other
177	2	Comedy	Y	Y	Final
845	2	Russian Drama	N	N	Other
1795	4	American Politics	Y	Y	Final
2566		Junior Seminars	N	N	Other
3921	4	Judicial Politics	Y	Y	Final

Nesting and joins. Queries need to be able to incorporate data from multiple tables. Commonly, tables need to be equijoined together, for example when the user wishes to examine data spread across foreign key relationships in a normalized database schema. In the SIEUFERD query model, the introduction of a new table instance can be done by defining a *nested relation*, optionally constrained by an equijoin condition against its parent relation:

courses						readings			
id	area_id	title	may_pdf	may_audit	exam_type	id	⋈ course_id	author_name	title
56	2	Roman Art	N	Y	Other	44	56	Ramage	Roman Art
						8,838	56	Gombrich	Art and Illusion
177	2	Comedy	Y	Y	Final	4,998	177	Moliere	The Miser
						12,138	177	Feydeau	A Flea in Her Ear
						16,878	177	Reza	Art
845	2	Russian Drama	N	N	Other	603	845	Pushkin	Little Tragedies
						9,207	845	Chekhov	The Seagull
						12,366	845	Vampilov	The Duck Hunt
1795	4	American Politics	Y	Y	Final				
2566		Junior Seminars	N	N	Other	9,935	2566	Pierre Loti	India
3921	4	Judicial Politics	Y	Y	Final	2,570	3921	Rosenberg, Gerald	The Hollow Hope
						17,629	3921	Lazarus, Edward	Closed Chambers

In the query above, the nested relation READINGS instantiates the database table with the same name, and equijoins itself against its parent relation COURSES on the COURSE_ID field, as indicated by the join icon (⋈) on the latter. The other side of the equijoin condition is the ID field in the COURSES relation. The latter information is omitted from the result layout to save space, but is displayed in the field selector (Figure 1). The one-to-many icon (≺) on the READINGS relation indicates that our system decided the latter may contain more than one tuple for each corresponding tuple in COURSES, the parent relation.

The joins described here have different semantics than the traditional flat joins encountered in SQL and most other visual query tools. Rather than duplicating tuples on one side of the operator for each occurrence of a matching tuple on the other, each tuple from the parent side of the join has a nested relation added to it holding zero or more matching tuples from the child side. This operator is known formally as a *nest equijoin* [45], though we will simply use the term *join* when unambiguous. One convenient property of nest equijoins is that tuples on the left-hand side of the operator do not disappear when the join fails to find matching tuples on the right; this can be seen in the query above for the course AMERICAN POLITICS, which has no books in its reading list.

It is often desirable to hide technical primary key fields, fields made redundant by equijoin conditions (e.g. COURSE_ID), or otherwise uninteresting fields, for presentation purposes. Continuing

the example above, our query model allows us to hide several fields without altering the query semantics:

courses					
title	may_ pdf	may_ audit	exam_ type	readings	
				author_name	title
Roman Art	N	Y	Other	Ramage	Roman Art
				Gombrich	Art and Illusion
Comedy	Y	Y	Final	Moliere	The Miser
				Feydeau	A Flea in Her Ear
				Reza	Art
Russian Drama	N	N	Other	Pushkin	Little Tragedies
				Chekhov	The Seagull
				Vampilov	The Duck Hunt
American Politics	Y	Y	Final		
Junior Seminars	N	N	Other	Pierre Loti	India
Judicial Politics	Y	Y	Final	Rosenberg, Gerald	The Hollow Hope
				Lazarus, Edward	Closed Chambers

The hidden fields could be recalled at any time using the field selector. As before, the field selector can also be used to see the exact join conditions between READINGS and COURSES.

Nested relations can be used very effectively to display data spread over many tables in a database schema. In the following example, we pull data from five database tables to see more information about each university course:

courses										
area		title	readings		sections					
title	code		author_ name	title	type	num	meetings			
							day	start	end	
Literature and the Arts	LA	Roman Art	Ramage	Roman Art	L	01	T	14:30	15:20	
			Gombrich	Art and Illusion			Th	14:30	15:20	
					P	01	Th	19:30	20:20	
Literature and the Arts	LA	Comedy	Moliere	The Miser	L	01	M	11:00	11:50	
			Feydeau	A Flea in Her Ear			W	11:00	11:50	
					P	01	W	12:30	13:20	
			Reza	Art	P	02	W	12:30	13:20	
					P	03	W	13:30	14:20	
					P	04	F	11:00	11:50	
					P	05	W	14:30	15:20	
					P	06	Th	11:00	11:50	
Literature and the Arts	LA	Russian Drama	Pushkin	Little Tragedies	S	01	T	11:00	12:20	
			Chekhov	The Seagull			Th	11:00	12:20	
			Vampilov	The Duck Hunt						

Notice that tuples in the READINGS relation occur independently of tuples in the SECTIONS relation; this kind of visualization can not be constructed in tools based on flat tabular results (see Related Work). Also notice the absence of the one-to-many icon (≺) on the AREA relation: because the latter relation was joined on its instantiated table's primary key, our system deduced that at most one tuple can exist in AREA for each parent tuple in COURSES.

Sorting. Each nested relation can be sorted on a sequence of its direct child fields, indicated by subscripted sort icons ($_{123}$) on the latter. In the following example, the root-level COURSES relation is sorted ascending on the MAX_ENROLL field, while individual sets of READINGS are sorted by AUTHOR_NAME, then by TITLE:

courses									
title	max _enroll	readings		sections					
		author_ name	title$_2$	type	num	meetings			
						day	start	end	
Russian Drama	0	Chekhov	The Seagull	S	01	T	11:00	12:20	
		Pushkin	Little Tragedies			Th	11:00	12:20	
		Vampilov	The Duck Hunt						
Junior Seminars	12	Pierre Loti	India	S	01	M	13:30	16:20	
Judicial Politics	24	Lazarus, Edward	Closed Chambers	L	01	W	11:00	11:50	
						F	11:00	11:50	
		Rosenberg, Gerald	The Hollow Hope	P	01	Th	14:30	15:20	
				P	02	W	13:30	14:20	
				P	04	W	11:00	11:50	
				P	03	T	11:00	11:50	
Roman Art	25	Gombrich	Art and Illusion	L	01	T	14:30	15:20	
		Ramage	Roman Art			Th	14:30	15:20	
				P	01	Th	19:30	20:20	

It is possible to sort on both primitive and relation fields, though we omit the exact semantics of the latter case here. Following any explicit sort terms, our system automatically sorts every relation on a tuple-identifying subset of its retrieved fields. This ensures that all query results are retrieved in a deterministic order. The automatic sort is usually on an indexed primary key; see *set projection* below.

Filter. Using the filter popup (Figure 1), a filter can be defined on any field, indicated by the filter icon (▼). Filters on relation fields restrict the set of tuples retrieved in that relation, while filters on primitive fields restrict the tuples of the parent relation. In the following example, the MEETINGS relation is filtered to show only tuples for which the DAY is W:

courses									
title	max _enroll	readings		sections					
		author_ name	title	type	num	meetings			
						day ▼	start	end	
Comedy	99	Moliere	The Miser	L	01	W	11:00	11:50	
		Feydeau	A Flea in Her Ear	P	01	W	12:30	13:20	
				P	02	W	12:30	13:20	
		Reza	Art	P	03	W	13:30	14:20	
				P	05	W	14:30	15:20	
American Politics	78			L	01	W	11:00	11:50	
				P	01	W	13:30	14:20	
				P	01	W	13:30	14:20	
				P	04	W	14:30	15:20	
Judicial Politics	24	Rosenberg, Gerald	The Hollow Hope	L	01	W	11:00	11:50	
				P	02	W	13:30	14:20	

By default, the effect of a filter in a nested relation is propagated all the way to the root of the query by means of a HIDE PARENT IF EMPTY setting on each intermediate relation, indicated by the arrow-towards-root icon (↖) on the SECTIONS and MEETINGS relations above. In the example, the courses ROMAN ART and RUSSIAN DRAMA have disappeared because they do not have any Wednesday sections. If, rather than retrieving "a list of courses with at least one Wednesday section", we wanted to retrieve "a list of all courses, showing sections on Wednesday only", we could deactivate HIDE PARENT IF EMPTY on the SECTIONS relation:

courses									
title	max _enroll	readings		sections					
		author_ name	title	type	num	meetings			
						day ▼	start	end	
Roman Art	25	Ramage	Roman Art						
		Gombrich	Art and Illusion						
Comedy	99	Moliere	The Miser	L	01	W	11:00	11:50	
		Feydeau	A Flea in Her Ear	P	01	W	12:30	13:20	
				P	02	W	12:30	13:20	
		Reza	Art	P	03	W	13:30	14:20	
				P	05	W	14:30	15:20	
Russian Drama	0	Pushkin	Little Tragedies						
		Chekhov	The Seagull						
		Vampilov	The Duck Hunt						
American Politics	78			L	01	W	11:00	11:50	
				P	01	W	13:30	14:20	

Formulas. An important part of the expressiveness offered by SQL is the ability to include scalar and aggregate computations over primitive values in any part of the query. In the SIEUFERD query model, both kinds of calculations are supported by means of *calculated fields*. A calculated field is a primitive field, added to any relation by the user, that takes its value from a *formula* rather than from a particular column in an instantiated database table. Like other fields, calculated fields can be sorted or filtered on.

SIEUFERD formulas are syntactically similar to spreadsheet formulas, but belong to and reference entire columns of field values rather than hard-coded ranges of cells. This allows SIEUFERD queries, like SQL queries, to be defined independently of the exact data that might reside in a database at any given time. Without this design, the user might have to rewrite formulas if the data in the underlying data source changes, or if other parts of the query are changed in such a way as to add or remove tuples in the result. Forgetting to update formulas when input data is changed is a common kind of error in spreadsheets [26, 12], which we avoid.

The restriction that calculated fields always be primitive fields is an important one; we do not wish formulas to take the role of a textual query language embedded within the visual one. Formulas do not provide a relational algebra, but rather allow simple computations over primitive values.

Continuing the course catalog example, we can calculate the duration of each meeting of a course section:

courses							
title	sections						
	type	num	meetings				
			day	start	end	f_x duration	
Roman Art	L	01	T	14:30	15:20	50	
			Th	14:30	15:20	=minutes([end] - [start])	
	P	01	Th	19:30	20:20	50	
Comedy	L	01	M	11:00	11:50	50	
			W	11:00	11:50	50	
	P	01	W	12:30	13:20	50	
	P	02	W	12:30	13:20	50	
	P	03	W	13:30	14:20	50	
	P	04	F	11:00	11:50	50	
	P	05	W	14:30	15:20	50	
	P	06	Th	11:00	11:50	50	
Russian Drama	S	01	T	11:00	12:20	80	
			Th	11:00	12:20	80	

The calculated field DURATION, marked with the formula icon (f_x), is evaluated once for each tuple in MEETINGS, its containing relation. Using another calculated field, we can add up the durations as well, at the level of each course:

=sum([duration])

courses								
title	f_x total duration	sections						
		type	num	meetings				
				day	start	end	f_x duration	
Roman Art	150	L	01	T	14:30	15:20	50	
				Th	14:30	15:20	50	
		P	01	Th	19:30	20:20	50	
Comedy	400	L	01	M	11:00	11:50	50	
				W	11:00	11:50	50	
		P	01	W	12:30	13:20	50	
		P	02	W	12:30	13:20	50	
		P	03	W	13:30	14:20	50	
		P	04	F	11:00	11:50	50	
		P	05	W	14:30	15:20	50	
		P	06	Th	11:00	11:50	50	
Russian Drama	160	S	01	T	11:00	12:20	80	
				Th	11:00	12:20	80	

When using aggregate functions such as SUM or COUNT, the relation in which the calculated field is defined determines the level at which aggregate values are grouped. In the example above, because the TOTAL DURATION field is a child of the COURSES relation, a total is calculated for each course rather than, say, for each section. Each course includes in its total only tuples from the MEETINGS relation that are descendants of that course's tuple in the COURSES relation.

Filters and aggregate functions. When an aggregate function references a relation with a filter applied to it, the filter is evaluated before the aggregate. In the following example, the SECTIONS relation is filtered to only include lecture-type sections. The TOTAL DURATION for each course changes accordingly:

courses									
title	f_x total duration	sections							
		type	num	meetings					
				day	start	end	f_x duration	f_x percent	
Roman Art	100	L	01	T	14:30	15:20	50	50.00	
				Th	14:30	15:20	50	50.00	
Comedy	100	L	01	M	11:00	11:50	50	50.00	
				W	11:00	11:50	50	50.00	
Russian Drama	0								
American Politics	100	L	01	M	11:00	11:50	50	50.00	
				W	11:00	11:50	50	50.00	
Junior Seminars	0								
Judicial Politics	100	L	01	W	11:00	11:50	50	50.00	
				F	11:00	11:50	50	50.00	

It is equally valid to define a filter on the output side of an aggregate, e.g. on TITLE or TOTAL DURATION in the example above.

Flat joins. Traditional flat joins can be expressed by referencing a descendant relation from a formula without enclosing the reference in an aggregate function. In the following example, each course title is repeated once for each distinct author name in the reading list, because the AUTHOR REFERENCE field in the COURSES relation references the READINGS relation without the use of an aggregate function:

courses				
title	exam_ type	author reference f_x	readings	
			author_name	title
Roman Art	Other	Gombrich	Gombrich	Art and Illusion
Roman Art	Other	Ramage	Ramage	Roman Art
Comedy	Final	Feydeau	Feydeau	A Flea in Her Ear
Comedy	Final	=[author_name]		The Miser
Comedy	Final	Reza	Reza	Art
Russian Drama	Other	Chekhov	Chekhov	The Seagull
Russian Drama	Other	Pushkin	Pushkin	Little Tragedies
Russian Drama	Other	Vampilov	Vampilov	The Duck Hunt
American Politics	Final			
Junior Seminars	Other	Pierre Loti	Pierre Loti	India
Judicial	Final	Lazarus	Lazarus, Edward	Closed Chambers

The actual behavior is that of a left join, with a null value being returned for the course AMERICAN POLITICS, which has no readings in its reading list. To express an inner join instead, the HIDE PARENT IF EMPTY setting could be enabled on the READINGS relation. The left join semantics of these *inward* formula references help our visual query language maintain some desirable properties. In particular, the mere introduction of a new calculated field (e.g. AUTHOR REFERENCE) will never cause tuples to disappear from said field's containing relation (COURSES).

Set projection. By default, tuples retrieved for a relation always include the primary key fields of the relation's instantiated table, even if the user has hidden those fields from view. This allows our system to keep result tuples in a stable order as the user hides or shows fields, and to keep a one-to-one relationship between tuples on the screen and tuples in instantiated database tables. It also allows us to generate more efficient SQL queries, for example by avoiding expensive `SELECT DISTINCT` statements. The automatic inclusion of primary key fields in the projection of a particular relation can be avoided by means of the HIDE DUPLICATE ROWS option, indicated by the bracket icon ({):

courses		
title	sections	
	type	status
Roman Art	L	O
	P	X
Comedy	L	O
	P	O
	P	O
	P	O
	P	O
	P	X
	P	X
Russian Drama	S	O

courses		
title	sections {}	
	type	status
Roman Art	L	O
	P	X
Comedy	L	O
	P	O
	P	X
Russian Drama	S	O
American Politics	L	O
	P	X
Junior Seminars	S	O
Judicial Politics	L	O

3.3 Query Building

Having explained the query model, we now show how the user would actually build queries using our direct manipulation interface. We do this by means of an example query building session. The user is an investigative journalist who is writing a story about ethanol biofuel lobbying[2]. She has compiled, in the table PLANTS_OS, a list of major ethanol producers[3], and would like to find the total lobbying expenditures of each. Another table, LOBBYING, contains quarterly lobbying reports from US corporations in the years 1998 through 2012 (727,927 tuples)[4].

[2]E. Díaz-Struck (2013). Ethanol Industry Battles to Keep Incentives. http://eye.necir.org/2013/05/26/ethanol-industry-battles-to-keep-incentives

[3]Renewable Fuels Association/Maple Etanol SRL (2012)

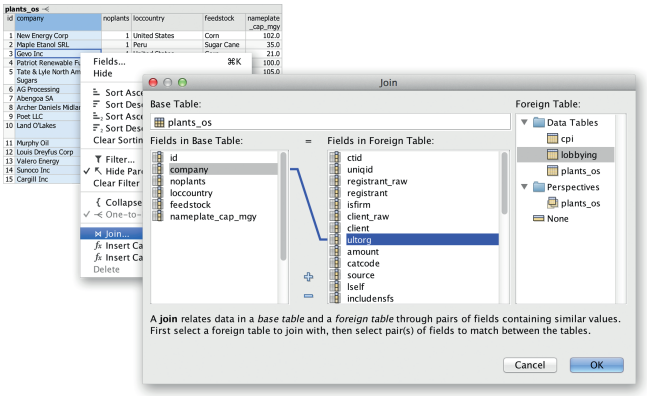
[4]The Center for Responsive Politics (2012) https://www.opensecrets.org

Base table. The user starts by opening the table of ethanol producers as a template for the new query:

plants_os

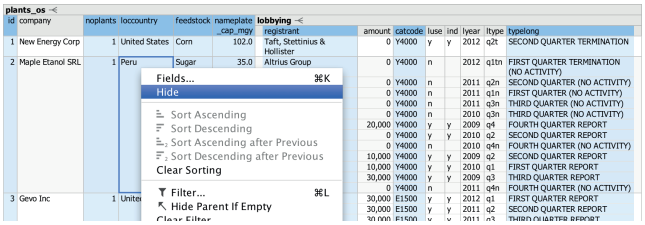
id	company	noplants	loccountry	feedstock	nameplate_cap_mgy
1	New Energy Corp	1	United States	Corn	102.0
2	Maple Etanol SRL	1	Peru	Sugar Cane	35.0
3	Gevo Inc	1	United States	Corn	21.0
4	Patriot Renewable Fuels	1	United States	Corn	100.0
5	Tate & Lyle North American Sugars	1	United States	Corn	105.0
6	AG Processing	1	United States	Corn	52.0

Join. To add another table to the query, the user selects the column or columns to join on and invokes the JOIN action from the context menu. This opens a dialog box for selecting the table to join with, in this case LOBBYING, and for selecting the corresponding columns from the latter to be matched in an equijoin constraint. The user joins the PLANTS_OS and LOBBYING tables on the COMPANY and ULTORG fields, respectively:



In cases where the database defines explicit foreign key relationships between tables, use of the above JOIN dialog is unnecessary; instead, all available joins will be available as hidden relations in the field selector. The effect is a schema navigation capability analogous to that of QBB [40], AppForge [50], and App2You [31].

Hide fields. After the join, a lot of columns are shown, so the user selects a few of them and invokes the HIDE action:



It is now easier to get a sense of the data. We have a new child relation field, called LOBBYING, containing the lobbying reports for each company:

plants_os

company	lobbying				
	amount	luse	ind	lyear	ltype
New Energy Corp	0	y	y	2012	q2t
Maple Etanol SRL	0	n		2012	q1tn
	0	n		2011	q2n
	0	n		2011	q1n
	0	n		2011	q3n
	10,000	y	y	2010	q3n
	30,000	y	y	2009	q3
	0	n		2011	q4n
Gevo Inc	30,000	y	y	2012	q1
	30,000	y	y	2011	q2
	30,000	y	y	2011	q3
	30,000	y	y	2009	q2

Sort. The user decides to sort the lobbying reports for each company most-recent-first, invoking the SORT DESCENDING action on the LYEAR field and then invoking the SORT DESCENDING AFTER PREVIOUS action on the LTYPE field. This sorts individual LOBBYING relations by year ($\blacktriangledown$) and then by quarter ($\blacktriangledown_2$):

plants_os

company	lobbying				
	amount	luse	ind	lyear	ltype
New Energy Corp	0	y	y	2012	q2t
Maple Etanol SRL	0	n		2012	q1tn
	0	n		2011	q4n
	0	n		2011	q3n
	0	n		2011	q2n
	0	n		2011	q1n
	0	n		2011	q3n
	10,000	y	y	2010	q3n
	30,000	y	y	2009	q3
	0	n		2011	q4n
Gevo Inc	30,000	y	y	2012	q1
	30,000	y	y	2011	q2
	30,000	y	y	2011	q3
	30,000	y	y	2009	q2

Aggregate formula. The user would now like to calculate a total lobbying amount for each company. She invokes the INSERT CALCULATED FIELD AFTER action to insert a calculated field (f_x) next to the COMPANY field, and enters the name SUM OF AMOUNTS in the new column's label cell. She then moves the cursor to one of the column's value cells, and enters a sum formula, clicking the AMOUNT column to insert the column reference:

=sum([amount])

plants_os

company	Sum of Amounts	lobbying				
		amount	luse	ind	lyear	ltype
New Energy Corp	0	0	y	y	2012	q2t
Maple Etanol SRL	70,000	0	n		2012	q1tn
		0	n		2011	q4n
		0	n		2011	q3n
					2011	q2n
		20,000	y	y	2009	
		30,000	y	y	2009	q3
		10,000	y	y	2009	q2
Gevo Inc	370,000	10,000	y	y	2012	q2
		30,000	y	y	2012	q1

Scalar formula. Reported lobbying amounts come from different years, some going back to 1998. The user would like to calculate inflation-corrected totals. A separate table CPI contains yearly Consumer Price Index values normalized for 2012. The user performs another JOIN, this time between LOBBYING and CPI, on the LYEAR and CYEAR fields, respectively. This brings the CPIV value for each lobbying report's year into the nested result. The user then adds another calculated field, this time under the same relation as the existing AMOUNT field, and enters a formula that calculates the inflation-adjusted amount for each report. We here have a useful example of an inward formula reference (to CPIV) that is not enclosed in an aggregate function:

plants_os

company	Sum of Amounts	lobbying						
		amount	Amount in 2012-dollars	luse	ind	lyear	cpi cpiv	ltype
New Energy Corp	0	0	0	y	y	2012	1.0000	q2t
Maple Etanol SRL	70,000	0	0	n		2012	1.0000	q1tn
		0	0	n		2011	0.9716	q4n
		0	0	n		2011	0.9716	q3n
		0	=[amount] / [cpiv]				0.9716	q2n
		0	0	n		2011	0.9716	q1n
		0	0	n		2010	0.9560	q4n
		0	0	n		2010	0.9560	q3n
		0	0	y	y	2010	0.9560	q2
		10,000	10,460	y	y	2010	0.9560	q1
		20,000	21,470	y	y	2009	0.9315	q4
		30,000	32,205	y	y	2009	0.9315	q3
		10,000	10,735	y	y	2009	0.9315	q2
		10,000	10,000	y	y	2012	0.9313	q2

cpi – Editor

cpi

cyear	cpiv
2012	1.0000
2011	0.9716
2010	0.9560
2009	0.9315
2008	0.9313

A new inflation-adjusted total can now be added as a calculated field at the PLANTS_OS level, shown adjacent to the existing non-adjusted sum:

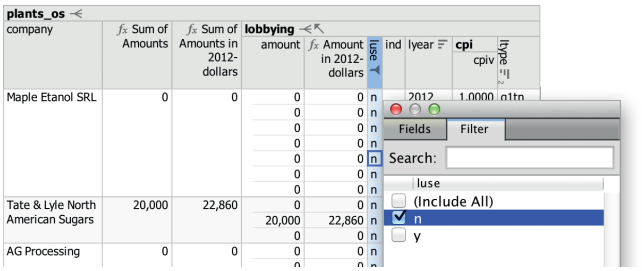
=sum([Amount in 2012-dollars])

plants_os

company	Sum of Amounts	Sum of Amounts in 2012-dollars	lobbying						
			amount	Amount in 2012-dollars	luse	ind	lyear	cpi cpiv	ltype
New Energy Corp	0	0	0	0	y	y	2012	1.0000	q2t
Maple Etanol SRL	70,000	74,870	0	0	n		2012	1.0000	q1tn
			0	0	n		2011	0.9716	q4n
			0	0	n		2011	0.9716	q3n
							2011	0.9716	q2n
			30,000	32,205	y	y	2009		
			10,000	10,735	y	y	2009	0.9315	q2
Gevo Inc	370,000	385,646	10,000	10,000	y	y	2012	1.0000	q2
			30,000	30,000	y	y	2012	1.0000	q1

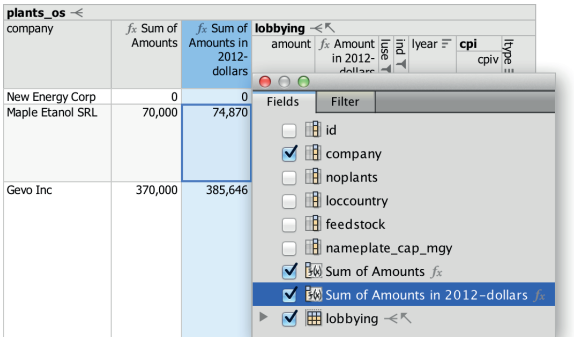
Filter. Lobbying reports may sometimes be amended, in which case the superseded reports should be excluded from totals to avoid double counting. The user can look for superseded reports

by invoking the FILTER action on the LUSE field and selecting the value N:



The user sees that there are superseded reports in the database with non-zero dollar amounts, and inverts the filter to exclude them.

Select fields. The user now decides to hide the individual reports altogether and instead reintroduce some of the fields that were hidden from the PLANTS_OS relation before, using the field selector:



Final touches. The user edits the field labels to make them a bit more readable, and sorts the companies by their lobbying totals. The underlying SQL column names can still be seen in the field selector. The user also enables a formatting option on the last column to produce a bar chart visualization. The result now looks presentable:

Lobbying by Ethanol Producers

Company	Plants	Feedstock	Sum of Amounts in 2012-dollars
Cargill Inc	2	Corn	16,725,489
Sunoco Inc	1	Corn	15,277,872
Archer Daniels Midland	8	Corn	9,277,472
Murphy Oil	1	Corn	7,729,618
Valero Energy	10	Corn	7,047,974
Land O'Lakes	1	Cheese Whey	4,821,907
Poet LLC	27	Corn	3,769,377
Louis Dreyfus Corp	2	Corn	2,310,378
Tate & Lyle North American Sugars	1	Corn	2,204,061
Abengoa SA	6	Corn	1,585,509
Gevo Inc	1	Corn	385,646

While the LOBBYING relation that feeds into the aggregate formula is now hidden, the user could easily make it visible again from the field selector, like she did for the previously hidden PLANTS and FEEDSTOCK fields. There are also shortcuts for unhiding hidden fields referenced from the formula, or the hidden filter, indicated by the dashed cell icons (⬚).

4. FORMATIVE USER STUDY

We conducted a formative user study with 14 participants (denoted A through N, 5 male, median age 42). 7 had experience with SQL, 11 used Excel daily. In the first part of the study, done by users A-I, users were given standardized tasks aimed at assessing the initial learnability of our tool. No prior training was given; instead, initial tasks were designed to act as training tasks for subsequent ones. In the second part of the study, and as time permitted during earlier sessions, users were given a chance to do more open-ended tasks on datasets we provided. Here, we gave participants demos and instructions for operating our tool, in order to gather higher-level observations than would be possible during pure learning tasks. In this section we discuss a selection of observations from our study; see our full paper [5] for more details.

Manual joins. Performing the lobbying query from Section 3.3, most users moved through the manual join dialog quickly and correctly on their first attempt. Still, users preferred automatic joins once introduced to them, see below. Users attempting the inflation correction portion of the query had no problems with the join against the CPI table; only users DG required a hint that they would need to use the JOIN feature again.

Formulas. When first attempting to perform a sum aggregation, users BCDE started by looking for an explicit sum action, as would be found in Excel's toolbar. Users CGK looked for an Excel-style formula builder. Having eventually realized that they needed to insert a calculated field and enter a formula themselves, users DEFK had initial trouble learning how to physically enter the formula, trying for example to enter the formula in an already-existing column, or in the column header.

In Excel, sums can be produced either using formulas or pivot tables. The two interfaces are largely separate, with users often preferring one or the other. Our system follows the formula approach. Users CH commented that they thought of pivot tables when first trying to compute a sum, while users BEI thought of pivot tables during other tasks.

A significant difference between spreadsheet formulas and SIEUFERD formulas is that the latter, like SQL queries, reference entire columns of values rather than an explicit range of cells. Users ABCFH expected this on their first attempts to insert a reference in a sum formula. Users DEGN expected the spreadsheet model, initially attempting to select a range of cells. A related challenge was to understand the level at which a calculated field should be inserted in order for sums to be grouped in the right way. The fact that the position of a formula in the relation hierarchy determines the grouping of aggregate functions is a further deviation from the spreadsheet model, while the lack of an explicit `GROUP BY` clause may be confusing to SQL users. User H tried to specify the set of columns to group by in the aggregate function itself, as in the formula =SUM([NAME],[AMOUNT]), while user F tried to hide every field other than the one to be summed. User G attempted to invoke the HIDE DUPLICATE ROWS action. Users CFGH also tried placing the calculated field next to the value to be summed rather than at the parent level. User G thought aloud:

"Wouldn't it be fantastic if there was a way simply to operate at that group level rather than these individual entries? [After creating a new formula at the correct level:] Is it doing it that way? Oh, that's perfect. ... That is meeting my heart's desire. But I wouldn't have the cue for that."

Despite initial difficulty with formulas in a training task, users applied them quickly and accurately in a follow-up task, despite the follow-up task requiring more steps. This suggests users are able to apply formulas effectively after first learning them, but that there is significant potential for improved learnability. We agree with users AM, who suggested adding an explicit sum action like that of Excel. This feature would automatically generate a sum formula above the nearest one-to-many relationship, which would then serve as an example to the user to learn from.

After initial learning, users appreciated the behavior of formulas. Users CEGK noted explicitly that the behavior of aggregate functions, including grouping and subtotaling behavior, made sense. Users ILK also commented that the all-column nature of formula references made sense and was an advantage over Excel's range-style references. User K noted:

"I just feel like I have a truer sense of what I'm adding up, or

what's being considered in this format vs. the traditional Excel. Because [in Excel] you could be pulling from the wrong places, you can be getting weird numbers, you could accidentally hit a field that now ends up in your calculation."

Field selection; automatic joins. Working on the course catalog dataset that was seen in Section 3.2, users were generally able to use the automatic foreign key join feature without trouble. The exception was user N, who had a hard time because of the lack of visible indications in the result area that more fields could be shown. User G also noted this issue. Users IKN specifically looked for an action named "Unhide", like in Excel. This suggests that our user interface needs a more visible affordance for accessing hidden fields. We expect hidden fields to be far more common in SIEUFERD than in Excel, since a typical database query projects only a small subset of columns available from instantiated database tables. The design of an improved unhide affordance should take this into account.

Users EGHJKL reacted particularly enthusiastically to the automatic join feature, using words such as "fantastic", "wow", "damn", and "amazing". User E noted:

"Yes, the manual join made sense, but that was a very simple situation. I wouldn't want to have done the joins on this [more complicated database]. The fact that I was just able to double-click and expand it out, that meant, it dumbed the task down to the level that I was happy performing it."

5. CONTROLLED USER STUDY

In a second user study, we aimed to get a more precise idea of how users might rate our system compared to an existing industry tool. We chose the "Query Design" facility of Microsoft Access 2016 as a control. Being part of the Office Professional suite, it is one of the most common visual query tools available. It is also a good example of a query builder that uses a diagram-based approach rather than direct manipulation of results (see Related Work).

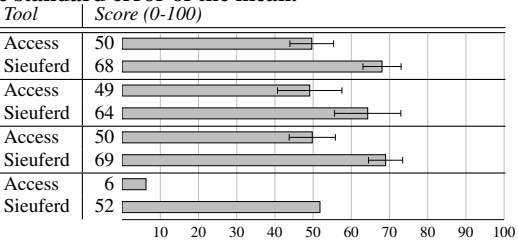
The controlled study was a within-subjects counterbalanced design, measuring usability using the System Usability Scale (SUS) [9]. Tullis and Stetson [47] recommend sample sizes of 12-14 users to get reasonably representative results from within-subjects studies based on the SUS survey; we collected data from 14 users (5 male, median age 36). 2 had prior experience with the Access query designer, 6 had significant exposure to SQL. 2 used Excel daily, the rest weekly or monthly. We met with each user for a single study session, structured as follows:

1. Complete demographic/background survey.
2. Briefly discuss the sample database that will be used for tasks, consulting a schema diagram on paper. The paper diagram remains available to the user during the tasks that follow.
3. Work through some standardized tasks to evaluate Tool 1. Stop after about 20 minutes. The first tool is SIEUFERD for half of the users and Microsoft Access for the other half, randomized.
4. Complete SUS survey for Tool 1.
5. Work through the same tasks in Tool 2, under otherwise identical conditions. Stop after about 20 minutes.
6. Complete SUS survey for Tool 2.
7. Discussion and feedback.

The standardized tasks [5], all done on the 7-table "Northwind" example database that shipped with older versions of Microsoft Access, are intended to be realistic examples of queries that a user might want to run on such a database. They incorporate joins, filters, sorting, scalar calculations and aggregates, but are limited to queries that can be expressed in Microsoft Access' visual query designer; this excludes queries requiring nested results as well as multi-block queries (e.g. aggregates used as inputs to other aggregates). In both tools, we configured foreign key relationships upfront so that the user would not have to manually specify exact join constraints between tables. The first five tasks are guided training tasks, intended to expose the user to all features, in both tools, that are needed to complete the subsequent unguided tasks. The guided tasks tended to take about half of the 20 minutes that users had available to try each tool. After the guided tasks, users were asked to try solving four unguided tasks without help. Since the main purpose of tasks was to give the user enough of an impression of each system to complete the subsequent SUS survey, we gave hints during unguided tasks whenever users reported being stuck.

Table 2: Mean SUS survey results for the controlled study, using various standard scales. Higher scores are better. Error bars show the standard error of the mean.

Scale	*Tool*	*Score (0-100)*
Raw SUS	Access	50
	Sieuferd	68
Learnability	Access	49
	Sieuferd	64
Usability	Access	50
	Sieuferd	69
Percentile	Access	6
	Sieuferd	52



The results of the study are shown in Table 2. The raw SUS score is reported along with separate Learnability and Usability scores as defined by Lewis and Sauro [34], as well as a percentile rating among 30 other studies in the B2B (Business Software) category as detailed by Sauro [42]. The difference in raw SUS scores between Access and SIEUFERD is statistically significant ($p = 0.0019$ with two-tailed paired t-test).

Interpreting the results, with the caveat that these observations are based on only 20-minute interactions with each tool, we see that SIEUFERD significantly outperformed Microsoft Access in terms of usability. Most of the difference can be attributed to the poor performance of Microsoft Access, considering its low ranking on the percentile scale; SIEUFERD simply achieved an average rating compared to other business software. This supports the original hypothesis of our paper: database querying is hard, but can be made significantly easier using a direct manipulation interface. SIEUFERD still has significant potential for improved usability. In conversations with users, the main requests for future design improvements were (1) the ability to get an overview of the complete database schema from within the query interface and (2) reduced dependency on formulas during query building. This is consistent with observations from the formative study.

6. CONCLUSION

SIEUFERD is a visual query system that achieves SQL-like expressiveness from a pure direct manipulation interface. Whereas previous direct manipulation systems either sacrifice expressiveness or hide the actual query from the user, SIEUFERD integrates the query and its result into a single interactive visualization, using spreadsheet concepts like filters and formulas to expose the complete state of the current query. Compared with the diagram-based query designer of Microsoft Access 2016, users greatly preferred our direct manipulation interface, with the latter scoring 46 percentiles higher on a SUS-based percentile scale. In future work, we hope to incorporate *editing* of data in our system; this will allow SIEUFERD to act as a complete schema-independent end user front-end for relational databases.

7. REFERENCES

[1] A. Abouzied, J. Hellerstein, and A. Silberschatz. DataPlay: Interactive tweaking and example-driven correction of graphical database queries. In *Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12)*, pages 207–218, New York, NY, USA, 2012. ACM.

[2] S. Achler. GBXT: A gesture-based data exploration tool for your favorite database system. In *Model and Data Engineering*, pages 224–237. Springer International Publishing, Cham, Switzerland, 2014.

[3] M. Angelaccio, T. Catarci, and G. Santucci. Query by Diagram: A fully visual query system. *Journal of Visual Languages & Computing*, 1(3):255–273, 1990.

[4] E. Bakke and E. Benson. The schema-independent database UI: A proposed holy grail and some suggestions. In *Proceedings of the 5th Biennial Conference on Innovative Data Systems Research (CIDR '11)*, 2011.

[5] E. Bakke and D. R. Karger. Expressive query construction through direct manipulation of nested relational results. In *Proceedings of the 2016 International Conference on Management of Data (SIGMOD '16)*, pages 1377–1392, New York, NY, USA, 2016. ACM.

[6] E. Bakke, D. R. Karger, and R. C. Miller. Automatic layout of structured hierarchical reports. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2586–2595, December 2013.

[7] E. Baralis and J. Widom. An algebraic approach to static analysis of active database rules. *ACM Transactions on Database Systems (TODS)*, 25(3):269–332, September 2000.

[8] T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI '06)*, 2006.

[9] J. Brooke. SUS: A quick and dirty usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, editors, *Usability evaluation in industry*, pages 189–194. Tailor & Francis, London, UK, 1996.

[10] M. Burnett, J. Atwood, R. Walpole Djang, J. Reichwein, H. Gottfried, and S. Yang. Forms/3: A first-order visual language to explore the boundaries of the spreadsheet paradigm. *Journal of Functional Programming*, 11:155–206, March 2001.

[11] T. Catarci, M. F. Costabile, S. Levialdi, and C. Batini. Visual query systems for databases: A survey. *Journal of Visual Languages & Computing*, 8(2):215–260, 1997.

[12] J. P. Caulkins, E. L. Morrison, and T. Weidemann. Spreadsheet errors and decision making: Evidence from field interviews. *Journal of Organizational and End User Computing*, 19(3):1, 2007.

[13] K. S.-P. Chang and B. A. Myers. Using and exploring hierarchical data in spreadsheets. In *Proceedings of the 34th Annual ACM Conference on Human Factors in Computing Systems (CHI '16)*, New York, NY, USA, 2016. ACM.

[14] W.-K. Chen and P.-Y. Tu. VisualTPL: A visual dataflow language for report data transformation. *Journal of Visual Languages & Computing*, 25(3):210–226, 2014.

[15] C. Clack and L. Braine. Object-oriented functional spreadsheets. In *Proceedings of the 10th Glasgow Workshop on Functional Programming (GlaFP '97)*, 1997.

[16] E. F. Codd. Relational completeness of data base sublanguages. In *Database Systems*, pages 65–98. Prentice Hall, 1972.

[17] S. El-Mahgary and E. Soisalon-Soininen. A form-based query interface for complex queries. *Journal of Visual Languages & Computing*, 29:15–53, 2015.

[18] R. G. Epstein. The TableTalk query language. *Journal of Visual Languages & Computing*, 2(2):115–141, 1991.

[19] R. Fagin. Multivalued dependencies and a new normal form for relational databases. *ACM Transactions on Database Systems (TODS)*, 2(3):262–278, 1977.

[20] Y. Han, G. Wang, G. Ji, and P. Zhang. Situational data integration with data services and nested table. *Service Oriented Computing and Applications*, 7(2):129–150, 2013.

[21] L. Hella, L. Libkin, J. Nurmonen, and L. Wong. Logics with aggregate operators. *Journal of the ACM (JACM)*, 48(4):880–907, July 2001.

[22] G.-J. Houben and J. Paredaens. A graphical interface formalism: Specifying nested relational databases. In *Proceedings of the IFIP TC2 Working Conference on Visual Database Systems*, pages 257–276, 1989.

[23] Y. E. Ioannidis. Visual user interfaces for database systems. *ACM Computing Surveys (CSUR)*, 28(4es), 1996.

[24] G. Jaeschke and H. J. Schek. Remarks on the algebra of non first normal form relations. In *Proceedings of the 1st ACM SIGACT-SIGMOD Symposium on Principles of Database Systems (PODS '82)*, pages 124–138, New York, NY, USA, 1982. ACM.

[25] H. V. Jagadish, A. Chapman, A. Elkiss, M. Jayapandian, Y. Li, A. Nandi, and C. Yu. Making database systems usable. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 13–24, New York, NY, USA, 2007. ACM.

[26] D. Janvrin and J. Morrison. Using a structured design approach to reduce risks in end user spreadsheet development. *Information & management*, 37(1):1–12, 2000.

[27] M. Jayapandian and H. V. Jagadish. Automated creation of a forms-based database query interface. *Proceedings of the VLDB Endowment*, 1:695–709, August 2008.

[28] M. Jayapandian and H. V. Jagadish. Expressive query specification through form customization. In *Proceedings of the 11th International Conference on Extending Database Technology (EDBT '08)*, pages 416–427, New York, NY, USA, 2008. ACM.

[29] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the 2011 annual conference on Human Factors in Computing Systems (CHI '11)*, pages 3363–3372, New York, NY, USA, 2011. ACM.

[30] E. Kandogan, E. Haber, R. Barrett, A. Cypher, P. Maglio, and H. Zhao. A1: End-user programming for web-based system administration. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*, pages 211–220, New York, NY, USA, 2005. ACM.

[31] K. Kowalzcykowski, A. Deutsch, K. W. Ong, Y. Papakonstantinou, K. K. Zhao, and M. Petropoulos. Do-It-Yourself database-driven web applications. In *Proceedings of the 4th Biennial Conference on Innovative Data Systems Research (CIDR '09)*, 2009.

[32] D. Król, J. Oleksy, M. Podyma, and B. Trawiński. The analysis of reporting tools for a cadastre information system. In *Proceedings of the 9th International Conference on Business Information Systems (BIS '06)*, pages 150–163, 2006.

[33] M. Levene. *The Nested Universal Relation Database Model*, volume 595 of *Lecture Notes in Computer Science*. Springer Berlin/Heidelberg, 1992.

[34] J. R. Lewis and J. Sauro. The factor structure of the system usability scale. In *Proceedings of the 1st International Conference on Human Centered Design (HCD '09)/HCI International 2009*, pages 94–103, Berlin, Heidelberg, 2009. Springer-Verlag.

[35] B. Liu and H. V. Jagadish. A spreadsheet algebra for a direct data manipulation query interface. In *Proceedings of the IEEE 25th International Conference on Data Engineering (ICDE '09)*, pages 417–428, April 2009.

[36] N. Lorentzos and K. Dondis. Query by Example for Nested Tables. In *Database and Expert Systems Applications*, pages 716–725. Springer, 1998.

[37] R. M. McCutchen, S. Itzhaky, and D. Jackson. Initial report on Object Spreadsheets. Technical Report MIT-CSAIL-TR-2016-001, MIT Computer Science and Artificial Intelligence Laboratory, January 2016.

[38] A. Nandi, L. Jiang, and M. Mandel. Gestural query specification. *Proceedings of the VLDB Endowment*, 7(4):289–300, 2013.

[39] Y. Papakonstantinou, M. Petropoulos, and V. Vassalos. QURSED: Querying and reporting semistructured data. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pages 192–203, New York, NY, USA, 2002. ACM.

[40] S. Polyviou, G. Samaras, and P. Evripidou. A relationally complete visual query language for heterogeneous data sources and pervasive querying. In *Proceedings of the 21st International Conference on Data Engineering (ICDE '05)*, pages 471–482, Washington, DC, USA, 2005. IEEE Computer Society.

[41] L. Qian, K. LeFevre, and H. V. Jagadish. CRIUS: User-friendly database design. *Proceedings of the VLDB Endowment*, 4(2):81–92, 2010.

[42] J. Sauro. *A practical guide to the System Usability Scale: Background, benchmarks & best practices*. Measuring Usability LLC, 2011.

[43] B. Shneiderman. Direct Manipulation: A step beyond programming languages. *IEEE Computer*, 16(8):57–69, 1983.

[44] M. Spenke and C. Beilken. A spreadsheet interface for logic programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '89)*, pages 75–80, New York, NY, USA, 1989. ACM.

[45] H. J. Steenhagen, P. M. G. Apers, and H. M. Blanken. Optimization of nested queries in a complex object model. In *Proceedings of the 4th International Conference on Extending Database Technology (EDBT '94)*, pages 337–350, New York, NY, USA, 1994. Springer New York.

[46] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional databases. *Communications of the ACM*, 51(11):75–84, November 2008.

[47] T. S. Tullis and J. N. Stetson. A comparison of questionnaires for assessing website usability, 2004. Usability Professionals Association (UPA) 2004 Conference.

[48] J. Tyszkiewicz. Spreadsheet as a relational database engine. In *Proceedings of the 2010 International Conference on Management of Data (SIGMOD '10)*, pages 195–206, New York, NY, USA, 2010. ACM.

[49] L. Wegner, S. Thelemann, J. Thamm, D. Wilke, and S. Wilke. Navigational exploration and declarative queries in a prototype for visual information systems. In C. Leung, editor, *Visual Information Systems*, volume 1306 of *Lecture Notes in Computer Science*, pages 199–218. Springer Berlin/Heidelberg, 1997.

[50] F. Yang, N. Gupta, C. Botev, E. F. Churchill, G. Levchenko, and J. Shanmugasundaram. WYSIWYG development of data driven web applications. *Proceedings of the VLDB Endowment*, 1(1):163–175, 2008.

[51] M. M. Zloof. Query-by-Example: A data base language. *IBM Systems Journal*, 16(4):324–343, 1977.

On the Intuitiveness of Common Discretization Methods

[Short Version]

Mario Boley
Cluster of Excellence MMCI and Saarland
University Saarbrücken, Germany
mboley@mmci.uni-saarland.de

Ankit Kariryaa
University of Bielefeld
Bielefeld, Germany
ankit.ky@gmail.com

ABSTRACT

Data discretization methods are usually evaluated in terms of technical criteria that are related to some specific data analysis goal like the preservation of variable interactions. In this paper, we provide a different evaluation principle that assesses the quality of a chosen discretization as the degree to which it coincides with human intuition. This is motivated from the setting of interactive exploratory data analysis where discretizations should be simple, self-explanatory, and fix across results in order to reduce the cognitive load on the user. We present a study design for measuring the intuitive discretization choices of a general human population for a set of discretization problems and present the results of a study trial that we performed with 153 respondents and four problem classes—each using the categories "low", "normal", and "high". Through this trial, we evaluated eight discretization methods from three families: range-based discretization, count-based discretization, and clustering-based discretization. Our results partially confirm results from Cognitive Linguistics that assume prototype-based categorization, which is most closely resembled by clustering-based methods, as a predominant human discretization mechanism. They also show, however, an affinity of participants to sometimes compromise cluster quality in favor of approximating certain category proportions.

1. INTRODUCTION

Metric measurements, i.e., numerical data adhering to an interval or a ratio scale, are ubiquitous in real-world data analysis. Yet, many analysis algorithms require at least part of their input data in the form of simple binary features (e.g., Subgroup Discovery [Atzmueller, 2015], Redescription Mining [Parida and Ramakrishnan, 2005], and various data summarization techniques [Wille, 2005, Vreeken et al., 2011, Geerts et al., 2004]). This is why the data mining and statistics literature provides a wide range of data discretization techniques that can be used for producing such features from metric input (see, e.g., Kontkanen and Myllymäki [2007], Chapeau-Blondeau and Rousseau [2009], Nguyen et al. [2014]). Usually these techniques are evaluated solely from the technical perspective of how well they retain properties of the original data distribution and/or how they affect the performance of specific data analysis algorithms. In this paper we provide the complementing evaluation perspective of **intuitive linguistic discretization** in which one asks *how well is a discretization enabling an effective interaction between computer algorithms and human users as well as facilitating a discussion of algorithmic findings among humans.*

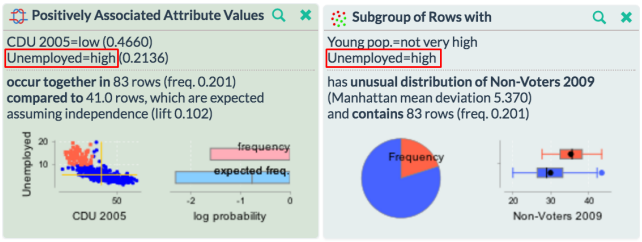


Figure 1: Two data analysis results produced by different algorithms in Creedo [Boley et al., 2015], both of which use the self-explanatory symbol "unemployed=high"; it is desirable that symbol has fix definition across results and that this definition is intuitive, i.e., approximately coinciding with user's category "high" (were she to know the distribution of "unemployment").

This perspective is relevant whenever algorithmic results are supposed to be interpreted by humans; especially when there are many such results as it is characteristic for exploratory data analysis and pattern discovery tasks. For example, consider a data scientist operating an interactive pattern discovery suite (e.g., MIME [Goethals et al., 2011], Cortana [Meeng and Knobbe, 2011], or VIKAMINE [Atzmueller and Lemmerich, 2012]). Typically, the scientist would run a number of data analysis algorithms with different parameter settings, the results of each of which she would investigate and compare with one another. Finally, she would distill out the most important findings for further discussion with her peers. From this scenario we can derive several desirable properties for discretization:

1. Since the results of different methods and different parameters should be comparable to one another, we want a stable and *generic* discretization that works

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16) August 14th, 2016, San Francisco, CA, USA.

ACM ISBN .

DOI:

reasonable well for various tasks and typical analysis methods. This is in contrast to discretizations that are optimized for one specific setting as it is the case for supervised discretization techniques.

2. Moreover, the discretization should be *self-explanatory* in order to reduce the cognitive load of the data scientist. That is, we are looking for a discretization that summarizes metric variables in a comprehensible way through a small number of linguistic terms like "low", "normal", and "high".

3. Finally, the discrete symbols should be *intuitive*. That is, ideally the symbols' definitions approximately correspond to those that humans would instinctively pick themselves to talk about the data domain among each other.

Fig. 1 summarizes these criteria for an exemplary result set produced by different pattern discovery algorithms. Based on requirements 1 and 2, we think of an abstract (exact) **linguistic discretization problem** as: *given* a sample S of values of a metric variable defined on a real interval X and a set of k ordered linguistic quantification categories, *find* $k-1$ cut-off values in X that separate the given categories for that variable. Based on requirement 3, we say that a discretization given by a set of cut-off values is **intuitive** if it tends to be close to the set of cut-off values that users of a desired target audience would pick themselves had they to make their choice purely based on the sample S (as opposed to *concrete* linguistic discretization tasks where prior information about the variable is available). In this article we investigate empirically the degree to which common discretization approaches exhibit this form of intuitiveness.

Studying the precise mechanism of human discretization is a profound topic with connections to Linguistics (where it is referred to as *categorization*, see Taylor [2003] and references therein) as well as Cognition and Neuroscience (e.g., Dehaene et al. [1998, 2008]). Here, we generally take on a rather naive point of view and simply propose to test how well algorithmic discretization of quantities aligns with human categorization while staying agnostic about the precise mechanism that governs it. A particular interesting proposition from Cognitive Linguistics [Evans, 2007], that we take up here, is that the predominant mechanism for human linguistic discretization is based on prototypes (going back to a seminal work of Rosch [1973] in Cognitive Psychology). This proposition says that categories are associated with typical representative members (and that there can be values that are not a real representative of any category). In Computer Science this intuition was formalized as fuzzy linguistic discretization through fuzzy logic (see Ishibuchi et al. [2006] and references therein). This approach, however, requires specific analysis and model induction algorithms. Here we are interested in a general purpose preprocessing method, and, hence, we focus on traditional interval-based (or exact) discretization methods. Among those, clustering-based methods come closest to the idea of prototype-based categorization. Therefore we put a special emphasis on the evaluation of those methods.

To summarize the **contributions** of this paper: firstly, we develop a study design for measuring the intuitive discretization choices of a general target audience and that therefore operationalizes all of the theoretical concepts mentioned above. Secondly, we report results that have been generated with this design through an open study trial involving 153 participants that was particularly targeting the general categories "low", "normal", and "high". Our findings partially confirm the prototype-based proposition, but also show that it is violated when the distribution of the input sample is spread out too uniformly. In particular, we observed an affinity of participants to sometimes compromise cluster quality in favor of approximating certain category proportions.

Figure 2: Cut-off values of geometric-width discretization for $k = 7$ and $X = [0, 1]$ or the quantiles of geometric-frequency labeling for $k = 7$.

2. FORMAL DISCRETIZATION METHODS

In this section we define the formal discretization methods that we want to evaluate. First, however, we need to fix some basic notation. Let $X = [a, b] \subseteq \mathbb{R}$ be the real interval given by the upper and lower bounds $a, b \in \mathbb{R}$, respectively. We are interested in categorizing elements of X into a fixed number k of ordered discrete categories $K = \{1, \ldots, k\}$. To a human user these categories would be presented as interpretable words like $\{\text{extremely low}, \text{very low}, \ldots, \text{extremely high}\}$. A **discretization** of X is a function $c : X \to \{1, \ldots, k\}$ given by $k-1$ cut-off values $c_1 < c_2 < \ldots, c_{k-1}$ through $c(x) = \min\{i : c_i \geq x\}$. An (empirical) **discretization method** maps finite samples $S \subseteq X$ to a uniquely defined discretization. As a convention we define as $S = \{s_1, \ldots, s_n\}$ with $s_i \leq s_j$ for $i < j$. Many discretization methods actually only yield a **labeling**[1] $l : S \to K$ of the given sample rather than cut-off values on the real interval. For those cases, we consider the **canonical discretization** of a labeling l as the one given by the cut-off values

$$c_i = (\max\{s \in S : l(s) = i\} + \min\{s \in S : l(s) = i+1\})/2 \ ,$$

for $i \in \{1, \ldots, k-1\}$. That is, cut-off values are defined as the arithmetic mean between the extreme values of adjacent category labels.

The first and most simple family of discretization methods that we consider are **ranged-based methods** that define cut-off values as a simple function of the underlying interval (sample-independent variant) or the range of the given data sample (sample-dependent variant). The simplest member of this family is **sample-independent equal-width discretization**, which is given by the cut-off values

$$c_i = a + i(b-a)/k$$

for $i \in \{1, \ldots, k-1\}$. For **sample-dependent equal-width discretization** the smallest and the largest sample element are used in place of the interval boundaries a and b, i.e., cut-off value i is defined as $s_1 + i(s_n - s_1)/k$. While these methods are very simple to define, depending on the given category names, both of these approaches

[1]Labelings resulting from discretization methods of course must be monotone, i.e., $l(s) \leq l(s')$ if $s \leq s'$.

(a)

(b)

Figure 3: Example populations of cups (a) and sunglasses (b) for the *prize* narrative in the study trial.

can be counter-intuitive: for example for "high", "normal", and "low" they set the normal range to be of equal size as the two extreme ranges. Therefore, for an odd number of categories $k > 2$, we define sample-dependent and sample-independent **geometric-width discretization** as alternative range-based approaches that cut the range into increasingly fine pieces when approaching the interval (or sample) borders. That is, for the sample-independent variant, the cut-off values are defined as

$$c_i = \begin{cases} b - (b-a)g_{(k-1)/2-i+2}, & \text{for } i \leq k/2 \\ a + g_{i-(k-1)/2+1}, & \text{for } k/2 < i < k \\ 1, & \text{for } i = k \end{cases}$$

with the geometric sums $g_m = \sum_{j=1}^{m} 2^{-j}$, and for the sample-independent variant, a and b are again replaced by s_1 and s_n, respectively. See Fig. 2 for an illustration.

As a second family of discretization methods we consider **frequency-based discretizations**. Those methods determine labelings based on desired counts of data values per category and are indifferent to the metric proximity between values. Technically, these labelings are most conveniently defined through the sample **quantiles** $q(\alpha) = \min\{s_i \in S : i/n \geq \alpha\}$ for $\alpha \in [0, 1]$. A sequence of fractions $\alpha_1 < \alpha_2 < \cdots < \alpha_k = 1$ gives rise to a labeling $l(s) = \min\{i \in K : p(\alpha_i) \geq s\}$. The most well-known instantiation of this scheme is **equal-frequency labeling**, which uses the set of equidistant quantiles given by $\alpha_i = i/k$ for $i \in K$. Again it can be linguistically somewhat counter-intuitive when all categories contain an equal number of sample values. For "low", "normal", and "high", this would imply that only a minority of data-values is considered "normal" and two third are either "high" or "low". To address this issues, for odd $k > 2$ we again define a variant based on increasingly refined categories (this time in terms of the quantiles), that we refer to here as **geometric-frequency labeling**. It is given by the fractions

$$\alpha_i = \begin{cases} 1 - g_{(k-1)/2-i+2}, & \text{for } i \leq k/2 \\ g_{i-(k-1)/2+1}, & \text{for } k/2 < i < k \\ 1, & \text{for } i = k \end{cases}$$

where g_m denotes the geometric sum as above.

As a final family of discretization methods we consider **clustering-based methods**. These methods determine a labeling based on a set of k reference values $R \subset X$, each of which is the representative for one of the categories. Assuming that R consists of the elements $r_1 < r_2 < \cdots < r_k$, the resulting labeling is then defined by $l(s) = i$ where r_i is a reference value that minimizes $|s - r|$ with $r \in R$ (breaking ties, e.g., by using the minimal such value). Naturally, one wants to use the set of reference values that are closest to their associated sample values. If one uses the sum of squared differences, $\sum_{s \in S}(r_{l(s)} - s)^2$, to measure this closeness, this approach yields **k-means-based labeling** (the mean of a set of values minimizes the sum of the squared distances). Since the reference values in this approach can be arbitrary elements of the underlying interval X they are susceptible to outlying sample values, which can lead to counter-intuitive discretizations. This can be addressed by using the reference values that minimize the sum of absolute errors, $\sum_{s \in S} |r_{l(s)} - s|$. Since, the sum of absolute errors of a set of values is minimized by any median value of that set, this variant is called **k-medians-based labeling**.

3. EMPIRICAL DESIGN

In this section we develop the study design (empirical method) for comparing formal discretization methods to discretization performed by humans. This includes a questionnaire for posing abstract discretization tasks to a general audience, a set of discretization tasks as re-usable test cases for the given as well as for follow-up studies, and measures for the quantification of the similarity of human and formal discretization results.

3.1 Questionnaire

The purpose of the questionnaire is to gather data from members of a general target audience on how they intuitively perform abstract linguistic discretization tasks. This measurement problem entails a central difficulty. While potential participants are used to perform intuitive discretization for concrete variables, it is likely to not work as intended to directly pose to them an abstract task about an unknown variable: it would trigger a formal approach to the problem and/or possibly yield a low engagement with the task and consequently relatively arbitrary answers.

Therefore, the key idea of our questionnaire design is to decorate the abstract tasks with concrete **narratives** of tangible variables from everyday life. The trick is that we use variables that have a value distribution which greatly depends on the specific sub-population they are defined on, and then to leave the sub-population ambiguous—with the given sample as the only means to infer it. This way the task has to be solved factually with the same information as the underlying abstract task. Of course, the given narrative might still influence the responses. It is therefore advisable to use multiple narratives so that their effects cancel out when averages over the whole result set are taken.

In our study trial, we opted for two narratives: *prices of products* and *ages of humans*. For both variables, one is used to heavily alter the usage of quantification terms across different classes of products and groups of people, respectively. For instance, even a "very low" price for a TV set is likely to be considered "high" if it were the price of a light bulb. Similarly, the age of a "young" high-school teacher would be considered "older" for a college student. The questionnaire design emphasizes this sub-population dependency by introducing the narrative with two named and labeled **example populations** that show a contrasting variable distribution. In our study trial, we used for the prize narrative the examples of cups and sunglasses (see Fig. 3) and for the age narrative the examples of members of a fencing team and

In different contexts numbers can have different interpretations. What you consider [**example category 1**] in one case, you might consider [**example category 2**], or [**example category 3**] in another. Consider the image below of a set of [**example population 1**]. The number underneath each [**population member**] shows [**variable**] in [**unit**]. The labels below the picture show an exemplary categorization of the numbers into [**category list**] that you perhaps would roughly agree to in the context of this set.

[**image of population 1 with variable labels**]
[**example cut-off points 1**]

Now compare this categorization into [**category list**] to the next categorization for [**variable**] of [**example population 2**]. Despite being different, each of the categorizations make sense in their respective set.

[**image of population 2 with variable labels**]
[**example cut-off points 2**]

Text 1: Leading text of questionnaire, which introduces narrative along with example populations.

the inhabitants of an elderly housing facility. In the questionnaire, images of the example populations are embedded into an introductory text that explains the sub-population dependence of the linguistic terms. The verbatim text-frame is given in Text 1.

Following this introductory passage, a number of actual discretization tasks is presented to the participant. In order to support the narrative, the sample values are embedded into images that depict anonymous populations for the variable. For the two narratives in our trial those images are given in Figs. 4 and 5, respectively. The tasks are introduced with the instruction text given in Text 2. Note that we do explicitly mention the possibility of choosing cut-off values that are not part of the given sample itself. This possibility can be further emphasized by using this option in the example discretizations.

In summary the proposed questionnaire design allows to pose a number of abstract linguistic discretization tasks to participants from a general population by decorating them with a concrete narrative. It is required that all tasks on one instance of the questionnaire use the same linguistic categories and that their sample ranges match the chosen narrative. This might require to rescale some of them. In the next subsection, we discuss these and other issues abounding when creating a full study design around this questionnaire.

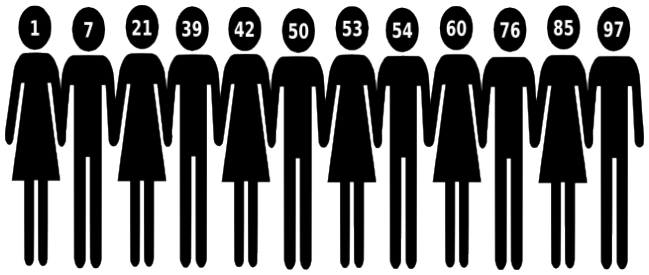


Figure 4: Task image for *age* narrative.

In this short survey, we ask for your opinion on what means [**category list**] in the context of three anonymous groups of [**population type**]. Underneath each of the images below, please fill into the designated boxes what you consider [**category list**]. Note that you can fill in numbers that do not occur in the sample itself. When you are done, please do not forget to click the submit button. Thanks a lot for your participation.

[**image of task 1**]
[**input fields for cut-off values**]
...
[**image of task z**]
[**input fields for cut-off values**]

Text 2: Instructions and task part of questionnaire.

3.2 Discretization tasks

When setting the discretization tasks for the study, there are two components that have to be defined more or less independently: the linguistic categories to be used as well as the actual numerical samples. Regarding the first component it is important to note that the validity of any results of the study, when interpreted strictly, is tied to the specific quantifiers used. Although certain insights can arguably be transferred between different category sets, it is generally unclear whether the human expectation for appropriate interval sizes varies depending on if they are called "low", "normal", and "high" or, e.g., "reduced", "moderate", and "increased". Similarly, quantifiers for specific kinds of variables, e.g., "long" for length, might carry their own expectational bias and may be not fully compatible to their generic counter-parts.

In the given instantiation of the study design we choose to focus only on categorization into

$$K = \{\text{“low”}, \text{“normal”}, \text{“high”}\} \ .$$

The rational for this choice was that these are the perhaps most widely applicable quantifiers for numerical values. Moreover, using three categories arguably constitutes a pareto-optimal choice when trading off the interpretability of the categories (individually and jointly) and their accuracy in representing the underlying numerical range.

Turning to the samples, the goal is to have a diverse set of tasks which is likely to allow to differentiate between the different discretization methods even with a relatively small number of values. On the one hand, for the aim to have a consistently high response quality it is desirable to work with small samples. The larger the sample size the more variation is likely to occur among participants in the degree to which they fully process individual sample values. On the other hand, the smaller the sample the lesser the results are likely to generalize to realistic sample sizes in Data Analysis. In particular, seven plus/minus two apparently constitutes a phase transition between the usage of different mental processing mechanisms according to the classic result of Miller [1956]. Therefore, in the given study we use the **sample size** $|S| = 12$ throughout all discretization tasks. Moreover,

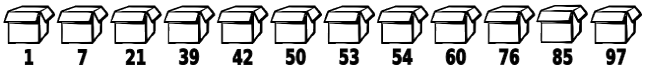


Figure 5: Task image for *prize* narrative.

Task	Class	Sample	c_1	c_2
a)	Uniform	$\{5, 18, 24\}$, $\{30, 32, 32, 50\}$, $\{70, 75, 87, 91, 95\}$	25	70
b)	Normal	$\{1, 7, 21\}$, $\{39, 42, 50, 53, 54, 60\}$, $\{76, 85, 97\}$	25	65
c)	Exponential	$\{40, 42, 42, 43\}$, $\{47, 47, 48, 48, 53\}$, $\{62, 70, 74\}$	45	60
d)	Mix	$\{4, 6, 8, 10\}$, $\{22, 23, 24, 25, 28\}$, $\{37, 49, 56\}$	11	35
e)	Uniform	$\{15, 27, 27\}$, $\{\underline{31}, \underline{35}, \underline{37}, 51, 53, 54\}$, $\{80, 81, 90\}$	30	70
f)	Normal	$\{31, 31, 35, 39\}$, $\{77, 79, 82, 82, 82\}$, $\{93, 93, 98\}$	40	90
g)	Exponential	$\{24, 24, \underline{34}\}$, $\{41, 43, 46, 49, \underline{56}\}$, $\{63, 64, 65, 81\}$	35	63
h)	Mix	$\{1, 3, 7\}$, $\{\underline{20}, 30, 37, 37, 38, 39, 44\}$, $\{55, 68\}$	18	46

Table 1: First two group of samples generated for task classes in study trial with median cut-off values of respondents. Underlined sample elements show discrepancy of resulting labeling with k-medoids.

in order to fit our narratives, we scale the **variable range** to $X = [0, 100]$ but, again in order to reduce the cognitive burden of the participants (and thus reduce variation in result quality) we work with rounded samples $S \in \{1, \ldots, 100\}^*$. In particular, we remove 0 from the sampling range in order to maintain the intuition of the price narrative.

For generating the samples, we define four **classes of discretization tasks**—uniform, normal, exponential, and mixture—based on four continuous random variables with probability density functions p_{uni}, p_{nrm}, p_{exp}, and p_{mxt}, respectively. A discretization task for a class with pdf p is then generated by drawing a sample of 12 independent realizations of the rounded and truncated version of the corresponding random variable, i.e., using the probability mass function

$$f(n) = \int_{n-1}^{n} p(x)/Z \, dx$$

for $n \in \{1, \ldots, 100\}$ with $Z = \int_0^{100} p(x) \, dx$. The formal definitions of the continuous pdfs are as follows.

Uniform is simply defined by the uniform pdf $p_{\text{uni}}(x) = 1$ for $x \in [0, 100]$. For a task from this class we do not expect significant value clusters to appear. Hence, there might be a tendency among participants to resort to simpler principles than clustering-based discretization.

Normal is defined through $p_{\text{nrm}}(x) = \phi_{M,S}(x)$, i.e., the Gaussian pdf with uniform random mean M and standard deviation S drawn from $[0, 100]$. Tasks from this class are likely to show a central tendency towards a random mean. Hence, it can be expected that human assignment of "normal" will reflect that tendency in contrast to the sample independent range-based discretization methods.

Exponential is defined by $p(x) = O + R \exp(-Rx)$ with a uniform random offset term O from $[0, 50]$ and a uniform random rate parameter R from $[0.2, 0.8]$. Tasks from this class are expected to have a highly skewed distribution, which should render symmetric range-based discretizations counter-intuitive.

Mixture is defined by $p(x) = \phi_{M_1,S_1}(x) + \phi_{M_2,S_2}(x)$ as the mixture of two Gaussians with uniform random means and standard deviations as defined for the class normal. Samples from this class are expected to be bi-modal with a high, a low, and normal range around each mode. This is generally hard to reflect adequately with three categories only, but it is to be expected that count-based and clustering-based approaches can find reasonably intuitive compromises.

3.3 Evaluation measures

After designing the test discretization tasks as well as a questionnaire for querying human solutions to these tasks, it remains to define how we want to compare the discretizations produced by formal methods with those of the study participants. We will do this on two levels of resolution: on the first, we just compare the discretizations in terms of how they label the given sample; on the second, we measure the difference of the actual cut-off points.

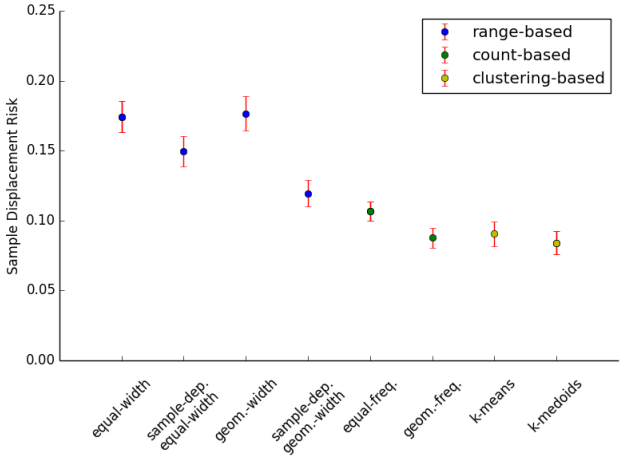


Figure 6: Sample displacement risk per method over all tasks in study with 95%-confidence intervals.

Let c and d be two discretizations of the range $X = [a, b]$ using the categories $K = \{1, \ldots, k\}$ and $S \in X^*$ be a finite sample of X. Independent of whether we want to quantify the difference between c and d through their cut-off values on the whole range X or just in terms of how they label the elements of S, we first have to fix the **displacement loss** between two categories, i.e., how much we consider it harmful to use a category j in place of the true category i. For that we propose to use the relative difference of the category numbers $l(i, j) = |i - j|/(k - 1)$ (we normalize here with $k - 1$ rather than k so that l reduces to the 0/1-loss when $k = 2$).

When evaluated on all category pairs abounding from applying c and d to the sample S, this loss function leads to the **sample displacement loss** for discretizations defined by

$$l_S(c, d) = \frac{1}{|S|} \sum_{x \in S} l(c(x), d(x)) \ .$$

As described above, this measure considers the categoriza-

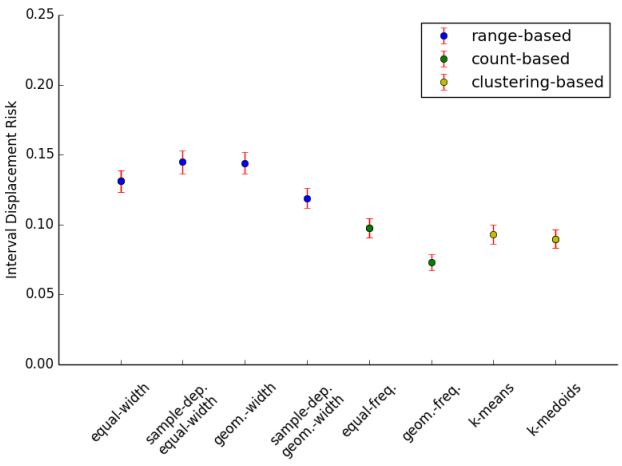


Figure 7: Interval displacement risk per method over all tasks in study with 95%-confidence intervals.

tions as mere labelings of the data sample and, beyond that, does not reflect how the discretizations differ when viewed as linguistic categorizations of the underlying domain. For that purpose we define the **interval displacement loss**

$$l_X(c,d) = \frac{1}{b-a}\int_{x\in X} l(c(x),d(x))\,dx$$

which quantifies disagreement between the discretizations in terms of the size of the underlying domain-pieces with a certain displacement. This integral can be simply computed as the sum of the piece-wise constant losses on the intervals resulting from cutting X with all cut-off values in c and d.

Now assume we have a bag of study results R consisting of pairs of samples and discretizations $\{(S_1,d_1),\ldots,(S_m,d_m)\}$ where discretization d_i is the result of a respondent for the task involving sample S_i. For a discretization method $m : X^* \to (X \to K)$ we can then determine its **empirical sample displacement risk** as

$$r_R = \frac{1}{R}\sum_{(S,d)\in R} l_S(m(S),d)\ .$$

Given that the set of respondents and the set of tasks is representative for some larger population of tasks and users of interest, this empirical risk will approximate the real population displacement risk (defined through the expected loss-value over this population) for the method m. Switching to the interval displacement loss, we can define similarly the empirical interval displacement risk based on a set of study results as well as the underlying population displacement risk.

4. RESULTS

In this section we report the results of an open online study trial[2] using the design developed in Sec. 3. The trial was conducted over the course of 10 days with a total of 153 respondents. We advertised the study through a call for participation that was published via internal mailing lists of 6 academic institutions from 4 different countries (UK, Germany, Finland, and Israel) as well as through social media in 3 different networks (Facebook, LinkedIn, and Google+). The call encouraged participants to re-share the invitation for participation with potentially interested colleagues and friends. Hence it was a convenience- and snow-ball sampling scheme of participants with the goal to maximize the number of responses—sacrificing control over the participant demographics.

For the same purpose and also to ensure an as high as possible quality of responses, we intended to keep the expected time and attention for participation low. Hence, we settled to issue only 3 discretization tasks per participant. Moreover, since this was the first study of this kind, we wanted to be able to meaningfully inspect the results, in particular, the chosen cut-off values for each of the generated tasks. For that reason we opted for a task sampling scheme that generates a certain number of repetitions per sample (for each narrative). In more detail, we iteratively fixed groups of 4 random tasks (one per task class). Three tasks of a group were then issued to each requested questionnaire uniformly at random (decorated by a random narrative) until each task in the group (for each narrative) received at least 25 responses. Only then a new group was generated. Thus, we traded off representativeness for the individual task classes for an attempt to acquire confident estimates of preferred cut-off values per sample. See Tab. 1 for a list of all samples for which the full number of responses was reached along with the median respondents' cut-off values.

4.1 Overall Outcome

Aggregating over all trial results, i.e, all responses for all discretization problems, the following picture emerges for the empirical sample displacement risk (see Fig. 6 and also Tab. 2, upper portion, row "all"). There are three methods leading the field: k-medoids-based labeling (risk[3] of apprixmately 0.0839 ± 0.0083), geometric-frequency labeling (0.0876 ± 0.0073), and k-means-based labeling (0.0904 ± 0.0089). While the results do suffice to confidently separate this group from the rest of the methods, they are insufficient to confidently separate them from one another. The next group consists of equal-frequency labeling (0.1067 ± 0.0069) and sample-dependent geometric-width labeling (0.1193 ± 0.0095). The remaining range-based methods are at the end of the spectrum with a small but significant advantage for sample-dependent equal-width (0.1494 ± 0.0109).

Turning to the interval displacement risk (see Fig. 7 and Tab. 2, lower portion, row "all"), the first observation is that the magnitude of empirical loss values is somewhat and their variation is notable smaller as for the sample displacement risk. Consequently we have smaller confidence intervals. The ranking of the methods are slightly shifted with geometric-frequency labeling (0.0729 ± 0.0058) now leading confidently in front of the following group consisting of k-medoids-based labeling (0.0898 ± 0.0063), k-means-based labeling (0.0931 ± 0.007), and labeling based on equal-frequency (0.0974 ± 0.0067). At the end of the field we have again the range-based methods. Out of those methods, just as with the sample displacement risk, sample-dependent geometric-width performs best. However, for the interval displacement

[2]All results can be downloaded from http://www.realkd.org/wp-content/uploads/2016/05/discretization-study-results.csv

[3]We give all risks here rounded to 4 digits with $\alpha = 0.95$ two-sided confidence intervals.

	si equal-width	sd equal-width	si geom.-width	sd geom.-width	equal-freq.	geom.-freq.	k-means	k-medoids
sample displacement risk								
normal	0.1439 ± .0225	0.1150 ± .0188	0.1791 ± .0282	0.1150 ± .0188	0.1129 ± .0137	0.0916 ± .0177	**0.0789 ± .0185**	**0.0789 ± .0185**
exponential	0.1684 ± .0214	0.2419 ± .0168	0.2097 ± .0252	0.2216 ± .0152	0.1201 ± .0177	0.1019 ± .0156	**0.0875 ± .0169**	0.1078 ± .0154
uniform	0.1342 ± .0151	0.1988 ± .0170	**0.0841 ± .0129**	0.0844 ± .0121	0.1131 ± .0108	0.0910 ± .0126	0.1571 ± .0160	0.1074 ± .0161
mixture	0.2515 ± .0213	0.0413 ± .0124	0.2279 ± .0181	0.0498 ± .0111	0.0796 ± .0093	0.0649 ± .0091	0.0413 ± .0124	**0.0410 ± .0124**
all	0.1742 ± .0110	0.1494 ± .0109	0.1765 ± .0123	0.1193 ± .0095	0.1066 ± .0069	0.0876 ± .0073	0.0904 ± .0089	**0.0839 ± .0083**
interval displacement risk								
normal	0.1254 ± .0125	0.1788 ± .0184	0.1240 ± .0132	0.1509 ± .0161	0.1410 ± .0162	**0.0898 ± .0147**	0.1160 ± .0168	0.1160 ± .0168
exponential	0.0792 ± .0102	0.1927 ± .0145	0.1427 ± .0126	0.1719 ± .0099	0.0730 ± .0114	0.0559 ± .0103	**0.0557 ± .0105**	0.0718 ± .0093
uniform	0.0941 ± .0110	0.1340 ± .0113	**0.0800 ± .0113**	0.0892 ± .0108	0.0954 ± .0121	0.0876 ± .0119	0.1420 ± .0114	0.1104 ± .0106
mixture	0.2272 ± .0110	0.0680 ± .0073	0.2280 ± .0108	0.0576 ± .0066	0.0782 ± .0080	**0.0585 ± .0061**	0.0607 ± .0081	0.0607 ± .0081
all	0.1310 ± .0077	0.1448 ± .0082	0.1439 ± .0078	0.1189 ± .0072	0.0974 ± .0067	**0.0729 ± .0058**	0.0930 ± .0070	0.0898 ± .0063

Table 2: Empirical sample displacement and interval displacement risks with 95% confidence intervals—taken over all tasks and per task class. All numbers are rounded to 4th digit after decimal point.

risk, its confidence interval has a slight overlap with the sample-independent equal-width method.

4.2 Outcome per task class

When looking at the results per task class (see Tab. 2), one can make some notable observations specifically when looking at the performance of the clustering-based methods. Both, k-means and k-medoids, are the best or among the best methods in all task classes but *uniform*. Here, k-means is the second worst with respect to sample displacement risk and the worst with respect to interval displacement risk. Notably, k-medoids performs more robust for this task class, while its ranks (4 and 6, respectively) also deviate substantially from the ranks it achieves for the other classes. In contrast to clustering-based methods, range-based discretization in the form of sample-independent and dependent geometric-width, are specifically strong for the uniform tasks. They are also competitive for mixture but perform both weakly for normal as well exponential.

Finally, we can observe that geometric-frequency performs consistently well across all problem types independent of the risk functional. In fact, for the interval displacement risk it performs best or at least not significantly worse than the best for all classes. Interestingly, when looking at the median cut-off values for all individual samples for which a large number of responses was generated (Tab. 1), we can see that there is only one sample (uniform sample 'k') for which the labeling of the median respondents' cut-off points disagrees substantially with k-medoids and two more (samples 'g' and 'h') where there is a minor disagreement. In all these cases, the respondents' median labeling has category frequencies closer to the geometric frequencies (0.25, 0.5, 0.25) than the solution of k-medoids (there is a similar trend for 'a' and 'g', where the median exactly respects the k-medoids objective, but there is still a substantial number of respondents who did not adhere to it and produced category proportions closer to the geometric frequencies).

5. CONCLUSIONS AND OUTLOOK

With the presented study design we were able to gather for the first time insights on the human expectation for discretizing numerical data into the discrete categories: "low", "normal", and "high", which are important categories, e.g., for providing a simple intuitive discretization in data analysis suites. Particular findings are:

1. Clustering-based methods appear to yield good results essentially confirming the proposition from Cognitive Psychology and Cognitive Linguistics, which says that humans tend to perform categorization based on similarity to category prototypes. It seems, however, that this mechanism alone is not enough to fully replicate human discretization choices for quantitative linguistic categories (such as "low", "normal", and "high'). In our trial we could observe a tendency to sometimes deviate from optimal clustering-based solutions, presumably, in order to create more satisfying category frequencies.

2. Particularly, for the chosen linguistic categories, a frequency of 0.5 for the "normal" category, and a frequency of 0.25 for the categories "low" and "high" each appear to be attractive. This is testified by the fact that the frequency-based method with these parameters performed robustly well across different tasks.

3. Ranged-based methods that disregard the sample (or almost disregard it except for the extreme values) appear to be too simplistic for robustly creating an intuitive labeling and can not compete with the other method classes. Hence, while the relative differences of metric attributes seem to have a notable effect on labeling decisions, the absolute scale of metric information seems to be at most of minor importance.

Generally, we hope that the given work will open the door for systematically deriving novel approaches for intuitive discretization and evaluating them with the proposed or a modified study design. Some open questions we consider to be of particular importance are the following.

1. To what degree are the trends we discovered representative for the underlying problem classes and for specific target audiences. In the performed trial, representativeness for task classes has been sacrificed for more representativeness of the individual tasks, and the population was mostly uncontrolled.

2. What is an intuitive mechanism for deriving precise cut-off points from a given labelling? The given trial data showed no clear trend of how human cut-off values relate to their labelings.

3. Finally, the perhaps most interesting direction for future research is to investigate to what degree the identified trends hold up for finer categorizations, e.g., into "very low", "low", etc. and larger samples per task. A particular question for the frequency-based methods is whether the geometric proportions are really the expected continuation of the 0.25/0.5/0.25 scheme.

References

Martin Atzmueller. Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49, 2015.

Martin Atzmueller and Florian Lemmerich. Vikamine–open-source subgroup discovery, pattern mining, and analytics. In *Machine Learning and Knowledge Discovery in Databases*, pages 842–845. Springer, 2012.

Mario Boley, Maike Krause-Traudes, Bo Kang, and Björn Jacobs. Creedo—scalable and repeatable extrinsic evaluation for pattern discovery systems by online user studies. In *IDEA 2015*, 2015.

François Chapeau-Blondeau and David Rousseau. The minimum description length principle for probability density estimation by regular histograms. *Physica A: Statistical Mechanics and its Applications*, 388(18):3969–3984, 2009.

Stanislas Dehaene, Ghislaine Dehaene-Lambertz, and Laurent Cohen. Abstract representations of numbers in the animal and human brain. *Trends in neurosciences*, 21(8): 355–361, 1998.

Stanislas Dehaene, Véronique Izard, Elizabeth Spelke, and Pierre Pica. Log or linear? distinct intuitions of the number scale in western and amazonian indigene cultures. *Science*, 320(5880):1217–1220, 2008.

Vyvyan Evans. *A glossary of cognitive linguistics*, volume 251. Edinburgh University Press, 2007.

Floris Geerts, Bart Goethals, and Taneli Mielikäinen. Tiling databases. In *Discovery science*, pages 278–289. Springer, 2004.

Bart Goethals, Sandy Moens, and Jilles Vreeken. Mime: a framework for interactive visual pattern mining. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 757–760. ACM, 2011.

Hisao Ishibuchi, Tomoharu Nakashima, and Manabu Nii. *Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining*. Springer Science & Business Media, 2006.

Petri Kontkanen and Petri Myllymäki. Mdl histogram density estimation. In *International Conference on Artificial Intelligence and Statistics*, pages 219–226, 2007.

Marvin Meeng and Arno Knobbe. Flexible enrichment with cortana–software demo. In *Proc. Benelearn*, pages 117–119, 2011.

George A Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

Hoang-Vu Nguyen, Emmanuel Müller, Jilles Vreeken, and Klemens Böhm. Unsupervised interaction-preserving discretization of multivariate data. *Data Mining and Knowledge Discovery*, 28(5-6):1366–1397, 2014.

Laxmi Parida and Naren Ramakrishnan. Redescription mining: Structure theory and algorithms. In *AAAI*, volume 5, pages 837–844, 2005.

Eleanor H Rosch. Natural categories. *Cognitive psychology*, 4(3):328–350, 1973.

John R Taylor. *Linguistic categorization*. OUP Oxford, 2003.

Jilles Vreeken, Matthijs Van Leeuwen, and Arno Siebes. Krimp: mining itemsets that compress. *Data Mining and Knowledge Discovery*, 23(1):169–214, 2011.

Rudolf Wille. Formal concept analysis as mathematical theory of concepts and concept hierarchies. In *Formal Concept Analysis*, pages 1–33. Springer, 2005.

ReVACNN: Real-Time Visual Analytics for Convolutional Neural Network

Sunghyo Chung
Korea University
sunghyo.chung@gmail.com

Sangho Suh
Korea University
sh31659@gmail.com

Cheonbok Park
Korea University
chunbok94@gmail.com

Kyeongpil Kang
Korea University
rudvlf0413@korea.ac.kr

Jaegul Choo
Korea University
jchoo@korea.ac.kr

Bum Chul Kwon
IBM T.J. Watson Research
bumchul.kwon@us.ibm.com

ABSTRACT

Recently, deep learning has gained exceptional popularity due to its outstanding performances in many machine learning and artificial intelligence applications. Among various deep learning models, convolutional neural network (CNN) is one of the representative models that solved various complex tasks in computer vision since AlexNet, a widely-used CNN model, has won the ImageNet challenge[1] in 2012. Even with such a remarkable success, the issue of how it handles the underlying complexity of data so well has not been thoroughly investigated, while much effort was concentrated on pushing its performance to a new limit. Therefore, the current status of its increasing popularity and attention for various applications from both academia and industries is demanding a clearer and more detailed exposition of their inner workings. To this end, we introduce ReVACNN, an interactive visualization system that makes two major contributions: 1) a network visualization module for monitoring the underlying process of a convolutional neural network using a filter-level 2D embedding view and 2) an interactive module that enables real-time steering of a model. We present several use cases demonstrating benefits users can gain from our approach.

Categories and Subject Descriptors

H.2.8 [**Data Visualization**]: Convolutional Neural Network; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Design, Performance

Keywords

Deep Learning; Visualization; Convolutional Neural Network

[1] http://www.image-net.org/challenges/LSVRC/

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16) August 14th, 2016, San Francisco, CA, USA.

1. INTRODUCTION

Recently, deep learning has made major breakthroughs in many machine learning problems such as computer vision [6] and speech recognition [4]. A traditional neural network model is basically composed of multiple layers, each of which contains multiple nodes where each node is computed as a linear combination of nodes in the previous layer, followed by a nonlinear transformation such as a sigmoid, a tanh, a softmax function. However, neural network has not been widely used until recently since it was difficult to train due to the significant computing time, its sensitivity to initialization and hyper-parameters, and other issues. Various treatments have been proposed including dropout [9], batch normalization [5], and alternative nonlinear functions such as a rectified linear unit [8], which successfully handled most of the existing issues.

Beyond the traditional model, the neural network structure has evolved in various forms, leading to tremendous success in important applications. Largely responsible for this recent success is convolutional neural network (CNN), a type of neural network suited for real-world image classification tasks. Although convolutional neural networks have been originally proposed by LeCun et al. [7] back in the early 1990s, demonstrating an outstanding performance in hand-written digit recognition, it was not widely used until 2012 when Krizhevsky et al. [6] achieved a superior performance on image classification tasks in ImageNet challenge, using a deep architecture model of convolutional neural network. This propelled major research movement towards creating variants in architectures and improving algorithms for even higher performance. In just few years, much progress has been made to the point of approaching or even surpassing human abilities in various challenging tasks.

While making significant achievements, the understanding of underlying processes in these models received less examination, and the need for tools and techniques for exploring and understanding the inner workings of these various models ensued. However, complicated deep learning structures are difficult to understand. Different types of layers such as convolution, pooling, and fully-connected layers interact with each other, handling different parts of data characteristics. Furthermore, each layer has different sets of hyper-parameters to determine before training the model. Thus, such a model selection process including setting the number of layers and nodes, and hyper-parameter values has not been intuitive nor straightforward, leaving users with no idea about how to properly perform this process.

In addition, the significant amount of time required to train a deep learning model has made the training process largely detached

from dynamic user intervention. For example, a recently proposed model called ResNet [3], which is considered one of the state-of-the-art models, takes days to weeks to train using ImageNet datasets with the fastest graphics processing unit available.

In response, we propose a proof-of-concept visual analytic system for allowing users to understand and steer a deep learning model in real time during the training process. To the best of our knowledge, our system is one of the first systems that visualize detailed information about the model during the training process and support dynamic user interactions with the model in real time.

In particular, the main contributions of this paper are summarized as follows:

- Real-time visualization of how each node/filter in a deep learning model is trained, e.g., the stability of nodes/filters and the relationships between them
- Real-time model steering by dynamically adding/removing nodes and layers during the training process

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 presents detailed description of our system and its visual components. Section 4 presents usage scenarios. Finally, Section 5 concludes our discussion with plans for future work.

2. RELATED WORK

In this section, we discuss recent efforts towards interactive visualization of deep learning for its in-depth understanding and user control.

Bruckner et al. [1] developed the system called deepViz, an interactive visualization based on the timeline framework that shows the heatmap representations of filters in each layer, the confusion matrix, and the clustered images at different checkpoints for understanding and diagnosing the network. Zeiler and Fergus [12] showed the practical application of a visualization system for the diagnostic purpose by utilizing a feature inversion technique called deconvolution to refine the model and further improve performances. With the system that visualizes live activations in real time and features at each layer, Yosinski et al. [11] made contributions to the visualization of convolutional neural network by providing several new regularization methods that produce qualitatively clearer visualization of images.

More on the interactive visualization side, a web-based implementation, such as ConvNetJS [2], made training convolutional neural network possible in a browser using the Javascript library. In addition, Bolei et al. [13] developed another web-based interface where a user can select an activation of a particular data item at a particular layer and check the highly activated nodes together in the other layers.[3] Harley et al. [2] visualized a convolutional neural network in a three-dimensional space where the network structure and the used images are shown simultaneously. Additionally, Google's TensorFlow library provides a graphical user interface called TensorBoard[4], which visualizes neural network as a computational graph where users can check the status of the trained model and change the detailed configurations. More recently, Google also made a web interface called TensorFlow Playground[5] publicly available so that users can play with neural network models using several toy data sets. On the other hand, NVIDIA developed its own deep learning library and a web-based monitoring system called DIGITS.[6]

Even with these various efforts, there exist significant room to improve the interactive visualization aspects of deep learning models along with the recent advancement in this area. Among them, real-time monitoring and steering of deep learning has not been properly addressed, which is the focus of our system proposed in this paper.

3. REVACNN: REAL-TIME VISUAL ANALYTICS FOR CONVOLUTIONAL NEURAL NETWORK

To empower users to dynamically monitor and interact with a convolutional neural network in real time during its training stage, we propose ReVACNN. In this section, we present (1) the system overview, (2) the visualization modules for real-time monitoring and steering of the model, and (3) the implementation details of the proposed system. The front-end of our web-based system is implemented using HTML, CSS, and Bootstrap. D3.js[7] is used for animating filters ('jittering') in the diagram. All the computations are currently performed with Javascript in the browser on the client side.

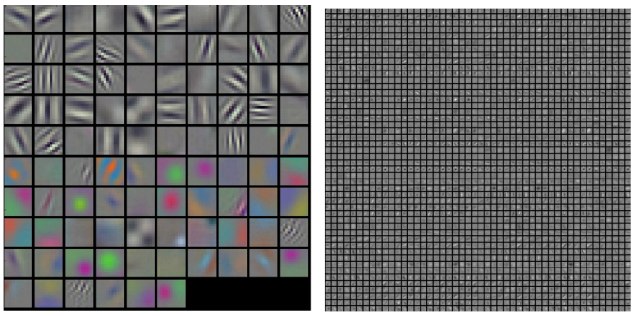
3.1 System Overview

The main goal of our system is to provide real-time steering capabilities in an easy-to-use manner. To this end, we decided to build our proof-of-concept system based on Javascript-based deep learning library called ConvNetJS.[8] In contrast to other major deep learning libraries such as Theano, Tensorflow, Torch, and Caffe, ConvNetJS runs completely in an easily accessible web browser on the client side, which is appropriate in visualizing dynamic changes of deep learning models and responding immediately to user interactions in real time. Note, however, that in exchange of such ease of use and real-time interactivity, ConvNetJS that our system uses lacks the GPU-based acceleration of computations that most of the other major libraries offer. In the scope of the current paper, we mainly investigate the real-time visualization capabilities for viewing and steering the deep learning process, using ConvNetJS as an example. We leave the topic of integrating other libraries with our real-time visual analytics system as future work.

In fact, our visual interface is built upon the implementation of CIFAR-10 demo using ConvNetJS,[9] as shown in Fig. 4, which uses the CIFAR-10 dataset[10] for object recognition. In addition, we developed additional capabilities of monitoring and steering the CNN model in real time. In summary, the main functionalities of ReVACNN we added are as follows:

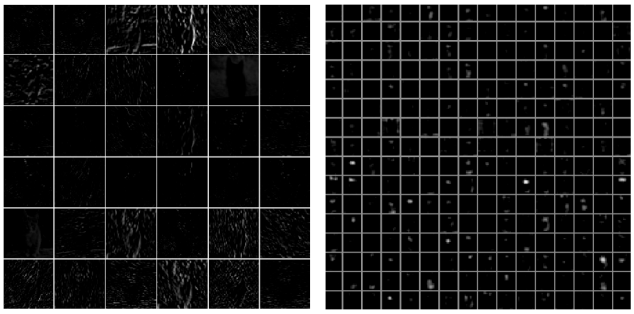
- **Network visualization and configuration view**. This module provides users with a visual overview of the network and more importantly, the ability to monitor the dynamic training process in real time. Moreover, users can modify the network dynamically and incrementally, adding or removing nodes and layers with simple "point-and-click" interactions.

[2] http://cs.stanford.edu/people/karpathy/convnetjs/
[3] http://people.csail.mit.edu/torralba/research/drawCNN/drawNet.html
[4] https://www.tensorflow.org/versions/r0.8/how_tos/graph_viz/index.html
[5] http://playground.tensorflow.org/
[6] https://developer.nvidia.com/digits
[7] https://d3js.org/.
[8] https://github.com/karpathy/convnetjs
[9] http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html
[10] https://www.cs.toronto.edu/~kriz/cifar.html



(a) Filters in conv1 layer (b) Filters in conv2 layer

Figure 1: Filter coefficients in AlexNet



(a) 1st convolution layer (b) 5th convolution layer

Figure 2: Activation maps in AlexNet [6]

- **Filter-level 2D embedding view**. This view shows the relationships of individual filters/nodes at a particular layer as a 2D embedding view. To generate such a view, we utilize a t-distributed stochastic neighbor embedding (t-SNE) [10]. Using the vector representation of each filter, we allow users to flexibly choose one among its filter coefficients, filter gradients, its activation map of a particular image, and its activation gradient maps so that users can explore various aspects of filters that are being trained.

The details of the above functionalities our system provides will be further discussed in Section 3.2.

3.2 Network Visualization and Configuration

To facilitate the understanding of how each node/layer has been trained, it is important that we directly interpret the filter coefficients, which correspond to the linear combination of coefficients or weights assigned to different positions of pixels in an image. In addition, given an image, an activation map corresponding to a particular filter gives another insight from the perspective of how much the filter gets activated depending on the image. For example, Fig. 1 shows filter coefficients found in AlexNet. Since the first-layer weights, as shown in Fig. 1a(a), are the filters directly looking at the raw pixel data of an input image, their images are often the most interpretable among the filters from all the other layers. However, as layers deepen, their meaningful interpretation becomes increasingly challenging. Fig. 1b(b) shows that the filters found in the second convolution layer are too complex and vague to be informative. Normally, an analysis of the first-layer weights can help users recognize whether the network has been successfully trained. Users can assess the success of training based on whether trained filters have smooth transitions among them so that they can capture as diverse patterns as possible. However, since such analysis relies on subjective judgment, it is clear that users need additional evidences to decide whether such argument is reasonable.

Additionally, Fig. 2 shows the activation maps found in the first and fifth convolution layers. Clearly, it is difficult to make sense of the activation maps in deeper layers because they are representing a composite mixture of already complex patterns. Also, these activation maps are shown to be relatively sparse, which means that the majority of pixels in these activation maps are mostly zero given an input image. This indicates that they hardly get activated, which may not be helpful in generating useful information. Nevertheless, identifying those filters bearing these characteristics by just looking at these activation maps is a difficult task. Our visualizations approach helps users handle the task more easily.

3.2.1 Network visualization

As shown in Fig. 3, our network visualization module provides users with a quick overview of the model. In addition, users can gain insight from the dynamic evolution of the network during the training process. In particular, among its various parameters representing dynamic evolution of the network, our system highlights how stable or converged each node is during the algorithm iterations in the form of jittering animation of nodes. That is, those nodes with a large amount of movements in their jittering animations indicate that they are being actively trained at a given moment. The quantitative value to determine this amount is computed as the magnitude of an average gradient back-propagated per each filter coefficient in the corresponding node.

In addition, the path connecting two layers shows how input images are being forward-propagated through the network layers. That is, the thickness of a path corresponds to the sum of pixel values on a particular filter in the corresponding layer. Note that only those whose values belong to the top 50% are configured to be visible in order to avoid a visual clutter in the visualization module. This path visualization has additional benefit of helping users identify how convolution layers are outputting filtered images and most importantly, which path influences the softmax layer responsible for classifying an input image. From this visualization module, users can also easily add or delete filters in the hidden layer with simple "point-and-click" interactions, and the change in the model is reflected in real time. The interactive feature helps to steer the training process of the model.

3.2.2 Training statistics visualization

During training, the loss function serves as a clue for identifying whether the network is properly trained. Thus, our module, shown in Fig. 4, displays the training loss as a line chart. Users can keep track of the temporal progress of the loss function. Since the batch size is set as four by default, the loss is plotted each time as the average training loss of a batch of four input images. If the batch size is modified by users, the loss chart also changes accordingly. In addition, other statistics, such as training accuracy and validation accuracy, are updated for each input image and shown to users for an in-depth analysis. Other hyper-parameters such as the learning rate, the momentum, the batch size, and the weight decay, can be modified. To facilitate the understanding of how each node/layer has been trained, it is important that we directly enables users to observe changes in the training accuracy immediately. Using the capability, users can properly adjust learning rates, batch sizes, and momentum values when the network is stuck in an undesirable local minimum.

3.2.3 Filter-level 2D embedding visualization

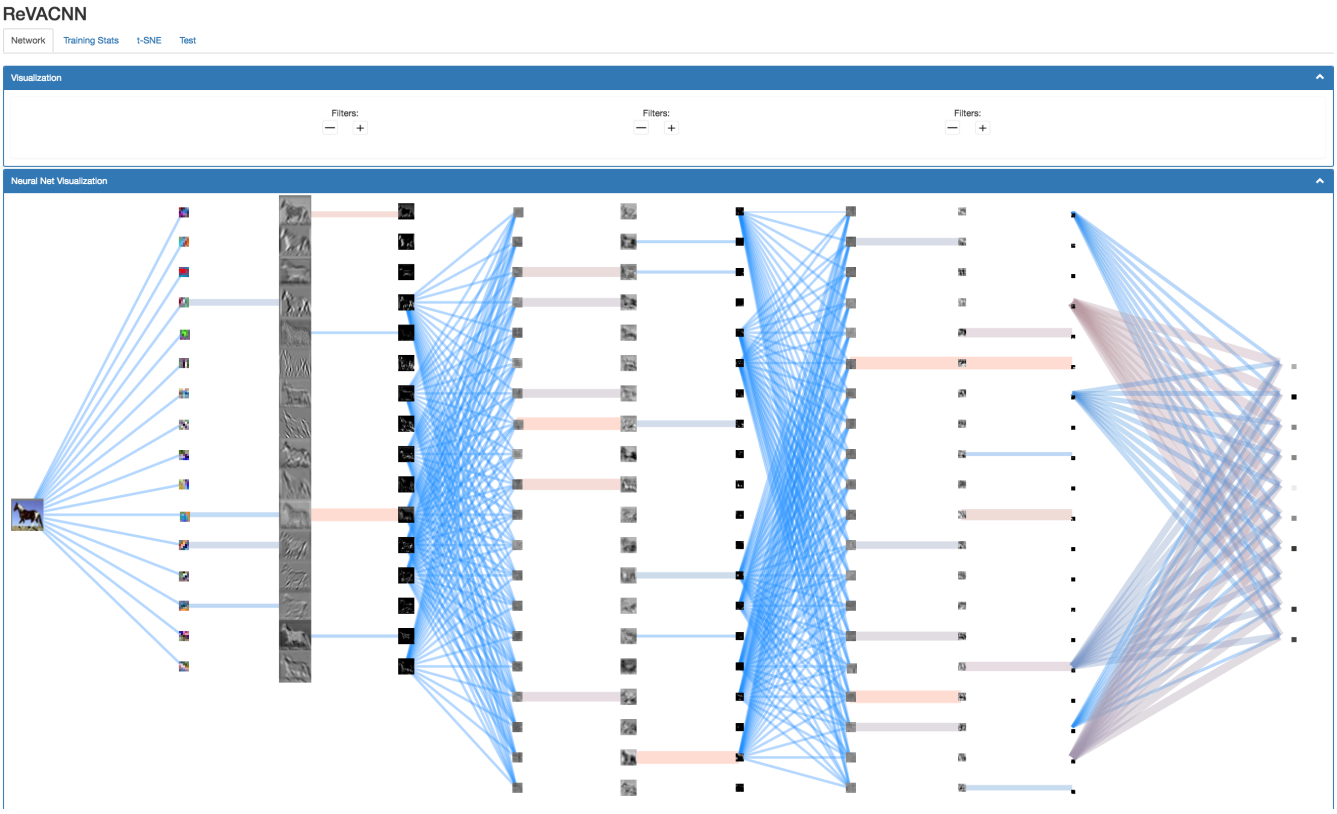


Figure 3: Network visualization of ReVACNN

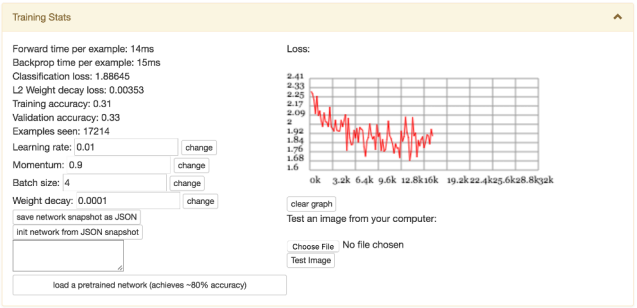


Figure 4: Training statistics view of ReVACNN

As described above, the filter coefficients and the activation maps have frequently been the main subject of visualization when analyzing a convolutional neural network. In our system, we explore them using their 2D embedding view computed by t-SNE. As shown in the left side of Fig. 5, users can open up each layer panel and observe the 2D embedding view of filter coefficients, filter gradients, its activation maps, and the activation gradients at the corresponding layer by clicking the radio button in the left pane. When users change the network architecture and initiate training, the left pane changes accordingly to reflect the model's layer configuration. This t-SNE view of the system provides users with the capabilities of node-level as well as layer-level exploration. In the case of node-level exploration, users can view the similarity between filter coefficients and activation maps or even activation gradient maps in the selected layer. In the case of layer-level exploration, the comparison of clusters of filter coefficients, the activation maps, the activation gradient maps between different layers can reveal insights for understanding of the model and further diagnosis. The usage scenarios demonstrating the usefulness of this view will be discussed in detail in Section 4.

4. RESULTS

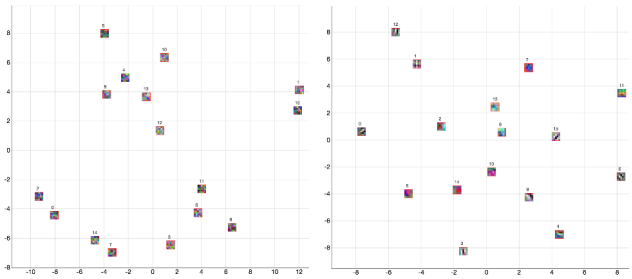
In this section, we present two use cases demonstrating the advantage of our system to monitor and steer the deep learning model in real time.

4.1 Real-Time Dynamic Model Configuration

In this section, we present four use cases where we can reveal various insights from our filter-level 2D embedding view in which a user can extract valuable insights about the model. In these use cases, we used a CNN model, which has an input layer that takes in $32 \times 32 \times 3$ input images and three convolutional layers with a filter size of 5×5 with the stride size of 1 and padding of 2 where the three convolutional layers–each of which has 16, 20, and 20 filters, respectively–have both ReLU and pooling layers behind each convolutional layer, finally followed by a softmax layer with ten classes as the last layer of our network.

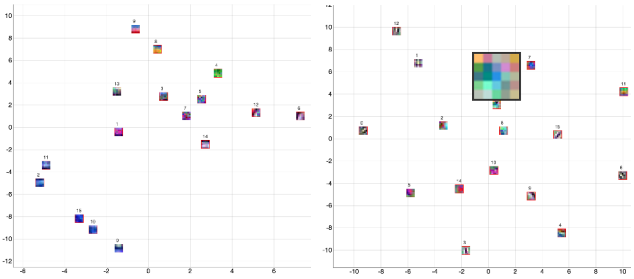
Cluster patterns. In general, neural network and deep learning models are sensitive to initialization, hyper-parameters, and other settings. Thus it is difficult to properly train the model so that it performs reasonably well even for the training data. Our filter-level 2D embedding view provides important insights about the characteristics of a properly trained model. While training the above-specified model, we checked the 2D embedding of filter coefficients at 30

Figure 5: 2D embedding view of ReVACNN



(a) Clear cluster pattern (b) No cluster patterns

Figure 6: Comparison of cluster patterns of filter coefficients



(a) Improperly trained filters (b) Properly trained filters

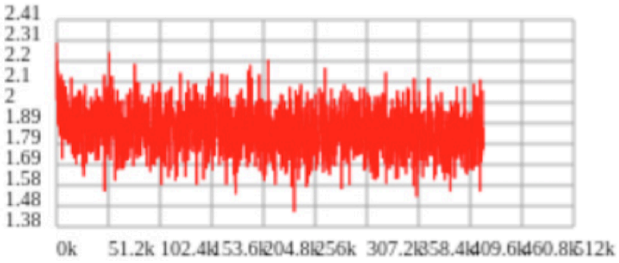
Figure 7: RGB patterns of the first-layer filters

epochs where the accuracy was quite low, e.g., 0.32. In this case, the 2D embedding view of filter coefficients shows a clustered pattern among these filters, as shown in Fig. 6a(a). This indicates that those filters belonging to a particular cluster capture somewhat redundant patterns from input data. In other words, they are not trained well enough to extract diverse patterns from the training data. On the other hand, at 120 epochs where the accuracy reaches 0.78, the 2D embedding view of filter coefficients exhibits somewhat evenly distributed filters with no clear cluster patterns, as shown in Fig. 6b(b). This example indicates an important characteristics of a well-trained model that the diversity of trained filters is generally desirable in achieving a greater classification accuracy.

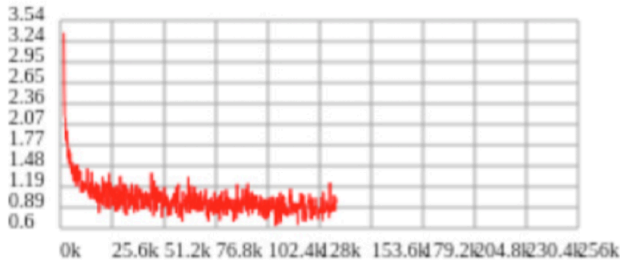
RGB patterns. The first convolutional layer takes an input image that has the depth of three RGB channels and generates each filter that linearly combines all the three channels. Similar to the previous case, when the model is not properly trained, i.e., showing a low accuracy value, we found that a trained filter often reflects only a single color channel as opposed to a combination of all the three color channels, as seen from all-blue colored filters in Fig. 7a(a). However, as seen in Fig. 7b(b), when the model shows a relatively good performance, each filter usually combines the information from all the color channels. Based on such different RGB patterns, one can infer that those filters that combine all the channels of the previous activation maps contribute to improving the generalization ability of the trained model.

4.2 Steering

In this use case, we set up our model as follows: three convolutional layers followed by ReLU and pooling layers after each convolution layer; 20 filters in each convolution layer (based on this property, we call this model a '20-20-20' model), with the filter size of 5×5; a fully-connected layer as the last layer. We trained a '20-20-21' model and found that this model does not train well, and its loss function value does not go below 1.48, as illustrated in Fig. 8a(a). On the other hand, we initially trained a '20-20-20' model, which converged relatively fast and showed a much better loss function value well below 1.19. Utilizing our interaction



(a) 20-20-21 model trained from scratch



(b) 20-20-21 model generated after filter addition

Figure 8: Effects of dynamic filter addition

capability of dynamically adding a node during training in the network visualization module, we added a filter after 15 epochs as the loss function graph reached a plateau. Accordingly, the model was dynamically changed from a '20-20-20' model into a '20-20-21' model while maintaining the currently trained model except for the added filter. Finally, the resulting model still maintained a relatively good loss function value, even experiencing a slight increase in accuracy at times. Without this dynamic network configuration process, the '20-20-21' model, which is our target model, would be more difficult to train from scratch. This shows the importance and the value of dynamic network configuration in real time during the training process.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed ReVACNN, a real-time visual analytics system for a convolutional neural network. It supports exploring and steering the network by visualizing its layers and nodes. Additionally, we provided a filter-level 2D embedding view by applying t-SNE to various filter information, such as filter coefficients, filter gradients, the activation maps, and the activation gradients. Through these capabilities offered by our system, one can obtain in-depth information such as whether the network is trained properly or not as well as other insights about the trained filters. By using such information, one can flexibly steer the model and achieve better performances.

As our future work, we plan to improve our work as follows:

Real-time monitoring between GPU and CPU. Currently, in our proof-of-concept system, we relied on ConvNetJS, a Javascript-based library for deep learning. However, other more scalable libraries run most of the intensive computations in GPU, which has its own memory space separate from that of CPU. However, the front-end monitoring system usually works on the CPU side, so in order to truly achieve the real-time monitoring of the training process, memory copy operations from GPU to CPU should be frequently performed, which can degrade the computational efficiency of the training process. To handle this issue, some partial information could be selectively transferred based on a particular criterion, e.g., only when the nontrivial amount of changes of a parameter occur. Otherwise, multi-threaded syncing between the memory spaces of GPU and CPU, which performs memory copy operations only when the computing resource of GPU is available, could also be another option.

Advanced dynamic steering capabilities. So far, we provided the capabilities of dynamic node/layer addition/removal in our system. However, many other advanced dynamic steering capabilities could be developed. For instance, skipping some nodes/layers that are already trained sufficiently can accelerate the subsequent optimization steps. When nodes/layers are added/removed, their initialization could be carefully performed so that the newly added nodes/layers can capture complementary information of data to the existing nodes/layers. When removing nodes/layers, we could recommend those that have minimal impact to the overall performance, e.g., a redundant node from clustered nodes. We may be able to define the criteria to determine such minimal effects in various ways, e.g., the variable importance score of each node/layer.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. R2215-15-1021, Visual Analytics for Real-time User-driven Deep Learning).

References

[1] D. Bruckner, J. Rosen, and E. R. Sparks. deepviz: Visualizing convolutional neural networks for image classification. 2014.

[2] Adam W Harley. An interactive node-link visualization of convolutional neural networks. In *Advances in Visual Computing*, pages 867–877. Springer, 2015.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[4] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[8] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way

to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[10] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[11] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hods Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[12] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer vision–ECCV 2014*, pages 818–833. Springer, 2014.

[13] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

Clustrophile: A Tool for Visual Clustering Analysis

Çağatay Demiralp
IBM Research
cagatay.demiralp@us.ibm.com

ABSTRACT

While clustering is one of the most popular methods for data mining, analysts lack adequate tools for quick, iterative clustering analysis, which is essential for hypothesis generation and data reasoning. We introduce Clustrophile, an interactive tool for iteratively computing discrete and continuous data clusters, rapidly exploring different choices of clustering parameters, and reasoning about clustering instances in relation to data dimensions. Clustrophile combines three basic visualizations – a table of raw datasets, a scatter plot of planar projections, and a matrix diagram (heatmap) of discrete clusterings – through interaction and intermediate visual encoding. Clustrophile also contributes two spatial interaction techniques, *forward projection* and *backward projection*, and a visualization method, *prolines*, for reasoning about two-dimensional projections obtained through dimensionality reductions.

Keywords

Clustering, projection, dimensionality reduction, visual analysis, experiment, Tukey, out-of-sample extension, forward projection, backward projection, prolines, sampling, scalable visualization, interactive analytics.

1. INTRODUCTION

Clustering is a basic method in data mining. By automatically dividing data into subsets based on similarity, clustering algorithms provide a simple yet powerful means to explore structures and variations in data. What makes clustering attractive is its unsupervised (automated) nature, which reduces the analysis time. Nonetheless, analysts need to make several decisions on a clustering analysis that determine what constitutes a cluster, including which clustering algorithm and similarity measure to use, which samples and features (dimensions) to include, and what granularity (e.g., number of clusters) to seek. Therefore, quickly exploring the effects of alternative decisions is important in both reasoning about the data and making these choices.

Although standard tools such as R or Matlab are extensive and computationally powerful, they are not designed to support such interactive iterative analysis. It is often cumbersome, if not impossible, to run what-if scenarios with these tools. In response, we introduce Clustrophile, an interactive visual analysis tool, to help analysts to perform iterative clustering analysis. Clustrophile couples three basic visualizations, a dynamic table listing of raw datasets, a scatter plot of planar projections, and a matrix diagram (heatmap) of discrete clusterings, using interaction and intermediate visual encoding. We consider dimensionality reduction as a form of continuous clustering that complements the discrete nature of standard clustering techniques. We also contribute two spatial interaction techniques, *forward projection* and *backward projection*, and a visualization method, *prolines*, for reasoning about two-dimensional projections obtained through dimensionality reduction.

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16) August 14th, 2016, San Francisco, CA, USA.

DOI: 10.1145/1235

2. RELATED WORK

Clustrophile builds on earlier work on interactive systems supporting visual clustering analysis. The projection interaction and visualization techniques in Clustrophile are related to prior efforts in user experience with scatter-plot visualizations of dimensionality reductions.

2.1 Visualizing Clusterings

Prior research applies visualization for improving user understanding of clustering results across domains. Using coordinated visualizations with drill-down/up capabilities is a typical approach in earlier interactive tools. The Hierarchical Clustering Explorer [37] is an early and comprehensive example of interactive visualization tools for exploring clusterings. It supports the exploration of hierarchical clusterings of gene expression datasets through dendrograms (hierarchical clustering trees) stacked up with heatmap visualizations of data.

Earlier work also proposes tools that make it possible to incorporate user feedback into clustering formation. Matchmaker [28] builds on techniques from [37] with the ability to modify clusterings by grouping data dimensions. ClusterSculptor [30] and Cluster Sculptor [8], two different tools, enable users to supervise clustering processes in various clustering methods. Schreck *et al.* [35] propose using user feedback to bootstrap the similarity evaluation in data space (trajectories, in this case) and then apply the clustering algorithm.

Prior work has also introduced various techniques for comparing clustering results of different datasets or different algorithms [10, 29, 34, 37]. DICON [10] encodes statistical properties of clustering instances as icons and embeds them in the plane based on similarity using multidimensional scaling. Pilhofer *et al.* [34] propose a method for reordering categorical variables to align with each other and thus augment the visual comparison of clusterings. The recent tool XCluSim [29] supports comparison of several clustering re-

sults of gene expression datasets using an approach similar to that of the Hierarchical Clustering Explorer.

Clustrophile is similar to earlier work in coordinating basic and auxiliary visualizations to explore clusterings. Clustrophile focuses on supporting iterative, interactive exploration of data with the ability to explore multiple choices of algorithmic parameters along with hypothesis testing through visualizations and interactions as well as formal statistical methods. Finally, Clustrophile is domain-agnostic and is intended to be a general tool for data scientists.

2.2 Making Sense with and of Dimensionality Reductions

Dimensionality reduction is a common method for analyzing and visualizing high-dimensional datasets across domains. Researchers in statistics and psychology pioneered the use of techniques that project multivariate data onto low-dimensional manifolds for visual analysis (e.g., [2, 16, 15, 25, 38, 40]). PRIM-9 (Picturing, Rotation, Isolation, and Masking — in up to 9 dimensions) [15] is an early visualization system supporting exploratory data analyis through projections. PRIM-9 enables the user to interactively rotate the multivariate data while continuously viewing a two dimensional projection of the data. Motivated by the user behavior in the PRIM-9 system, Friedman and Tukey [16] first propose a measure, the projection index, for quantifying the "usefulness" of a given projection plane (or line) and, then, an optimization method, the projection pursuit, to find the most useful projection direction (i.e., one that has the highest projection index value). The proposed index considers the projections that result in large spread with high local density to be useful. In an axiomatic approach that complements the projection pursuit, Asimov introduces the grand tour, a method for viewing multidimensional data via orthogonal projections onto a sequence of two-dimensional planes [2]. Asimov considers a set of criteria such as density, continuity, and uniformity to select a sequence of projection planes from all possible projection planes and provides specific methods to devise such sequences. Note that the space of all possible two-dimensional planes through the origin is a Grassmannian manifold. Asimov's grand tours can be seen as geodesic curves with desired properties in this manifold.

Despite their wide use (and overuse), interpreting and reasoning about dimensionality reductions can often be difficult. Earlier work focuses on better conveying projection (reduction) errors, integrating user feedback into the projection process and evaluating the effectiveness of various dimensionality reductions. Low-dimensional projections are generally lossy representations of the data relations. Therefore, it is useful to convey both overall and per-point dimensionality reduction errors to users when desired. Earlier research proposes techniques for visualizing projection errors using Voronoi diagrams [3, 26] and "correcting" them within a neighborhood of the probed point [11, 39]. Stahnke *et al.* [39] suggest a set of interactive methods for interpreting the meaning and quality of projections visualized as scatter plots. The methods make it possible to see approximation errors, reason about positioning of elements, compare them to each other, and visualize the extrapolated density of individual dimensions in the projection space.

In certain cases, expert users have prior knowledge of how the projections should look. To enable user input to guide projections, earlier research has proposed various projection and interaction techniques [9, 13, 17, 20, 21, 44]. Enabling users to adjust the projection positions or the weights of data dimensions and distances is a common approach in earlier research for incorporating user feedback to projection computations. For example, iPCA [20] enables users to interactively modify the weights of data dimensions in computing projections. Similarly, X/GGvis [9] allows users to change the weights of dissimilarities input to the MDS stress function along the with the coordinates (configuration) of the embedded points to guide the projection process. Endert *et al.* [14] apply similar ideas to an additional set of dimensionality-reduction methods while incorporating user feedback through spatial interactions. The spatial interactions, *Forward projection* and *backward projection*, that we introduce here are developed for dynamically reasoning about dimensionality-reduction methods and the underlying data, not for incorporating user feedback.

Prior research also evaluates dimensionality-reduction techniques [7, 27] as well as visualization methods for representing dimensionally-reduced data [36]. Sedlmair *et al.* find that two-dimensional scatter plots outperform scatter-plot matrices and three-dimensional scatter plots in the task of separating clusters [36]. Lewis *et al.* [27] report that experts are consistent in evaluating the quality of dimensionality reductions obtained by different methods, but novices are highly inconsistent in such evaluations. A later study finds, however, that experts with limited experience in dimensionality reduction also lack clear understanding of dimensionality-reduction results [7].

Forward projection, *backward projection* and *prolines* are new techniques and complement earlier work in improving interactive reasoning with dimensionality reductions, particularly in order to facilitate dynamically asking and answering hypothetical questions about both the underlying data and the dimensionality reduction.

3. THE DESIGN OF CLUSTROPHILE

We developed Clustrophile for data scientists, using their regular feedback at each stage of the development process. We discuss below the design of Clustrophile, stressing the rationale behind our choices, basic visualizations and interactions.

3.1 Design Criteria

In our collaboration with data scientists, we identified four high-level criteria to consider in designing Clustrophile.

Show Variation Within Clusters Clustering is useful for grouping data points based on similarity, enabling users to discover salient structures in data while reducing the cognitive load. However, differences among data points within clusters are lost. Clustrophile has coordinated views—Table, Projection, and Clustering—that facilitate exploration of differences among data points at different levels of granularity. The projection view holds a scatter-plot visualization of the data reduced to two dimensions through dimensional reduction, thus providing a continuous spatial view of similarities among high-dimensional data points.

Allow Quick Iteration over Parameters In clustering analysis, users typically need to make several decisions, including which clustering method and distance (dissimilarity) measure to use, how many clusters to create, which features and data subsets to consider, and the like. After an initial clustering, users would like to be able to iterate on and refine these decisions. Clustrophile enables users to interactively update and apply clustering and projection algorithms and parameters at any point in their analysis.

Facilitate Reasoning about Clustering Instances Users often would like to know what features (dimensions) of the data points are important in determining a given clustering instance or how different choices of features or distance measures might affect the clustering. Clustrophile allows users to add/remove features interactively and to change distance measures used in clustering and projections.

Promote Multiscale Exploration The ability to interactively drill down into data is crucial for exploration and effective use of visual encoding variables, particularly in two-dimensional space.

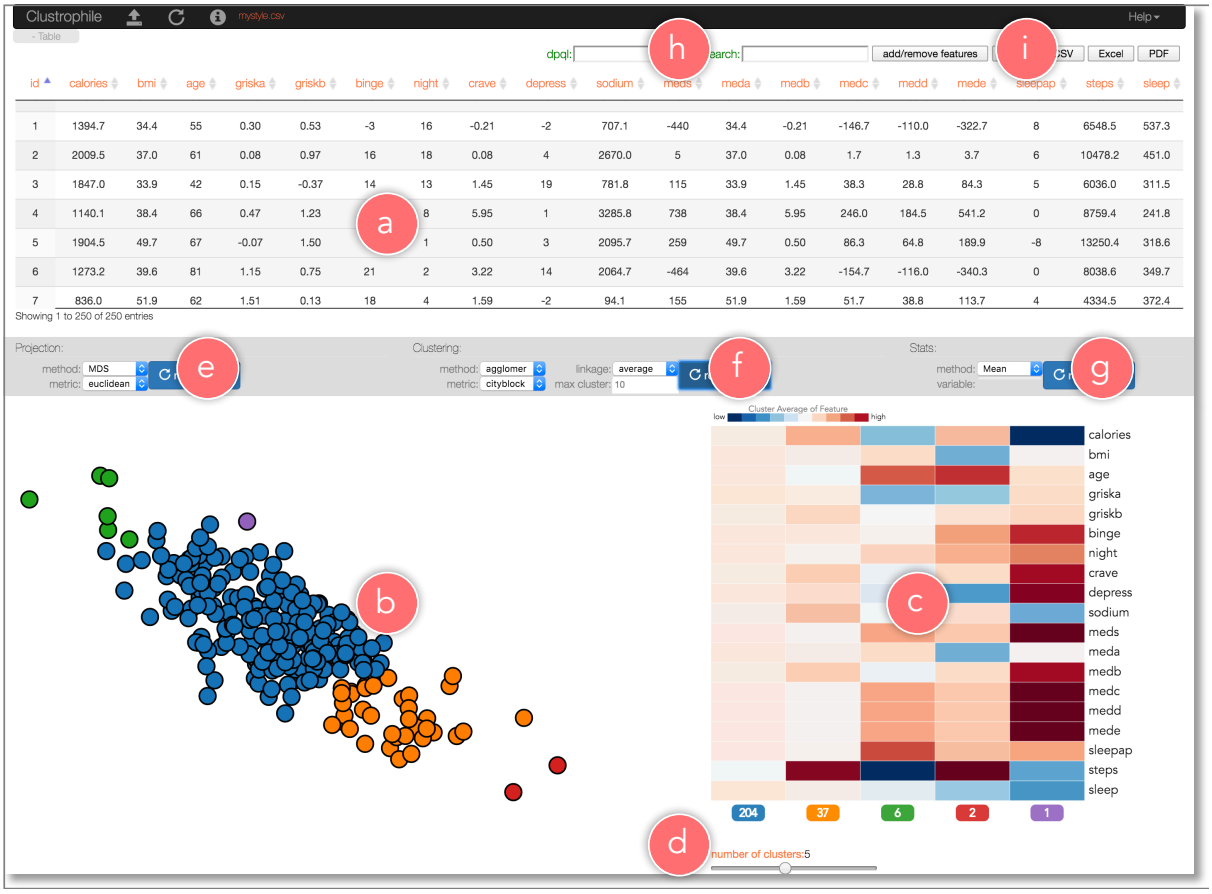


Figure 1: Clustrophile is an interactive visual analysis tool for computing data clusters and iteratively exploring and reasoning about clustering instances in relation to data subsets and dimensions through what-if scenarios. To this end, Clustrophile combines three basic visualizations, a) a table of raw datasets, b) a scatter plot of planar projections, and c) a matrix diagram (heatmap) of discrete clusterings, using interaction and intermediate visual encoding. Clustrophile enables users to interactively d) change the number of clusters, quickly explore several e) projection and f) clustering algorithms and parameters, run g) statistical analysis, including hypothesis testing, and dynamically filter h) the observations and i) features to which visual analysis is applied.

Clustrophile supports dynamic filtering of data across the views. In addition, Clustrophile makes possible the application of clustering and projection methods to filtered subsets of data, providing a semantic zoom-in and zoom-out capability.

3.2 Views

Clustrophile has five coordinated views: Table, Projection, Clustering, Statistics and Playground.

Table The Table view (Figure 1a) contains a dynamic table visualization of data. Tables in Clustrophile can be searched, filtered, sorted, and exported as needed (Figure 1h,i). Upon loading, data first appears as a table listing in this view, giving users a direct and familiar way to access the records. Clustrophile supports input files in the Comma Separated Values (CSV) format. Clustrophile also enables exporting the current table in CSV, Portable Document Format (PDF), or Excel file formats. Alternatively, users can simply the current table to the clipboard to paste in other applications.

Clustering The Clustering view (Figure 1c) contains a heatmap (matrix diagram) visualization of the current clustering. The columns of the heatmap corresponds to the number of clusters and are ordered from left to right based on size (i.e., the first column represents the largest cluster in the current clustering). The rows of the heatmap represent the features, and the color of each cell encodes the normalized average feature value for clusters. Clustrophile supports dynamic computation of clusterings using the kmeans and agglomerative clustering algorithms with several choices of similarity measures and, in the case of agglomerative clustering, linkage options (Figure 1f). The choices can be changed easily and clustering can be recomputed using the model panel above the clustering view. Similarly, users can dynamically change the number of clusters by using a sliding bar (Figure 1d).

Projection Clustering algorithms divide data into discrete groups based on similarity, but different degrees of variation within and between groups are suppressed. Clustrophile provides two-dimensional projections obtained using dimensionality reduction that complement the discrete clusterings. The Projection view (Figure 1b) contains a scatter-plot visualization of the current data reduced to two dimensions by using one of six dimensionality-reduction methods: Principal Component Analysis (PCA), Classical Multidimensional Scaling (CMDS), non-metric Multidimensional Scaling (MDS), Isomap, Locally Linear Embedding (LLE), and t-distributed Stochastic Neighbor Embedding (t-SNE) [42]. As with clustering, users can select among several similarity measures with which to run the projection algorithms (Figure 1e). Each cir-

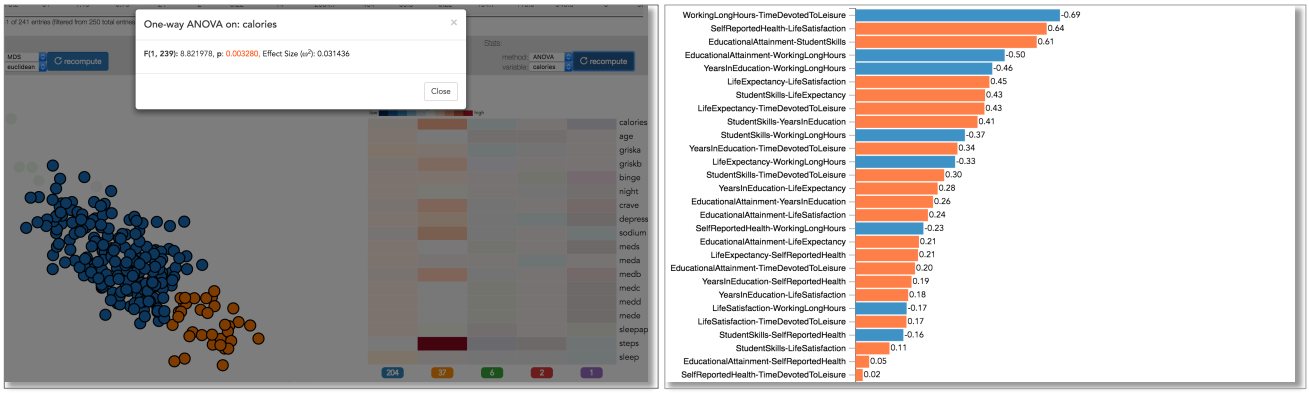


Figure 2: (Left) ANOVA test on calories between two selected clusters in a life style dataset. (Right) Correlation coefficients for all pairs of features (development indices) in a development indicators dataset [31] for OECD member states. Correlation values are sorted based on their absolute value. The sign of correlation is encoded by color.

cular node in the scatter plot represents a data point and their color encodes their cluster membership in the currently active clustering method.

Statistics This view displays the results of the most recent statistical computation. Currently, Clustrophile provides standard point statistics along with a hypothesis-testing functionality using ANOVA and pairwise correlation computations between features (Figure 2).

Playground Clustrophile enables the exploration of two- dimensional projections of the data through forward and backward projections. In the Playground view, users can create a copy of an existing data point and interactively modify its feature values to see how its projected position changes. Conversely, users can change the projected position and see what feature values satisfy this change.

3.3 Interactions

Brushing and Linking. We use brushing & linking to select data across and coordinate the views of Clustrophile. This is the main mechanism that lets users observe the effects of one operation across the views.

Dynamic filtering In addition to brushing, Clustrophile provides two basic mechanisms for dynamically filtering data (Figure 1h). First, its search functionality lets users filter the data using arbitrary keyword search on feature names and values. Second, users can also filter the table using expressions in a mini-language. For example, typing age $>$ 40 & weight$<$180 dynamical selects data points across views where the fields age and weight satisfy the entered constraint.

Adding and Removing Features Understanding the relevance of data dimensions or features to the analysis is an important yet challenging goal in data analysis. Clustrophile enables users to add and remove features (dimensions) and explore the resulting changes in clustering and projection results (Figure 1i).

3.4 Interacting with Dimensionality Reductions

Dimensionality reduction is the process of reducing the number of dimensions in a high-dimensional dataset in a way that maximally preserves inter-datapoint relations of some form as measured in the original high-dimensional space. As with clustering, most dimensionality-reduction techniques are unsupervised and learn salient structures explaining the data. Unlike clustering, however,

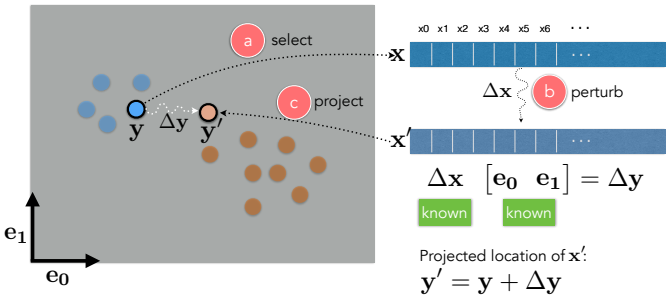


Figure 3: *Forward projection* enables users to a) select any data point **x** in the projection, b) interactively change the feature or dimension values of the point and c) observe how that changes the current projected location, **y**, of the point. For PCA, the positional change vector $\Delta\mathbf{y}$ can be derived directly by projecting the data change vector $\Delta\mathbf{x}$ onto the first two principal components, $\mathbf{e_0}$ and $\mathbf{e_1}$.

dimensionality-reduction methods discover continuous representations of these structures.

Despite its ubiquitous use, dimensionality reduction can be difficult to interpret, particularly in relation to original data dimensions. *What do the axes mean?* is probably users' most frequent question when looking at scatter plots in which points (nodes) correspond to dimensionally-reduced data. Clustrophile integrates *forward projection*, *backward projection*, and *prolines* to facilitate direct, dynamic examination of dimensionality reductions represented as scatter plots.

There are many dimensionality-reduction methods [42] and developing effective and scalable dimensionality-reduction algorithms is an active research area. Here we focus on principal component analysis (PCA), one of the most frequently used dimensionality-reduction techniques; note that the discussion here applies as well to other linear dimensionality-reduction methods. PCA computes (learns) a linear orthogonal transformation (high-dimensional rotation) of the empirically centered data into a new coordinate frame in which the axes represent maximal variability. The orthogonal axes of the new coordinate frame are called principal components. To reduce the number of dimensions to two, for example, we project the centered data matrix, rows of which correspond to data samples and columns to features (dimensions),

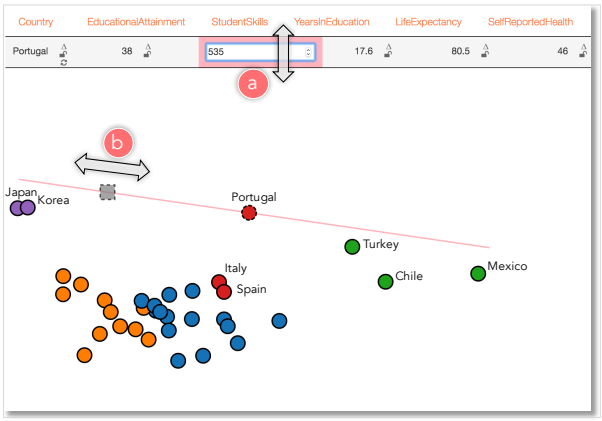


Figure 4: *Forward projection* in action. *Forward projection* enables user to explore if and how much StudentSkills explains the difference between Portugal and Korea in a projection of OECD member countries based on a set of development indices. The user a) dynamically changes the value of the StudenSkills dimension for Portugal and b) observes the dynamically updated projection. In this case, user discovers that StudentSkills is the most important feature explaining the difference between Portugal and Korea.

onto the first two principal components, $\mathbf{e_0}$ and $\mathbf{e_1}$. Details of PCA along with its many formulations and interpretations can be found in standard textbooks on machine learning or data mining (e.g., [5, 18]).

3.5 Forward Projection

Forward projection enables users to interactively change the feature or dimension values of a data point, $\mathbf{x}$, and observe how these hypothesized changes in data modify the current projected location, $\mathbf{y}$ (Figures 3,4). This is useful because understanding the importance and sensitivity of features (dimensions) is a key goal in exploratory data analysis.

We compute forward projections using out-of-sample extension (or extrapolation) [42]. Out-of-sample extension is the process of projecting a new data point into an existing projection (e.g., learned manifold model) using only the properties of the projection. It is conceptually equivalent to testing a trained machine learning model with data that was not part of training.

In the case of PCA, we obtain the two-dimensional position change vector $\Delta\mathbf{y}$ by projecting the data change vector $\mathbf{x}'$ onto the principal components: $\Delta\mathbf{y} = \Delta\mathbf{x}\,\mathbf{E}$, where $\mathbf{E} = \begin{bmatrix}\mathbf{e_0} & \mathbf{e_1}\end{bmatrix}$.

3.6 Visualizing Forward Projections: Prolines

It is desirable to see in advance what forward projection paths look like for each feature. Users can then start inspecting the dimensions that look interesting or important.

Prolines visualize forward projection paths based on a range of possible values for each feature and data point (Figures 6, 7). Let $\mathbf{x}_i$ be the value of the ith feature for the data point $\mathbf{x}$. We first compute the standard deviation σ_i for the feature in the dataset and devise a range $I = \begin{bmatrix}\mathbf{x}_i - k\sigma_i, & \mathbf{x}_i + k\sigma_i\end{bmatrix}$. We then iterate over the range with a step size of $c\sigma_i$, compute the forward projections as discussed above, and then connect them as a path. The constants k, c control respectively the extent of the range and the step size with which we iterate over the range.

Prolines will be straight lines for linear dimensionality-reduction

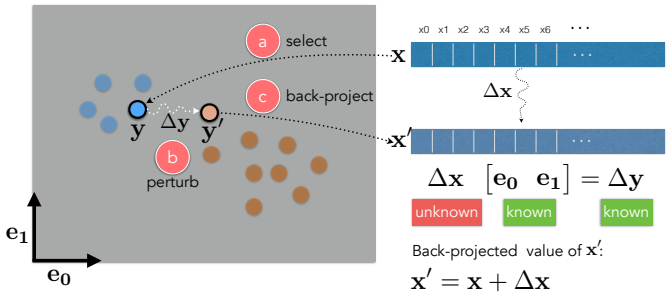


Figure 5: Through *backward projection*, users can a) select a node in the projection that corresponds to a data point $\mathbf{x}$, b) directly move the node in any direction and c) dynamically observe what data changes $\Delta\mathbf{x}$ would satisfy the hypothesized change $\Delta\mathbf{y}$ in the projected position. In PCA projections, $\Delta\mathbf{x}$ can be obtained by solving for it in the linear equation $\Delta\mathbf{x}\begin{bmatrix}\mathbf{e_0} & \mathbf{e_1}\end{bmatrix} = \Delta\mathbf{y}$.

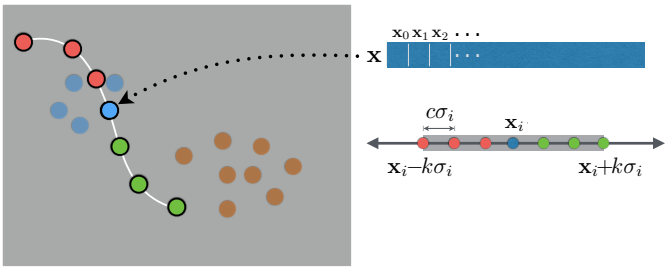


Figure 6: *Prolines* visualize paths of forward projections. For a given feature $\mathbf{x}_i$ of a data point $\mathbf{x}$, we construct a *proline* by connecting the forward projections of the points regularly sampled from a range of $\mathbf{x}$ values, where all features are fixed but $\mathbf{x}_i$ changes from $\mathbf{x}_i - k\sigma_i$ to $\mathbf{x}_i + k\sigma_i$. σ_i is the standard deviation of the ith feature in the dataset and k, c are constants controlling respectively the extent of the range and the step size with which we iterate over the range.

methods (Figure 7), and therefore computing forward projections only for the extremum values of the range I is sufficient. Also note that in the case of PCA projections *prolines* reduces to plotting the contributions of the feature to the principal components (loadings) as a line vector.

3.7 Backward Projection

Backward projection as an interaction technique is a natural complement of *forward projection*. Consider the following scenario: a user looks at a projection and, seeing a cluster of points and a single point projected far from this group, asks what changes in the feature values of the outlier point would bring the apparent outlier near the cluster. Now, the user can play with different dimensions using *forward projection* to move the current projection of the outlier point near the cluster. It would be more natural, however, to move the point directly and observe the change (Figures 5, 8, 9).

The formulation of *backward projection* is the same as that of *forward projection*: $\Delta\mathbf{y} = \Delta\mathbf{x}\,\mathbf{E}$. In this case, however, $\Delta\mathbf{x}$ is unknown and we need to solve the equation.

As formulated, the problem is underdetermined and, in general, there can be infinitely many data points (feature values) that project to the same planar position. Therefore, our implementation in Clustrophile supports both unconstrained and constrained backward projections. Users can introduce equality as well as inequality constraints (Figure 10).

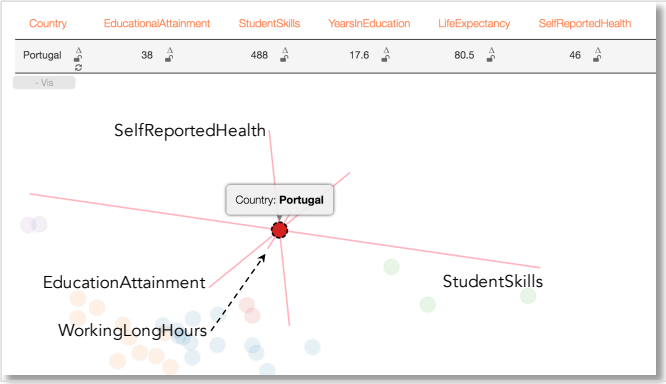


Figure 7: *Prolines* for Portugal in a PCA projection of OECD member countries based on their values for a set of development indices. *Prolines* will be straight lines for linear dimensionality-reduction methods. In addition, the length of each path corresponds to the speed (sensitivity, variability) along the corresponding dimension. For example, StudentSkills is the most sensitive feature determining the projection in this case. Note that *forward projection* animates the speed of the change along *prolines*, giving the user an additional cue about the importance of the dimension in the projection.

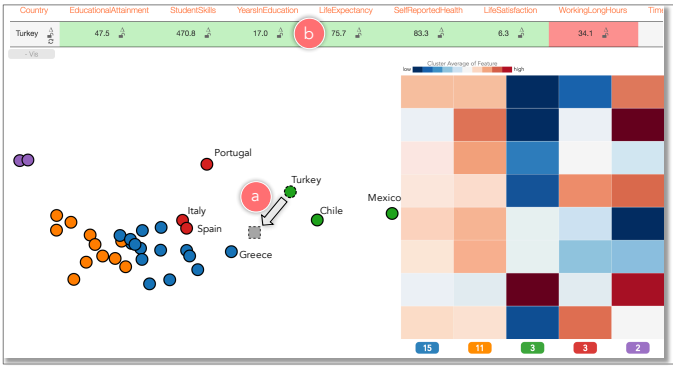


Figure 8: Unconstrained *backward projection*. A user, curious about the projection difference between Turkey and Greece, a) moves the proxy node for Turkey (gray square with dashed border) towards Greece. The feature values for Turkey are automatically updated to satisfy the new projected position as the node is moved. The user b) observes that, as Turkey gets closer to Greece, WorkingLongHours decreases (encoded with red) while EducationAttainment, StudentSkills, YearsInEducation, LifeExpectancy, SelfReportedHealth, and LifeSatisfaction increase (green). TimeDevotedToLeisure (not seen) stays constant (gray).

In the case of unconstrained backward projection, we find $\Delta\mathbf{x}$ by solving a regularized least-squares optimization problem.

$$\begin{aligned} \underset{\Delta\mathbf{x}}{\text{minimize}} \quad & \|\Delta\mathbf{x}\|^2 \\ \text{subject to} \quad & \Delta\mathbf{x}\,\mathbf{E} = \Delta\mathbf{y} \end{aligned}$$

Note that this is equivalent to setting $\Delta\mathbf{x} = \Delta\mathbf{y}\,\mathbf{E}^T$. In general, for linear projections we have the unconstrained back projection directly.

As for constrained backward projection, we find $\Delta\mathbf{x}$ by solving the following quadratic optimization problem:

$$\begin{aligned} \underset{\Delta\mathbf{x}}{\text{minimize}} \quad & \|\Delta\mathbf{x}\,\mathbf{E} - \Delta\mathbf{y}\|^2 \\ \text{subject to} \quad & \mathbf{C}\Delta\mathbf{x} = \mathbf{d} \\ & \mathbf{lb} \leq \Delta\mathbf{x} \leq \mathbf{ub} \end{aligned}$$

$\mathbf{C}$ is the design matrix of equality constraints, $\mathbf{d}$ is the constant vector of equalities, and $\mathbf{lb}$ and $\mathbf{ub}$ are the vectors of lower and upper boundary constraints.

3.8 System Details

Clustrophile is a web application based on a client-server model (Figure 11). We implemented Clustrophile's web interface in Javascript with help of D3 [6] and AngularJS [1] libraries. We generated the parser for the mini-language used to filter data with PEG.js [33]. Most of the analytical computations are performed on Clustrophile's Python-based analytics server, which has four modules: clustering, projection, statistics, and solver. These modules are mainly wrappers, making heavy use of SciPy [22], NumPy [43], and scikit-learn [32] Python libraries. The solver module uses CVXOPT [12] for quadratic programming.

4. USER FEEDBACK

Clustrophile is a research prototype under development and has been used over several months by data scientists and researchers in the healthcare domain. While we have not conducted a formal study, we briefly discuss the informal feedback we have gathered.

Users cared most about the time-saving aspects of Clustrophile. They were pleased with the ability to explore different clustering and projection algorithms and parameters without going back to their scripts. Similarly, among Clustrophile's favorite functionalities were the ability to add and remove features and iteratively recompute clusterings and projections on filtered data while staying in the context of data analysis session. We found that our users were more familiar with clustering than projection; indeed, for some the relation between clustering and projection view was not always clear.

The most important request from our users was scalability. As soon as they started using Clustrophile in commercial projects, they realized that they wanted to be able to analyze large datasets without losing Clustrophile's current interactive and iterative user experience (more on this in the following section).

5. SAMPLING FOR SCALE

The power of visual analysis tools such as Clustrophile comes from both facilitating iterative, interactive analysis and leveraging visual perception. Exploring large datasets at interactive rates, which typically involves coordination of multiple visualizations through brushing and linking and dynamic filtering, is, however, a challenging problem. One source of the challenge is the cost of interactive computation and rendering. Another is the perceptual and cognitive cost (e.g., clutter) users incur when dealing with large numbers of visual elements.

There are two basic approaches to this problem: precomputation and sampling [19]. Precomputation involves processing data into a form (typically tiles or cubes) to interactively answer queries (e.g., zooming, panning, brushing, etc.) that are known in advance. This approach has been the prevalent method both in the visualization community and the database community, from which most of the current techniques originate from. However, precomputation is not always feasible or, indeed, desirable. Scalable visualization tools based on precomputation are typically applied to the visualization

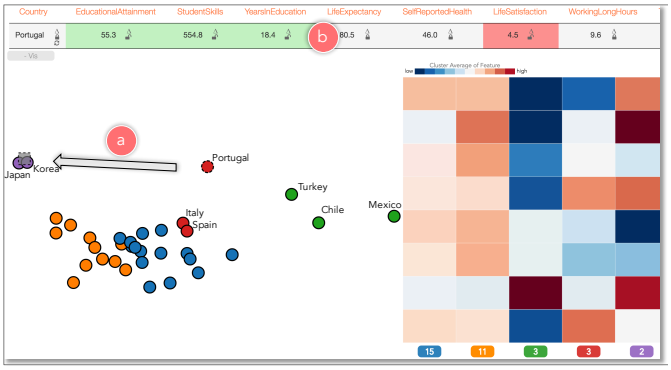


Figure 9: Constrained *backward projection*. A user explores the projection difference between Portugal and Korea, first fixing (i.e., setting equality constraints) all dimensions but EducationAttainment, StudentSkills, YearsInEducation, LifeSatisfaction and then a) moving the proxy node for Portugal nearer to Korea. The user b) observes that LifeSatisfaction decreases while EducationAttainment, StudentSkills, and YearsInEducation increase.

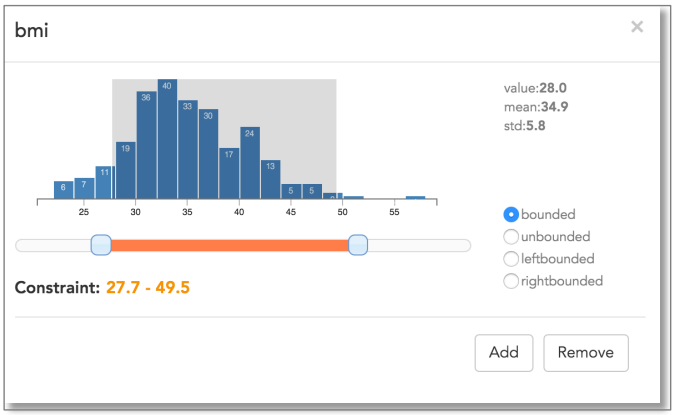


Figure 10: Clustrophile interface for entering inequality constraints for *backward projection*. Users can enter bounded, left and right bounded interval constraints. The histogram shows the distribution of the future (bmi, body mass index, in this case) for which the constraints are entered. The user can adjust the constraints interactively using the histogram brush or the slider.

of low-dimensional, spatial (e.g., map) datasets as precomputation is infeasible when the data is high dimensional, quickly expanding the combinatorial space of possible cubes or tiles. And, in general, precomputation is inflexible as it restricts the ability to run arbitrary queries.

Sampling, considering only a selected subset of the data at a time for analysis, is an attractive alternative to precomputation for scaling interactive visual analytics tools. Sampling has generality and the advantage of easing computational and perceptual/cognitive problems at once. In principle, there is no reason that sampling-based visual analysis should not be a viable and practical option. In the end, the field of statistics builds on the premise that one can infer properties of a population (read complete data) from its samples. There are, however, two major challenges that, we believe, also limit wider adoption of sampling in general [19].

First is a concern about potential biases introduced by sampling. This concern seems, however, to be at least partly unfounded, since neither aggregation bias of precomputation nor sampling bias of

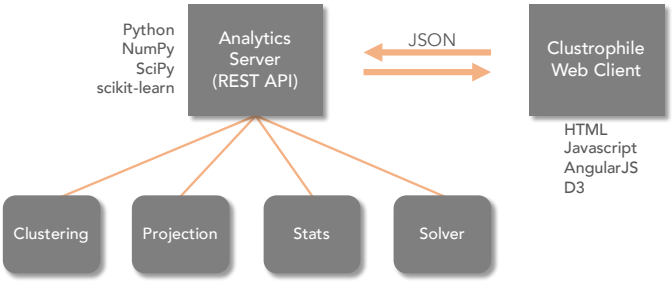


Figure 11: Clustrophile architecture.

complete data appear to cause as much concern. In recent work, Kim *et al.* improve the effectiveness (and trustworthiness) of sampling-based visualizations by guaranteeing the preservation of relations (e.g., ranking) within the complete data [24]. The second challenge is the lack of understanding how users interact with sampling in visual analytics tools or how sampling affects the user experience and comprehension. Can we develop models of user behavior regarding sampling? How can we improve the user experience with sampling through visualization and interaction? How can users control the sampling process without being experts in statistics?

Addressing these challenges would accelerate the adoption of sampling and improve the utilization of the unique opportunity that sampling provides in enabling visual analysis on large datasets without losing the power of iterative, interactive visual-analysis workflow that tools like Clustrophile facilitate.

6. REFLECTIONS ON PROJECTIONS

Using out-of-sample extrapolation, *forward projection* avoids re-running dimensionality-reduction algorithms. From the visualization point of view, this is not just a computational convenience but also has perceptual and cognitive advantages such as preserving the constancy of scatter-plot representations. For example, re-running (training) a dimensionality-reduction algorithm with addition of a new sample can significantly alter a two-dimensional scatter plot of the dimensionally-reduced data, despite all the original inter-datapoint similarities stay unchanged. Many of the dimensionality-reduction algorithms are based on eigenvector computations (if $\mathbf{v}$ is an eigenvector of a matrix so is $-\mathbf{v}$). Even different runs on the same dataset can result in different—typically, flipped—planar coordinates.

What about interacting with nonlinear dimensionality reductions? There are out-of-sample extrapolation methods for many nonlinear dimensionality-reduction methods that make the extension of *forward projection* with *prolines* possible [4]. As for *backward projection*, its computation will be direct in certain cases (e.g., when an autoencoder is used). In general, however, some form of constrained optimization specific to the dimensionality-reduction algorithm will be needed. Nonetheless, it is highly desirable to develop general methods that apply across dimensionality-reduction methods.

7. VISUAL ANALYSIS IS LIKE DOING EXPERIMENTS

Data analysis is an iterative process in which analysts essentially run mental experiments on data, asking questions and (re)forming and testing hypotheses. Tukey and Wilk [41] were among the first to observe the similarities between data analysis and doing exper-

iments. They list eleven similarities, for example, "Interaction, feedback, trial and error are all essential; convenience dramatically helpful." Albeit often implicitly, the visualization literature makes a strong case for designing visual analysis tools to support quick, iterative analysis flow that is conducive to hypothesis generation and testing (e.g., [23]).

We integrate our spatial interaction techniques for exploring and reasoning with dimensionality reductions into Clustrophile, which uses familiar data-mining and visualization methods to facilitate iterative, interactive clustering analysis. Injecting new techniques into familiar workflows is an effective way for assessing their usefulness and adoption. Tukey and Wilk make an important observation on the adoption of new techniques as part of their analogy: "There can be great gains from adding sophistication and ingenuity ... to our kit of tools, just as long as simpler and more obvious approaches are not neglected."

It is a standard practice to design visualization tools by considering criteria determined to support user tasks. While this approach is necessary for creating useful tools, our experience in developing Clustrophile suggests that the design process can benefit from the regulating clarity of general, higher-level conceptual models. To explore and reason about data, analyts generally have the basic data-mining and visualization techniques. They often, however, lack interactive tools integrating these techniques to facilitate quick, iterative what-if analysis, which is essential for hypothesis generation and data reasoning. Extending Tukey and Wilk's analogy between data analysis and running experiments to visual analysis, *visual analysis like doing experiments*, provides a useful conceptual model for a large segment of visual analysis applications. Clustrophile, along with *forward projection*, *backward projection*, and *prolines*, contributes to the kit of tools needed to facilitate performing visual analysis in a similar way to running experiments.

8. REFERENCES

[1] AngularJS. http://angularjs.org/.

[2] D. Asimov. The grand tour: A tool for viewing multidimensional data. *SIAM J. Sci. Stat. Comput.*, 6(1):128–143, Jan. 1985.

[3] M. Aupetit. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 70(7-9):1304–1330, mar 2007.

[4] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in neural information processing systems*, 16:177–184, 2004.

[5] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[6] M. Bostock, V. Ogievetsky, and J. Heer. D^3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics*, 17(12):2301–2309, 2011.

[7] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner. Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *BELIV'14*. Association for Computing Machinery (ACM), 2014.

[8] P. Bruneau, P. Pinheiro, B. Broeksema, and B. Otjacques. Cluster sculptor, an interactive visual clustering system. *Neurocomputing*, 150:627–644, 2015.

[9] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen. Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, 17(2):444–472, 2008.

[10] N. Cao, D. Gotz, J. Sun, and H. Qu. Dicon: Interactive visual analysis of multidimensional clusters. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2581–2590, Dec 2011.

[11] J. Chuang, D. Ramage, C. Manning, and J. Heer. Interpretation and trust. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. Association for Computing Machinery (ACM), 2012.

[12] CVXOPT. http://cvxopt.org/.

[13] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. Association for Computing Machinery (ACM), 2012.

[14] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. Observation-level interaction with statistical models for visual analytics. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 121–130, Oct 2011.

[15] M. A. Fisherkeller, J. H. Friedman, and J. W. Tukey. Prim-9: An interactive multidimensional data display and analysis system. In *Proc. Fourth International Congress for Stereology*, 1974.

[16] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23(9):881–890, Sept 1974.

[17] M. Gleicher. Explainers: Expert explorations with crafted projections. *IEEE Trans. Visual. Comput. Graphics*, 19(12):2042–2051, dec 2013.

[18] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[19] J. M. Hellerstein. Interactive Analytics. In M. S. J. M. H. Peter Bailis, Joseph M. Hellerstein and M. Stonebraker, editors, *Readings in Database Systems*. MIT Press, 5th edition, 2015.

[20] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. ipca: An interactive system for pca-based visual analytics. *Computer Graphics Forum*, 28(3):767–774, 2009.

[21] S. Johansson and J. Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE Trans. Visual. Comput. Graphics*, 15(6):993–1000, nov 2009.

[22] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001. http://www.scipy.org/.

[23] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Enterprise data analysis and visualization: An interview study. In *IEEE Visual Analytics Science & Technology (VAST)*, 2012.

[24] A. Kim, E. Blais, A. Parameswaran, P. Indyk, S. Madden, and R. Rubinfeld. Rapid sampling for visualizations with ordering guarantees. *Proc. VLDB Endow.*, 8(5):521–532, jan 2015.

[25] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[26] S. Lespinats and M. Aupetit. CheckViz: Sanity check and topological clues for linear and non-linear mappings. *Computer Graphics Forum*, 30(1):113–125, dec 2010.

[27] J. M. Lewis, L. Van Der Maaten, and V. de Sa. A behavioral investigation of dimensionality reduction. In *Proc. 34th Conf. of the Cognitive Science Society (CogSci)*, pages 671–676, 2012.

[28] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg. Comparative analysis of multidimensional, quantitative data. *IEEE Trans. Visual. Comput. Graphics*, 16(6):1027–1035, nov 2010.

[29] S. L'Yi, B. Ko, D. Shin, Y.-J. Cho, J. Lee, B. Kim, and J. Seo. XCluSim: a visual analytics tool for interactively comparing multiple clustering results of bioinformatics data. *BMC Bioinformatics*, 16(11):1–15, 2015.

[30] E. J. Nam, Y. Han, K. Mueller, A. Zelenyuk, and D. Imre. Cluster Sculptor: A visual analytics tool for high-dimensional data. In *2007 IEEE Symposium on Visual Analytics Science and Technology*. Institute of Electrical & Electronics Engineers (IEEE), oct 2007.

[31] OECD Better Life Index. http://www.oecdbetterlifeindex.org/. Accessed: May 27, 2016.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[33] PEG.js. http://pegjs.org/.

[34] A. Pilhofer, A. Gribov, and A. Unwin. Comparing clusterings using bertin's idea. *IEEE Trans. Visual. Comput. Graphics*, 18(12):2506–2515, dec 2012.

[35] T. Schreck, J. Bernard, T. von Landesberger, and J. Kohlhammer. Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization*, 8(1):14–29, 2009.

[36] M. Sedlmair, T. Munzner, and M. Tory. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Trans. Visual. Comput. Graphics*, 19(12):2634–2643, dec 2013.

[37] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results [gene identification]. *Computer*, 35(7):80–86, jul 2002.

[38] R. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. I. *Psychometrika*, 27(2):125–140, 1962.

[39] J. Stahnke, M. Dörk, B. Müller, and A. Thom. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Trans. Vis. Comput. Graph.*, 22(1):629–638, 2016.

[40] W. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[41] J. W. Tukey and M. Wilk. Data analysis and statistics: an expository overview. In *Proceedings of the November 7-10, 1966, fall joint computer conference*, pages 695–709. ACM, 1966.

[42] L. Van der Maaten, E. Postma, and H. Van den Herik. Dimensionality reduction: A comparative review. *Technical Report TiCC TR 2009-005*, 2009.

[43] S. v. d. Walt, S. C. Colbert, and G. Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[44] M. Williams and T. Munzner. Steerable, progressive multidimensional scaling. In *IEEE Symposium on Information Visualization*. Institute of Electrical & Electronics Engineers (IEEE), 2004.

A Visual Approach for Interactive Co-Training

Qi Han
Institute for Visualization and Interactive Systems
University of Stuttgart
Germany
Qi.Han@vis.uni-stuttgart.de

Weimeng Zhu
Institute for Natural Language Processing
University of Stuttgart
Germany
st113027@stud.uni-stuttgart.de

Florian Heimerl
Institute for Visualization and Interactive Systems
University of Stuttgart
Germany
Florian.Heimerl@vis.uni-stuttgart.de

Steffen Koch
Institute for Visualization and Interactive Systems
University of Stuttgart
Germany
Steffen.Koch@vis.uni-stuttgart.de

Thomas Ertl
Institute for Visualization and Interactive Systems
University of Stuttgart
Germany
Thomas.Ertl@vis.uni-stuttgart.de

ABSTRACT

Co-training is a popular semi-supervised method to build classifiers by combining labeled and unlabeled data. It trains two classifiers with a small amount of initially labeled data and iteratively retrains them after exchanging their high confidence instances. As the initial amount of labels is very small, however, the performance can suffer from the label pollution problem. We therefore propose an interactive visual approach that improves the stability of co-training through user inspection of transferred instances. It includes a visualization of classifier uncertainties and disagreement. It further helps users to quickly identify possible mistakes of the automatic approach by guiding user's attention to the instances which are labeled differently than the majority of their nearest neighbors and instances which are labeled differently by the two base classifiers. To help users examine such instances, we also include a visual explanation which shows important features of an instance along with its raw data. We show the effectiveness of our approach with a usage scenario and by comparing it with the classical co-training approach through experiments. Finally, we discuss limitations and propose several possibilities for future improvement.

Keywords

interactive machine learning, visualization, machine learning, semi-supervised learning, co-training, multi-view learning, bootstrapping

1. INTRODUCTION

Nowadays, a huge amount of text data are produced every day. The number of emails received by individuals is growing every minute. Documents produced by online communities or other organizations are also increasing quickly. This poses great challenges to human individuals as well as organizations to manage and analyze this data to obtain insights. To classify data into different categories is one of the common methods to organize it. Automatic classification can free humans from repeatedly doing the same classification tasks, as they can learn classification rules from them. However, to obtain a classifier with good performance, people need to label many data points to feed into the classification algorithm, which is very time consuming. Semi-supervised learning methods [7, 22] can reduce human effort in labeling a lot of data. They combine information from a small amount of labeled and a large amount of unlabeled data to learn classification criteria. Clustering [12] can also automatically assign documents into categories. However, the natural clustering of the data does not necessarily inconsistent with the intention of users. Recently, researchers also suggest approaches to actively integrate human intention into the clustering results [4] . In this work, we focus on improving semi-supervised learning methods through human interaction.

Co-training [5] is one of the most popular semi-supervised learning methods. It starts by training two classifiers on two different feature sets with an initial set of labeled data. It proceeds by iteratively growing the set of labeled data and retrain the classifiers on this new dataset. In each iteration, it allows each of the two classifiers to label a few unlabeled data instances, which they can classify with a high confidence. These instances are added to the set of training data for subsequent iterations. The two classifiers are retrained on this new set and co-training can start a new iteration. Zho and Li [21] use a flow-graph to explain the co-training method. We add a visual element representing users into the graph to clarify the role of users as can be seen in Figure 1.

However, as the initial amount of labeled instances is very

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16) August 14th, 2016, San Francisco, CA, USA.

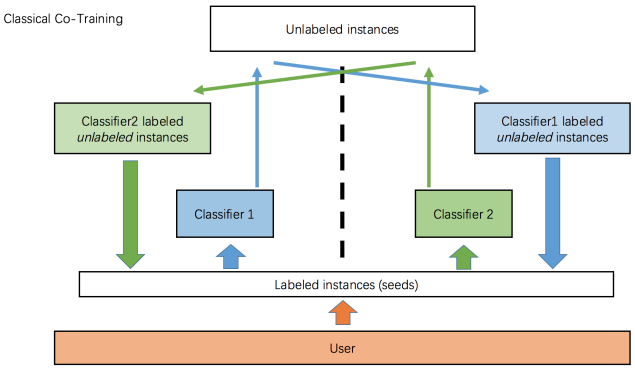


Figure 1: A graphical summarization of the classical co-training method.

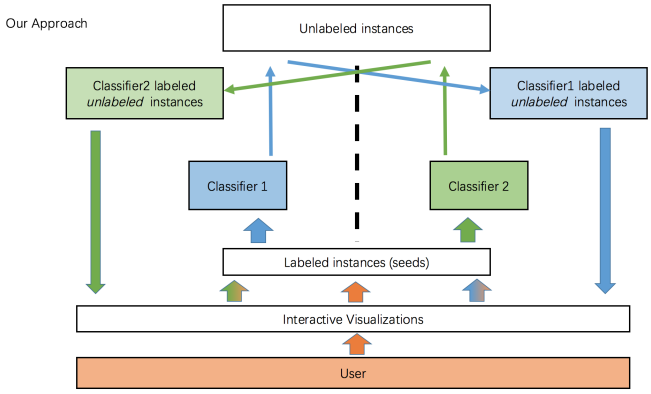


Figure 2: A graphical summarization of the interactive co-training approach proposed in this work.

small, the two base classifiers do typically not have a high classification performance at the beginning. They can thus potentially introduce labeling mistakes into the training set used to train new classifiers in the following iterations. This makes the performance of the co-training method unstable, especially when used on noisy data [9, 18]. In this paper, we propose an interactive visual approach for co-training. Our approach improves the stability of co-training through user inspection of the transferred instances. It includes a visualization based on Parallel Coordinates [11] to depict the uncertainties of the two base classifiers and the disagreement between them. Furthermore, it encodes the label distribution of the nearest neighbors of the instances. The visualization guides users' attention to the instances which are labeled differently than the majority of their nearest neighbors or the instances which are labeled differently by the two base classifiers. In doing so, it helps users to quickly identify possible mistakes of the automatic approach. Correcting those mistakes will likely boost the performance of the co-training algorithm. To help users examine these mistakes more closely, we also include a visual explanation which shows important features of an instance along with its raw data. Figure 2 depicts the main components of our approach.

2. RELATED WORK

In this section we discuss approaches that are related to ours. They can be broadly divided into two groups. The first one address machine learning in semi-supervised settings. The second group of approaches focuses on integrating interactive visualizations and machine learning to improve performance of automatic algorithms, or provide explanation for their decisions.

Blum et al. [5] suggest co-training to combine information form labeled and unlabeled data to train classifiers. They also show the effectiveness of co-training under the assumption that the two views on the data are conditionally independent. Since then, co-training has been applied in many domains, for example, to classify emails [14], or to label roles of named entities [9]. Additional research provides more insight about why and in which settings the co-training method works well [3]. In addition, limitations of the classical co-training method have been identified, such as its difficulties with noisy or unbalanced data [16, 18]. Muslea et al. [17] suggest to combine active learning and co-training to obtain a more stable and effective semi-supervised method. Our approach also tries to improve on the classical co-training method. However, we achieve this by letting users actively inspect or correct the automatic method through interactive visualizations.

Recently, there has also been many research efforts that aim to bring visual interaction and machine learning together to allow users to guide and steer machine learning methods [1, 13]. ModelTracker [2] is a visual approach for analyzing performance of machine learning models. FeatureInsight [6] and FeatureForge [10] propose visual approaches for feature engineering. Ribeiro et al. [15] propose a method for explaining the reasons behind predictions made by machine learning methods. They suggest a method to derive important features by each prediction. We propose a visual approach for classifier building in a semi-supervised way to reduce human effort and increase the trust of users in the resulting classification model.

3. APPROACH

In this section, we first describe the data processing work flow of our approach. Along with that, we also describe the reasons why we have chosen some algorithms over the others. We then highlight the tasks that we intend to support and introduce visualizations to address these challenges.

3.1 Data Processing

Our approach is based on the idea that we first build two classifiers based on two views of the data with just a few labels. These two classifiers are then utilized to label additional instances so that a high performance classifier can be obtained.

The first step in our workflow is to construct two different feature sets or so-called views from the datasets. In many cases, the dataset has a natural split of views. For example, for images with additional text descriptions we can construct one view from the image data and the other one from the textual data. For email data, we can construct one view from email meta data, like the header or sender of the emails and obtain the other view from the textual data. For datasets without a natural split, a simple procedure to acquire two views is to randomly assign features to one of the two views. Several other methods have been suggested to split a single set of features into two views, which are more suitable for

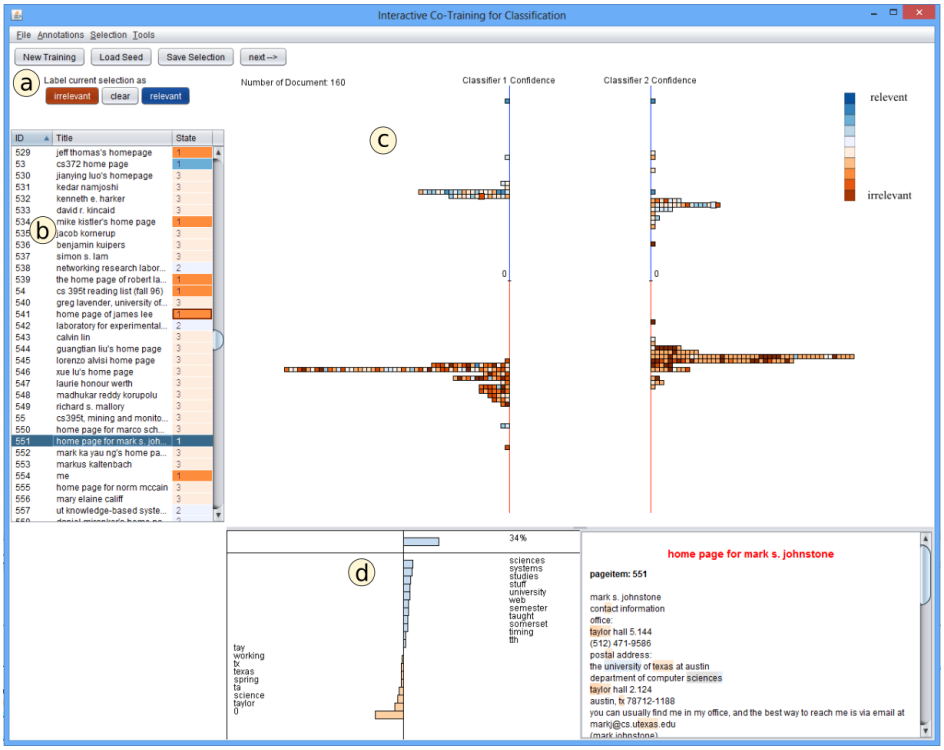


Figure 3: A screenshot of our system. It includes: a) the Control Panel comprising several control buttons b) the Instance Table view showing the instances of the dataset and classification states; c) the Overview view depicting the classification confidence of the two classifiers about the instances and d) the Explainer view showing the evidence of the classification decisions made by a classifier.

co-training algorithms than random split. Our approach can work with both types of datasets.

In the second step, we choose a classification algorithm for the base classifiers. As opposed to the original co-training paper, that proposes Naive Bayes [20], we have decided to use linear SVMs from the Liblinear package [8] in this work. SVM is a fast and robust classification algorithm that has been shown [14] to be more suitable than Naive Bayes for text classification tasks in co-training settings. In addition, we use Platt-scaling [19] to obtain a calibrated confident score of the classifications made by the SVM classifiers. This way, we can compare classification confidences of different SVM classifiers.

As mentioned in the introduction, our approach offers a view to visually explain the classification decisions. For this purpose, we need to identify the relevant features and their importance for the instances under inspection. Depending on the type of classifier, different measures of feature importance can be used for this. In this work, we use the feature weights of the linear SVM classifier.

3.2 Visual Approach

In this section we describe different views of our approach and introduce the interactions supported by them. We build interactive visualizations to help users quickly identify possibly mislabeled instances, obtain insights about the classification uncertainty of the classifiers, understand reasons behind decisions made by the classifiers, and maintain a stable mental map of all the instances.

3.2.1 *Overview visualization*

The overview visualization is updated after each round of co-training. All instances labeled in the last round of co-training are visualized in this view. Small squared glyphs are used to represent the instances. As each instance is given two different confidence scores by the two base classifiers, we depict each one of the instances as two glyphs. They are placed along two axes according to their confidence scores by the two classifiers. As the amount of instances labeled in one round of co-training can be large, we divide the whole range of the axes into bins. We then assign the instances to these bins. When more than one instance is placed into one bin, we stack them on top of each other. This way, users can identify individual instances easily and gain an impression of the distribution of classifier confidence scores (see Fig. 4). ModelTracker [2] uses a similar design to depict the test instances of one classifier. We aim to display disagreement between two classifiers and the mislabel possibility of the instances.

We assign colors to the instances to encode the label distributions of their neighboring instances. The similarity is measured according to the Euclidean distance in the feature space of the classifiers. For each instance in the visualization, we count how many of its neighbors are labeled as positive and how many of them are labeled as negative. The ratio of these two counts is mapped to color between positive color and negative color with linear interpolation. Thus, the more neighbors of an instance are labeled as positive, the more similar its color is to the positive color. As

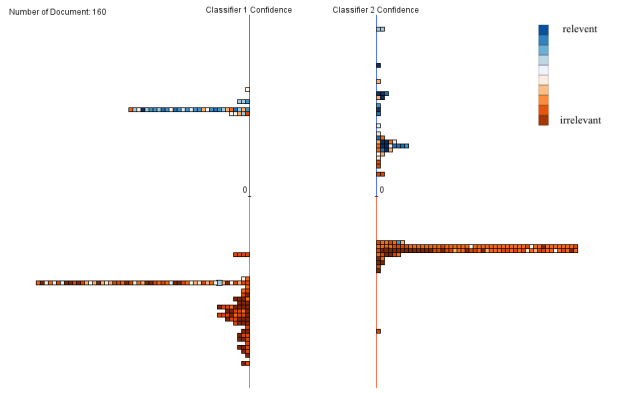


Figure 4: In the overview visualization, the instances are placed according to the confidence score of the classifiers. The colors of the instances encode the label distributions of their neighboring instances in the feature space of the classifiers.

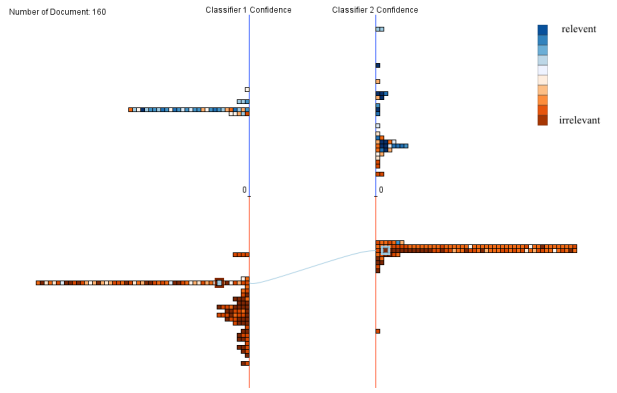


Figure 5: This figure shows the highlighting effect. The instance was positioned at the lower part of the overview view which means the instance was labeled as irrelevant. However, on the left side, the color of this instance is blue. It shows that the neighbors of the instance were labeled mostly as relevant. This inconsistence suggests that the instance might be labeled incorrectly.

distinct colors can be perceived in a pre-attentive way, users can quickly see instances that are classified as positive but have more negative neighbors or vice versa.

When users hover over one of the squares, it is highlighted by a halo around it (see Fig. 5). The color of the halo is identical to that of the square representing the same instances in the second classifier. We also show a spline connecting those two squares representing the same instance. By comparing the information from two views of the same instance, users can more easily decide whether it is worth to inspect the instance more closely.

3.2.2 *Interactive explainer view*

The explainer view is located directly below the overview visualization. It consists of one view depicting the important features along with their importance rating, and another view showing the raw data of an instance (see Fig. 6). When users hover over an instance, the explainer shows more detailed information about the instance and the classification.

On top of the explainer view, we use a bar chart starting at the center line of the view to depict classification confidence. When an instance is classified as positive, the bar grows to the right, when it is classified as negative, the bar grows to the left. Colors of the bars are consistent with the overview visualization. Features are sorted according to their importance and the bar charts are placed accordingly from top to bottom. This visualization allows users to see the most important features for the classification of an instance. This gives users cues about how the classification decision about that instance is made. To get additional information about the features influencing the classification decision, users can turn to the raw data view. The raw data view highlights highly weighted features by coloring the background of the features in the raw data. The highlights guide the attention of users during inspection of the raw data.

Additional interactions are implemented in the view to further improve the usefulness of the explainer view. Users can click on the bars depicting the features, which causes the raw data view to scroll to the position where the feature appears for the first time. If they click the bar once more, the raw data view scrolls to the next appearance of that feature. This interaction further helps users to efficiently pick important information from the raw data. If they press the shift-key while clicking on the feature bar, all the instances containing this feature will be highlighted in the overview visualization and the instance table view. When users find a feature which is responsible for a erroneous label, they can use this interaction to find out how the feature is distributed over the labels. This gives users hints on explanations of feature importance.

3.2.3 *Control panel and instances table*

The instance table (Fig. 7) works as a notebook for the users. It maintains a table of all the instances in the dataset and uses the third column of the table to depict if that instance is labeled as positive, negative, or not labeled at all. Users can sort the instances according to their ids or the status of their labels. The control panel consists a group of controls for users to interact with the co-training process. For example, they can change the labels of the instances manually using the controls in this panel. We highlight the instances which have been manually labeled by the users during the current round in the instance table.

4. EVALUATION

In this section, we first demonstrate how users can interact with the system through a detailed description of a usage scenario. We then go on to show the effectiveness of our system by comparing our approach with the classical co-training algorithms.

4.1 Usage Scenario

At the beginning, all instances are displayed within the instance list view 7. Users can label a few instances by clicking on them and reading through the raw data within the explanation panel. Once they have decided on the label of the instance, they can use the buttons in the control panel to label the instance as positive or negative.

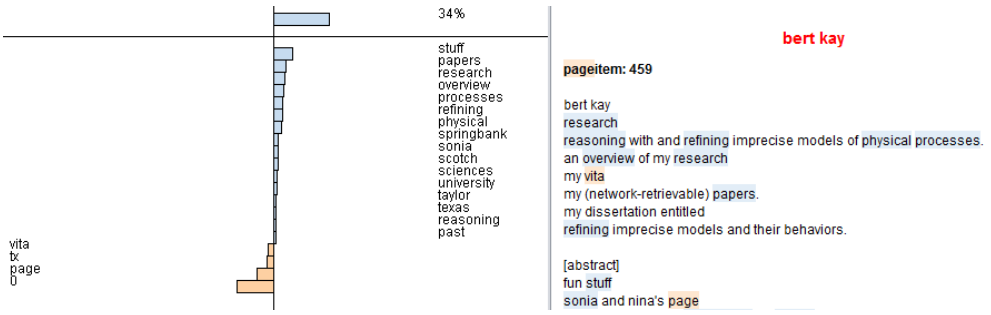


Figure 6: The explainer view shows most important features for the classification of an instance and the raw data of it.

ID	Title	State
529	jeff thomas's homepage	1
53	cs372 home page	1
530	jianying luo's homepage	3
531	kedar namjoshi	3
532	kenneth e. harker	3
533	david r. kincaid	3
534	mike kistler's home page	1
535	jacob kornerup	3
536	benjamin kuipers	3
537	simon s. lam	3
538	networking research labor...	2
539	the home page of robert la...	1
54	cs 395t reading list (fall 96)	1
540	greg lavender, university of...	3
541	home page of james lee	1
542	laboratory for experimental...	2

Figure 7: The list view shows the instances of the dataset and their current labeling status. Dark red indicates that the instance is labeled as irrelevant. Light red means that the instance has not been labeled yet but it is predicted by the classifiers as irrelevant. Similarly, dark blue and light blue means the instance is labeled or classified as relevant. User labeled instances are highlighted.

Then they can set the number of instances they wish the co-training algorithm to label in the next round and press the start button. The system iteratively labels more instances by choosing those instances rated most confidently by the classifiers.

Those newly labeled instances are depicted in the overview visualization. Users can identify the label given by the classifiers at the transfer time by observing the position of the instance squares. If the instance is positioned in the top half of the visualization, it is labeled as positive. The other way around, it is labeled as negative. They can also get an impression of how the labels of the nearest neighbors are distributed for each instance, by observing the color of squares representing the instances. Instances with more positive nearest neighbors are colored blue. Instances with more negative nearest neighbors are colored red. Seeing this can help them to quickly pick those instances, which have been labeled as positive but have a neighborhood which is labeled mostly as negative, or vice versa. Because disagreement to the majority of the labels of the nearest neighbors indicates a possible labeling error. If at least one of the classifiers has given high confidence to this label, users should be more motivated to check it, because it could have been mislabeled with a high probability. Correcting this will likely give a boost to the performance of the learned classifier.

Once they have identified one possible mistake of the classifier, they probably want to check if it is really mislabeled or not. By clicking on the instance, the explainer view shows the decisions of the classifiers and the important features on which the decision is based. Clicking on one of the shown features, the text panel will scroll to the position where that feature shows up in the text for the first time. This way users can quickly skim through long documents and concentrate only on information which is important for the classification. By clicking on one of the features, the instances which contain this feature will be also highlighted in the overview view.

Once users find a mislabeled instance, they can relabel it, and continue to examine other instances until they are satisfied with these set of newly labeled instances. They can then continue the co-training by clicking on the next button in the control panel. The system will use all the instances that were newly labeled in the subsequent co-training step, by letting the two classifiers retrain themselves based on the updated labeled set and further iteratively label more instances.

4.2 Comparison to Classical Co-Training

In this subsection, we describe the experiments to compare our approach and the classical co-training approach.

4.2.1 Experimental setup

The WebKB-Course dataset is composed by Blum et al. [5] to demonstrate the effectiveness of the co-training approach and has been subsequently used in several other works. We conduct our experiments on this dataset to compare our approach with the classical co-training algorithm.

This dataset contains data of 1051 web pages. For each web page we can construct two views/feature sets. One of them is based on the text on the web page. The other one is based on the anchor text of the links pointing to the page. The whole dataset can be divided into two parts: web pages of courses from a university or web pages of researchers in the university.

In the first step, we do experiments to obtain a estimation of error rate that we can obtain from the specific combination of the two views and the base classifier we use, which is a Support Vector Machine (SVM). For each experiment, we randomly choose 263 (25%) pages as test documents. From the rest of the 788 pages, we further randomly choose

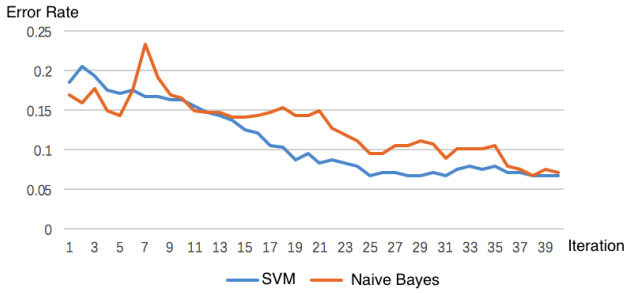


Figure 8: This figure shows the error curve of SVM compared to Naive Bayes.

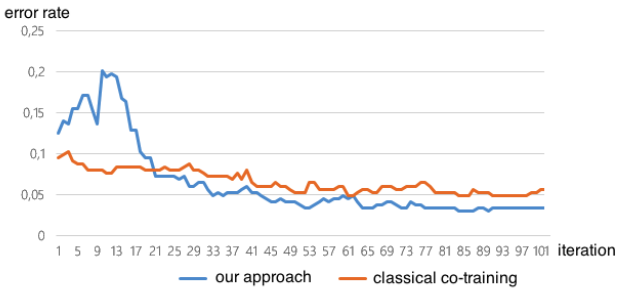


Figure 9: This figure shows the error curve of our approach compared with classical co-training approach with additional initial labels.

3 positive and 9 negative documents as seeds, as suggested by Blum et.al in their work. The rest of the 776 pages are treated as unlabeled data. We then train the SVM classifier with default parameter setting with the classical co-training algorithm and calculate the error rate of the obtained model on test data. We repeat the experiments 10 times and report the average of the results.

Fig. 8 shows that SVM achieves lower or comparable error rates as Naive Bayes Classifier in this setting. The results is in consistent with the suggestion of [14] and confirms that our choice of base classifier is reasonable.

In the second step, we conduct experiments to compare our approach with the classical co-training approach. In these experiments, we follow the same process to divide the dataset into test data, seeds and unlabeled data.

We repeat the experiment 10 times with two master students of computer science and set a time limitation of 10 minutes in total to finish each experiment. We set the number of newly labeled instances in each round to be 70, which means users have to manually inspect the instances labeled by the classifiers for about 6 times during each experiment. In average, users only read the raw data for examining possible errors about 20 times in one experiment session, most of which are mislabeled.

To enable a fair comparison with our approach, we randomly choose 20 more documents as extra seeds, to feed into the classical co-training algorithm, which does not receive further interaction during co-training. We also run ten times of the experiment on the classical co-training algorithm, and report the average performance.

4.2.2 *Experiment results*

The average error rate of the iterations during the experiment session is shown in Fig. 9.

It is obvious that error rate of both approaches decreases along with the iterations, showing that co-training is effective for such classification task. It is also reasonable that our approach makes more errors comparing to the classical co-training at the beginning, because our approach only uses 12 seeds, and the classical co-training uses 32 seeds. But the error rate decreases faster than the classical co-training, with the help of user's reviewing labels and is more stable in the later part of the co-training process. But in fact, our approach beats the classical co-training quite fast, far before the end, and keeps the advantage all the time.

During the experiments, we also notice that with the help of our visualization users tend to put less effort in manually reviewing the labels by reading the raw document text. Usually, users only checked suspicious instances obvious to them within the overview visualization, then they quickly view some information provided by the explainer view. If they keep the doubt regarding the label, they can go on to read the raw text and manually label them as a fall-back strategy. Even when they have to read the raw text, they say, the highlights made it easier for them to grasp important information from the text.

5. DISCUSSIONS AND FUTURE WORK

The result of our experiment indicates that users' interaction is effective in improving the co-training, as well as that we require less user effort in labeling the documents to achieve better performance. The latter may be due to the benefits of our visualization that helps users to identify mislabeled documents, with much less effort. Even through, our approach achieves a relatively lower error rate (5%) on WebKB-Course dataset, the improvement over the classical co-training algorithm is not very significant. The reason might be that the dataset with which we conduct our experiments is highly suitable for the co-training algorithm, so that both our approach and the classical co-training algorithm can obtain a low error rate with relatively few initial labeled data. More experiments on other datasets with different levels of difficulty for co-training algorithm will bring more insight about this aspect.

One limitation of our experiments is that we repeat the experiment with our approach 5 times for each of the two users. Although we did not observe significant learning effect between different runs of the experiments, due to the random assignment of the initial set of labeled data, an experiment with more users would be necessary to increase the validity of the results.

As we have mentioned in the data processing section, our approach needs to identify the relevant features and their importance for the instances under inspection. For some type of classifiers, it is straight forward to rate the importance of their features. With linear classifiers, such as linear SVMs, we can use the feature weights of the trained model as feature importance. For Naive Bayes classifier, we could use the probabilities of the features conditioned on the classes as a measure of their importance. For other types of classifiers, like kernel SVMs, there is research that proposes ways to

derive feature importance for individual test instances [15] for general type of classifiers. In this regard, our approach is quite general.

In this work we focus on using interactive visualizations to improve the performance of the co-training method. In the future we also want to investigate how to depict the changes of the classifiers during the co-training process. This could further increase users' trust in the resulting classifiers. In addition, we only handle binary classification problems, so far. In the future, we aim to extend the approach to multi-class classification problems. One way to achieve that is to divide multi-class classification problems into several binary problems using a 1 vs n strategy or n vs n strategies. Further, in this work, we only deal with one specific multi-view learning algorithm: the co-training algorithm. We could also extend our approach to work with general type of multi-view algorithms.

6. CONCLUSION

In this work we proposed a visual approach for co-training. It includes several visualizations to help users interact with the co-training process. The overview visualization depicts classification uncertainty and disagreement between two classifiers, and enables users to spot possible mislabels. The explainer view shows important features for an instance along with its raw data, and allows users to examine the classification of an instance more closely. The instance list view shows all the instances and their labeling status. The control panel lets user correct the mislabeled instance manually and start a next round of co-training iteration. Together they present an effective way for building classifiers in a human steered semi-supervised way. We showed the effectiveness of our approach through detailed description of a usage scenario and by comparing it to the classical co-training approach.

7. ACKNOWLEDGMENTS

We would like to thank the two master students for testing our approach, the reviewers for their valuable feedback, and our colleagues for the insightful discussions.

8. REFERENCES

[1] J. Allen, C. I. Guinn, and E. Horvtz. Mixed-initiative interaction. *IEEE Intelligent Systems and their Applications*, 14(5):14–23, 1999.

[2] S. Amershi, M. Chickering, S. M. Drucker, B. Lee, P. Simard, and J. Suh. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proc. ACM CHI*, CHI '15, pages 337–346. ACM, 2015.

[3] M.-F. Balcan, A. Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. In *Advances in neural information processing systems*, pages 89–96, 2004.

[4] A. Biswas and D. Jacobs. Active image clustering with pairwise constraints from humans. *Int. J. Comput. Vision*, 108(1-2):133–147, May 2014.

[5] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. ACM COLT*, COLT' 98, pages 92–100. ACM, 1998.

[6] M. Brooks, S. Amershi, B. Lee, S. M. Drucker, A. Kapoor, and P. Simard. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *Proc. IEEE VAST*, pages 105–112. IEEE, 2015.

[7] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010.

[8] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[9] S. He and D. Gildea. Self-training and co-training for semantic role labeling: Primary report. Technical report, Technical Report 891, University of Rochester, 2006.

[10] F. Heimerl, C. Jochim, S. Koch, and T. Ertl. FeatureForge: A novel tool for visually supported feature engineering and corpus revision. In *COLING (Posters)*, pages 461–470. Citeseer, 2012.

[11] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[12] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[13] A. Kapoor, B. Lee, D. Tan, and E. Horvitz. Interactive optimization for steering machine classification. In *Proc. ACM CHI*, pages 1343–1352. ACM, 2010.

[14] S. Kiritchenko and S. Matwin. Email classification with co-training. In *Proc. CASCON*, CASCON '01, pages 8–. IBM Press, 2001.

[15] C. G. Marco Tulio Ribeiro, Sameer Singh. "why should i trust you?": Explaining the predictions of any classifier. *Proc. ACM SIGKDD*, 2016.

[16] E. T. Matsubara, M. C. Monard, and R. C. Prati. On the class distribution labelling step sensitivity of co-training. In *Artificial Intelligence in Theory and Practice*, pages 199–208. Springer, 2006.

[17] I. Muslea, S. Minton, and C. Knoblock. Selective sampling with naive cotesting: preliminary results. In *The ECAI 2000 workshop on Machine Learning for information extraction*, 2000.

[18] D. Pierce and C. Cardie. Limitations of co-training for natural language learning from large datasets. In *Proc. EMNLP*, pages 1–9, 2001.

[19] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[20] I. Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York, 2001.

[21] Z.-H. Zhou and M. Li. Semi-supervised learning by disagreement. *Knowl. Inf. Syst.*, 24(3):415–439, Sept. 2010.

[22] X. Zhu. Semi-supervised learning literature survey. Technical report, Department of Computer Sciences, University of Wisconsin, Madison, 2005.

Visual Quality Assessment of Subspace Clusterings

Michael Hund[1], Ines Färber[2], Michael Behrisch[1],
Andrada Tatu[1]*, Tobias Schreck[3], Daniel A. Keim[1], Thomas Seidl[4]

[1] University of Konstanz, Germany {lastname@dbvis.inf.uni-konstanz.de}
[2] RWTH Aachen University, Germany {faerber@informatik.rwth-aachen.de}
[3] Graz University of Technology, Austria {tobias.schreck@cgv.tugraz.at}
[4] Ludwig-Maximilians-University, Munich, Germany {seidl@dbs.ifi.lmu.de}

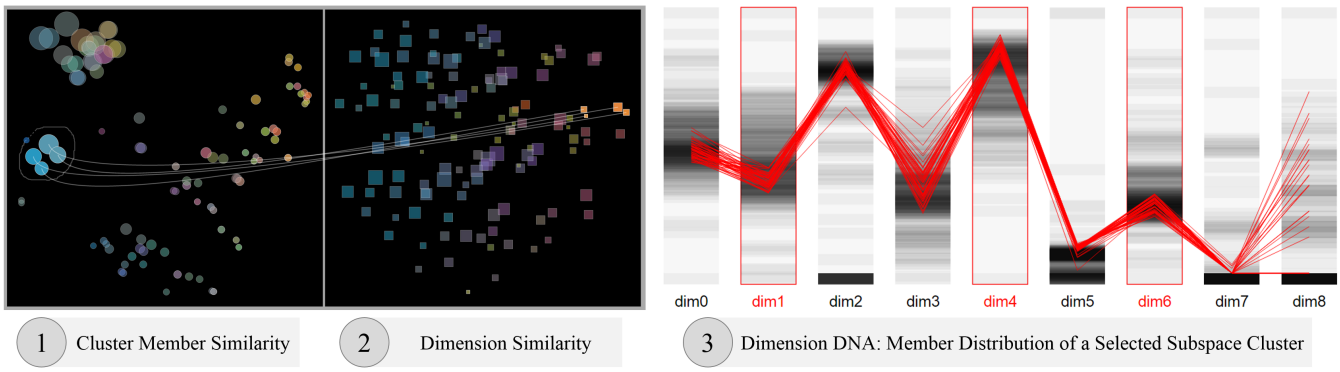


Figure 1: A comparative overview of 132 detected subspace clusters generated by the CLIQUE [2] algorithm: The two inter-linked MDS projections in the SubEval analysis framework show simultaneously the cluster member- (1) and dimension similarity (2) of subspace clusters. While the cluster member similarity view focuses on the object-wise similarity of clusters, the dimension similarity view highlights similarity aspects w.r.t. their common dimensions. The coloring encodes the similarity of clusters in the opposite projection. Both views together allow to derive insights about the redundancy of subspace clusters and the relationships between subspaces and cluster members. The DimensionDNA view (3) shows the member distribution of a selected subspace cluster in comparison to the data distribution of the whole dataset.

ABSTRACT

The quality assessment of results of clustering algorithms is challenging as different cluster methodologies lead to different cluster characteristics and topologies. A further complication is that in high-dimensional data, *subspace clustering* adds to the complexity by detecting clusters in multiple different lower-dimensional projections. The quality assessment for (subspace) clustering is especially difficult if no benchmark data is available to compare the clustering results.

In this research paper, we present SubEval, a novel subspace evaluation framework, which provides visual support for comparing quality criteria of subspace clusterings. We identify important aspects for evaluation of subspace clustering results and show how our system helps to derive quality assessments. SubEval allows assessing subspace cluster quality at three different granularity levels: (1) A global overview of similarity of clusters and estimated redundancy in cluster members and subspace dimensions. (2) A view of a selection of multiple clusters supports in-depth analysis of object distributions and potential cluster overlap. (3) The detail analysis of characteristics of individual clusters helps to understand the (non-)validity of a cluster. We demonstrate the usefulness of SubEval in two case studies focusing on the targeted algorithm- and domain scientists and show how the generated insights lead to a justified selection of an appropriate clustering algorithm and an improved parameter setting. Likewise, SubEval can be used for the understanding and improvement of newly developed subspace clustering algorithms. SubEval is part of SubVA, a novel open-source web-based framework for the visual analysis of different subspace analysis techniques.

CCS Concepts

•**Human-centered computing** → **Visualization design and evaluation methods;**

Keywords

Subspace Clustering; Evaluation; Comparative Analysis; Visualization; Information Visualization; Visual Analysis

*Former member.

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16) August 14th, 2016, San Francisco, CA, USA.

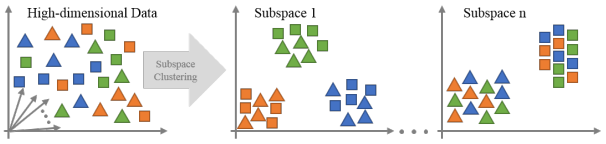


Figure 2: Subspace clustering in high-dimensional data: the same objects are grouped differently in different combinations of dimensions (=subspaces).

1. INTRODUCTION

In data analysis the selection and parametrization of clustering algorithms is usually a trial-and-error task requiring appropriate methods and analyst experience to assess the quality of the results. Furthermore, the selection of an appropriate algorithm design has a direct impact on the expected results. For example, k-Means-type clustering will likely favor voronoi-shape spaced partitions, while a density-based clustering (e.g., DBSCAN [12]) usually results in arbitrarily shaped clusters. The parameter setting, the underlying data topology and -distribution usually influence the clustering results, too. For varying applications, different cluster characteristics can be of interest for a user. Therefore, there is a need for efficient and effective evaluation methods to reliably assess the usefulness of a clustering result.

In high-dimensional data, clustering computation is influenced by the so-called *curse of dimensionality*. Noise, correlated, irrelevant, and conflicting dimension may detriment meaningful similarity computation for the input data [7]. Experiments show that the application of full-space clustering, i.e., a clustering that considers all dimensions, is often not effective in datasets with a large number of dimensions ($\geq 10-15$ dimensions) [20]. To overcome these problems, the notion of subspaces must be taken into consideration. *Subspace clustering* [21] aims to detect clusters in different, lower-dimensional projections of the original data space, as illustrated in Figure 2. The challenge is to simultaneously select meaningful subsets of objects and subsets of dimensions (=subspaces). In existing subspace cluster algorithms, the number of reported clusters is typically large and may contain substantial redundancy w.r.t. clusters and/or subspaces.

Quality assessment of subspace clustering shows to be particularly challenging as, besides the more complex result interpretation, evaluation methods for full-space clustering are not directly applicable. Generally, (subspace) clustering strives to group a given set of objects into clusters, such that objects within clusters are similar (*cluster compactness*), while objects of different clusters are dissimilar (*cluster separability*). This abstract goal leads to various *different*, yet valid and useful, cluster definitions [17]. Due to these diverging definitions it is challenging, if not impossible, to design or commonly agree on a single evaluation measure for (subspace) clustering results.

It is therefore desirable to have a unified approach for an objective quality assessment of (subspace) clustering based on different clustering methodologies, the data distribution and -topology and variety of application- and domain-dependent quality criteria. We tackle this multi-faceted analysis problem with a visual analysis process by which the computer's processing power and the human's skills in interpretation and association can be effectively combined. Numeric performance measures alone are not effective enough to give an all-embracing picture, as they are typically abstract and heuristic in nature, and defined in an application-independent way. Several application fields can benefit from such a user-supported evaluation approach: (1) selection of an appropriate clustering algorithm, (2) selection of appropriate parameter settings and (3) the design of new data mining algorithms, where algorithm scientists continuously evaluate the algorithm's results against original assumptions.

In this paper, we tackle the problem of visually evaluating the quality of one subspace clustering result. We present a novel open-source evaluation framework, called SubEval. It enhances standard evaluation approaches with effective visualizations to support the assessment of (subspace) clustering algorithms. Our contributions are as follows: (1) We present a summary of subspace cluster evaluation approaches, point to their specific foci and contrast their benefits and disadvantages. (2) We systematically structure the major evaluation criteria for subspace clustering results. (3) We discuss design- and user interaction requirements for visualizations to provide deep insights into the different quality criteria and (4) make the open-source tool SubEval available.

Compared to existing subspace visualization techniques like CoDa [15] or ClustNails [29], focusing on the knowledge extraction of subspace clusters, SubEval targets primarily the quality aspect of a clustering result. Our novel framework uses two interlinked MDS plots to simultaneously represent cluster member and subspace similarity and provides different tools for in-depth analysis of different quality criteria.

2. BACKGROUND

This section introduces definitions, terminology, concepts and related work that we rely upon to describe our approach.

2.1 Definitions and Terminology

Data record/object are used synonymously for a data instance of the dataset, i.e., $r_i \in \mathcal{R}$. A **subspace** s_l is defined as a subset of dimensions of the dataset: $s_l = \{d_i, ..., d_j\} \in \mathcal{D}$.

A **cluster** $c_j \subseteq \mathcal{R}$ contains a set of objects which are similar to each other based on a similarity function. A **clustering** result $\mathcal{C} = \{c_1, ..., c_n\}$ comprises the set of all clusters detected by an algorithm.

Crucial for the understanding of this paper is to differentiate between **full-space** and **subspace clustering**. Full-space clustering considers all dimensions ($\mathcal{D}$) for the similarity computation of its cluster members (e.g., k-Means).

A **subspace cluster** $sc_i = (s_i, c_i)$ considers only the subspace s_i for the similarity computation of the cluster members of c_i. As shown in Figure 2, a subspace clustering $\mathcal{SC} = \{sc_1, ..., sc_n\}$ consists of multiple clusters which are defined in specific subspaces. Based on the algorithm, cluster members and/or dimensions of any two clusters sc_i and sc_j may overlap, i.e., $|c_i \cap c_j| \geq 0$ and $|s_i \cap s_j| \geq 0$. The number of detected subspace clusters is typically large. For a dataset with d dimensions, there are $2^d - 1$ possible subspaces of which many may contain useful, but highly similar/redundant clusters. Same as for full-space clustering, there is a variety of different methodologies to compute useful subspace clusters [21]. However, there is no formal definition of a valid and useful (subspace) clustering result which has been accepted thoroughly by the community.

2.2 Visualization of (Subspace) Clusterings

Several techniques exist to visualize (subspace) clusters and allow users to extract semantics of the cluster structures.

The visual analysis and comparison of full-space clustering is a problem in high-dimensional data. Standard techniques like Parallel Coordinates, Dimension Stacking or Projection Techniques are applicable as a baseline [32]. Multidimensional glyphs can help to represent clusters in a 2D layout to support cluster comparison [31]. In [10], a Treemap-based glyph was designed to represent clusters and associated quality measures for visual exploration. In previous work, we considered a comparisons of hierarchical clusterings in a Dendrogram representation [9], and a comparison of Self-Organizing Map clusterings using a color-coding [8].

Visual comparison of subspace clusters is an even more difficult problem. In addition to full-space cluster visualization, also set-oriented information pertaining to subspace dimensions and possibly, multi-set membership of elements in clusters needs to be reflected. The first approaches in subspace cluster comparison is VISA [3] which visualizes subspace clusters in a MDS projection based on their cluster member similarity. Further approaches to visually extract knowledge of detected subspace clusters are ClustNails [29], SubVis [20], and an approach by Tatu et al. [28].

Visual redundancy analysis of subspace clusters is presented for example by CoDa [15] and MCExplorer [16]. Both, however, comprise only a single aspect, either dimension or cluster member redundancy. As discussed by Tatu et al. [28] clusters are only true redundant if the cluster member and the subspace topology are similar.

While the existing visualizations focus mainly on the extraction of knowledge for domain experts, SubEval changes the point of view and targets the depiction of quality criteria of subspace clusterings, such as non-redundancy, compactness, and the dimensionality of clusters.

2.3 Evaluation of Full-Space Clustering

In the following we summarize classical approaches for the evaluation of full-space clustering. We carefully investigate the advantages and drawbacks of the presented methods and highlight why visual interactive approaches are beneficial in many scenarios. As subspace clustering is a special instance of full-space clustering, the same challenges apply.

Evaluation Based on Internal Criteria.

Internal quality measures evaluate clusters or clustering results purely by their characteristics, e.g., cluster density. The literature provides a large variety of commonly used measures [22], each treating the cluster characteristics differently but usually focusing on compactness and separability. Internal measures, designed for evaluating full-space clustering, assume a single instance-to-cluster assignment and have not yet been adapted for (partially) overlapping clusters, as in subspace clustering. The criticism of this evaluation method, which does not qualify it for general performance quantification, is its subjectivity. Each measure usually favors a more particular cluster notion (e.g., RMSSTD [22] favors voronoi-shaped clusters). For each quality measure one could design an algorithm to optimize the clustering w.r.t. this particular quality measure, making comparisons to other approaches inappropriate.

External Evaluation Based on Ground Truth.

External quality measures compare the topology of a clustering result with a given ground truth clustering. Although benchmark evaluation is well accepted in the community and allows an easy comparison of different algorithms and parameter settings, the criticism to this evaluation method is manifold: The main problem of external quality measures lies in the use of a ground truth clustering itself. In most (real-world) applications and datasets with unknown data a ground truth is not available. Even if a ground truth labeling exists, it is either synthetically generated with specific clustering characteristics (c.f. criticism in Section 2.3), or it is providing a classification labeling instead of a clustering label [13]. Consequently, an algorithm, which does not correctly retrieve an already known categorization, cannot generally be regarded as bad result, as the fundamental task of clustering is to find previously unknown patterns.

Evaluation by Domain Experts.

The actual usefulness of a clustering for a certain application domain can only be assessed with a careful analysis by a domain expert. However, in many (higher-dimensional) real-world applications, the cluster result complexity is overwhelming even for domain experts. Accordingly, domain expert-based evaluation is not suited for a comparison of different clusterings, since (1) a domain expert cannot evaluate a large number of algorithms and/or parameter setting combinations, and (2) the evaluation depends on the expert and the application and does therefore not result in quantitative performance scores.

2.4 Evaluation of Subspace Clustering

In the following, we discuss current approaches for the evaluation of subspace clusterings and highlight why novel human-supported evaluation methods, such as provided by SubEval, are required for a valid quality analysis.

External Evaluation Measures.

The most commonly used method to assess the quality of a subspace clustering algorithm are external quality measures. As discussed above, the synthetically created ground-truth clusters are typically generated with particular clustering characteristics, and, for subspace clustering also with subspace characteristics. For real-world data the ground truth is not very expressive [13] and potentially varies depending on the used measure or data set [14, 24].

Internal Evaluation Measures.

The internal measures used for traditional (full-space) clustering are not applicable to subspace clustering results as (1) the existing methods do not allow for overlapping cluster members, (2) clusters need to be evaluated in their respective subspace only, i.e., it is not valid to assess the separability of two clusters which exist in different subspaces.

Domain Experts.

Often authors justify a new subspace clustering approach by exemplarily discussing the semantic interpretation of selected clusters, i.e., evaluation by domain scientists, which seems to be the only choice for some real-world data, e.g., [20]. Quite a few visualization techniques exist to support domain experts in the knowledge extraction of subspace clusters (c.f. Section 2.2). However, in subspace clustering we have to tackle three major challenges: (1) the subspace concept is complex for most domain experts, especially for non-computer-scientists, (2) the large result space and the redundancy makes it often practically unfeasible to investi-

gate all detected clusters and retrieve the most relevant ones, and (3) it is almost impossible to manually decide whether all relevant clusters have been detected or not.

Summarizing, existing quality measures for subspace clustering comprise the evaluation by external measures and/or a careful investigation by domain experts. Although both approaches have their advantages and disadvantages, they are valid and accepted in the community. Besides these techniques, we need novel methods which do not rely on ground-truth data and/or domain experts, but rather complement existing evaluation approaches. Therefore, our aim is to visualize the quality of a clustering for different clustering definitions. Furthermore, our approach supports the user in interpreting given subspace clustering result in terms of object groups and dimension sets, hence supports interactive algorithm parameter setting.

3. VISUAL QUALITY ASSESSMENT

In the following we summarize the most important quality criteria indicating a useful and appropriate *subspace* clustering result. Our quality criteria (*C1-C3*) are compiled from a literature review on objective functions for subspace clustering algorithms. Our coverage is not exhaustive, but targeted towards the major quality "understandings" in this field. For many applications, not all aspects need to be full-filled.

3.1 Quality Criteria for Subspace Clusterings

Non-Redundancy Criteria (C1).

One –if not the major– challenge in subspace clustering, is redundancy. It negatively influences a knowledge extraction due to highly similar, but not identical cluster results.

C1.1 Dimension Non-Redundancy. A useful subspace clustering algorithm emphasizes distincitive dimension/membership characteristics and avoids subspace clusters with highly similar subsets of dimensions.

C1.2 Cluster Member Non-Redundancy. A useful subspace clustering result focuses on important global groupings, avoiding clusters with many similar cluster members.

As elaborated in [28], subspace clusters are only *true redundant*, if they share most of their dimensions *and* most of their cluster members. Therefore, dimension- and cluster member redundancy have to be analyzed in conjunction.

C1.3 No Cluster-Splitup in Subspaces. Similar clusters occurring in different, non-redundant subspaces should be avoided. Generally, cluster-splitups cannot be considered redundant, as each cluster may contain new information. Yet, it provides reasons to suspect that the cluster members form a common cluster in a single, higher-dimensional subspace.

Object and Dimension Coverage Criteria (C2).

We define *object coverage* as the proportion of objects and *dimension coverage* as the proportion of dimensions of the datasets which are part of at least one subspace cluster. A high coverage of both objects and dimensions helps to understand the global patterns in the data.

C2.1 Object Coverage. To reason about all data objects, a useful subspace clustering algorithm extracts –not mandatorily a full– but obligatory high object coverage.

C2.2 Dimension Coverage. To reason about all dimension characteristics, a useful subspace clustering algorithm covers each dimensions in at least one subspace cluster.

Clustering Characteristics Criteria (C3).

Cluster characteristics are related to internal cluster evaluation measures. Although the following aspects are not summarized into a common measure for subspace clustering, most algorithms try to optimize the following properties:

C3.1 Cluster Compactness. Objects belonging to a cluster need to be similar in all dimensions of their respective subspace. Non-compact clusters represent dependencies between the cluster members which are not very strong.

C3.2 Cluster Separability. A useful algorithm assigns similar objects to the same cluster. Objects belonging to different clusters in the same subspace need to be dissimilar. A separability definition of clusters existing in different subspaces does not exist yet.

C3.3 High/Low Dimensionality. A high and a low dimensionality of a cluster can both be considered useful in different applications. While a high dimensionality is often interpreted as more descriptiveness, we argue that a low dimensional cluster can also be of interest, especially if a higher dimensional subspace contains the same cluster structures. That means, fewer dimensions correspond to lower cluster complexity. However, clusters with a very low dimensionality ($\sim$ 1-3 dimensions) are typically of no interest since no deeper knowledge can be extracted.

C3.4 High/Low Cluster Size. While most subspace clustering algorithms favor clusters with many members, we believe that in some applications clusters with a small cluster size are important, esp. when combined with *C3.1* and *C3.2*. Possible use case: a dataset contains many obvious structures, while smaller clusters may contain unexpected patterns.

3.2 Visual Design- and Interaction Requirements for Subspace Cluster Evaluation

In the following we summarize design requirements to assess the quality criteria as categorized above. In Section 4 we showcase one possible instantiation of the design requirements in our SubEval framework.

Cluster vs. Clustering. Crucial for the design of an evaluation system is to distinguish between the evaluation of a single *cluster* and the evaluation of a *clustering* result. For a single cluster, the different cluster characteristics (*C3*) are of interest, independent of a potential redundancy (*C1*) or coverage (*C2*) aspect. Likewise, for a clustering result the overall quality information, such as redundancy (*C1*) and coverage (*C2*) is important, i.e., a high-quality result can still contain a few clusters with e.g., low compactness (*C3.1*).

Reasoning for a Good/Bad Quality. Another important aspect is to distinguish between a *cluster/clustering quality* and *explanations/reasons* for a good/bad quality. The first aspect primarily states whether a clustering is useful or not, while the second one requires a more fine-grained level for an in-depth understanding.

Interactive Visualizations. For many of the presented quality criteria it is not mandatory to develop complex visualizations. Simple visual encodings and well-established visualizations, such as bar- or line charts, allow to extract quickly useful meta-information (e.g., the redundancy of dimensions in subspaces or the number of not clustered data records). We show examples in Figures 5 and 6. Even simple visualizations become powerful analysis tools if interactivity and faceted-browsing is applied, i.e., an analyst interactively selects all subspace clusters containing a frequently occurring dimension and gets details on-demand, such as data

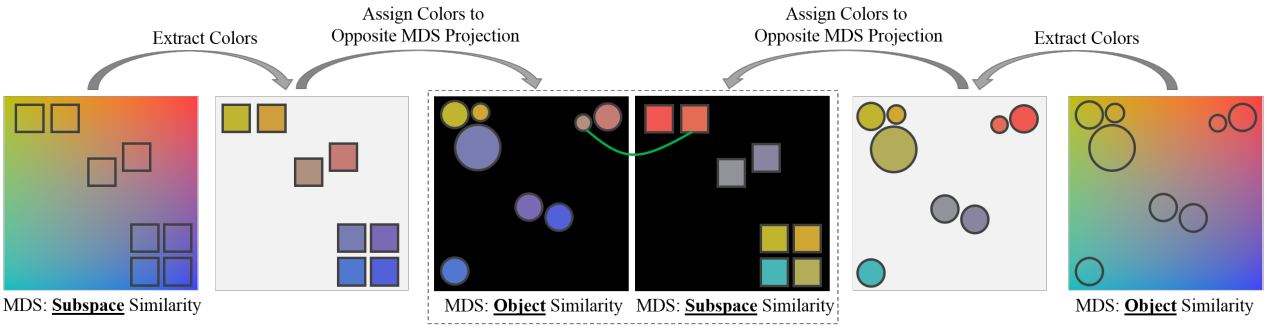


Figure 3: Schema for the two interlinked MDS projections: An MDS projection is computed for both, the subspace- and object similarity of the subspace clusters using an appropriate distance function. Afterwards, the projection is mapped on top of a perceptual linear 2D colormap where similar color correspond to a nearby location in MDS projection (similar objects). Finally, the colors of the subspace clusters of the *object similarity projection* are assigned to the clusters in the *subspace similarity projection* and vice versa. Interpretation: Nearby subspace clusters in a MDS projection with the same color are similar in both, the object and subspace space; nearby clusters with different colors in the object similarity projection are only in their cluster members, but not in the subspace.

distribution and commonalities of the selected clusters. This technique is known as *linking-and-brushing* [5].

Multi-Granularity Analysis. To get detailed information of the quality of subspace clustering result at different granularity levels, a multi-level analysis from overview to detail is required (see also the visual information seeking mantra by SHNEIDERMAN [26]: *Overview first, zoom and filter, then details-on-demand*). In the following, we describe a potential workflow with three granularity levels *(L1-L3)*:

L1 Overview. The user needs to quickly develop an overview of the redundancy aspect (*C1*) for all detected clusters to decide whether a result is generally useful or not. Quality must be assessed separately, but also related in two spaces: cluster member- and dimension space. Redundancy is highly correlated with similarity as many highly similar cluster imply a notion of redundancy. Therefore, an appropriate visualization must be able to depict (relative) similarity between data objects, as well as between dimension combinations. One possible visualization technique to fulfill these visual properties is Multi-dimensional Scaling (MDS) [11], as depicted in Figure 1. MDS approximates the high-dimensional distances in a low (2D) dimensional space, thus making it suitable for depicting redundancy aspects (*C1*). Set-oriented distance functions such as the Jaccard Index or the Overlap Coefficient are a possible mean to intuitively compute the similarity between two clusters or subspaces:

$$Jaccard\_Similarity(c_i, c_j) = 1 - \frac{|c_i \cap c_j|}{|c_i \cup c_j|}$$

A similarity value of 0 refers to two completely identical clusters. Likewise, the similarity can be computed between two subspaces. Based on the similarity notion of a specific application, a different distance function can be applied. Other subspace cluster properties, such as the cluster size or compactness, can be encoded with additional visual variables (e.g., color or size) into the points of the projection or by bar charts as presented, e.g., in Figures 5 and 6.

L2 Cluster Comparison. At the cluster comparison level, the user needs to validate a potential object- and/or dimension redundancy identified in (*L1*). The analyst will also have to examine the coverage of the cluster members and dimensions, and particularly compare the coverage of multiple clusters. As one potential solution we propose one MDS projection per subspace cluster, illustrating the object similarity by location in the MDS projection and highlight the cluster members accordingly as further described in Section 4.2. Another approach to analyze common members/dimensions in different clusters are Parallel Set visualization [6].

L3 Data Instance. At the last analysis level, the user needs to investigate the properties of a single selected cluster. Only at this fine-grained detail level the analyst will understand why specific objects are clustered within a subspace, and, more importantly, to find potential reasons why a clustering fails to identify a valid object to cluster relationship. One possible approach to analyze the data distribution of high-dimensional data are Parallel Coordinates [18], which show the distribution of multiple data objects among a large set of dimensions. It might be useful to combine the Parallel Coordinates with a box plot or another density measure in order to compare the data objects with the underlying data distribution of the dataset. An example for such an enhanced parallel coordinates plot can be found in Figure 1.

4. SUBEVAL: INTERACTIVE EVALUATION OF SUBSPACE CLUSTERINGS

In the following section, we introduce SUBEVAL which is one instantiation of the previously described multi-granularity analysis. The overview level (*L1*) uses two inter-linked MDS projections to simultaneously analyze cluster member- and dimension redundancy (Section 4.1). Section 4.2 (*L2*) introduces CLUSTDNA for detailed redundancy analysis and Section 4.3 (*L3*) describes DIMENSIONDNA to explore the distribution on a data instance level of one selected cluster.

4.1 Interlinked MDS for Cluster Member and Dimension Space Exploration

At the overview level, redundancy aspects *(C1)* are focused by visualizing the relative pair-wise similarity relationships of all clusters with the help of a MDS projection. In SUBEVAL simultaneously two interlinked MDS projections are used: the left MDS plot illustrates the similarity of subspace clusters w.r.t. the cluster members, and the right MDS

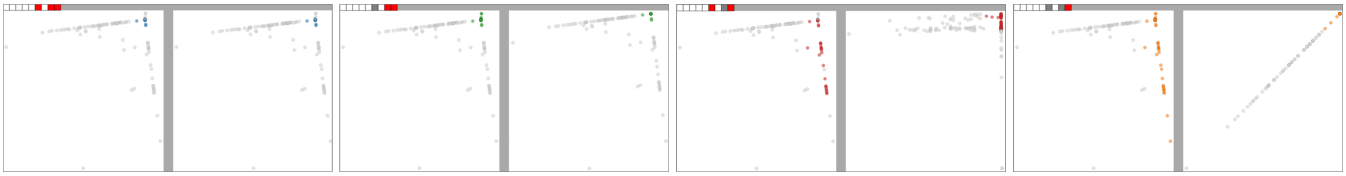


Figure 4: ClustDNA to compare the topology of 4 selected subspace clusters: each combined scatter plot represents an MDS projection of all data objects of the dataset in the subspace projection (right) and the SuperSpace (left; union of dimensions of all selected clusters). Cluster members are marked in color. The dimensions of the subspace are indicated by the top glyph (red = subspace -, grey = SuperSpace dimension).

plot depicts the similarity w.r.t. the dimension similarity. The user can change the similarity definitions in order to account for the different understanding of redundancy in the subspace analysis process. SubEval supports multiple set-oriented similarity measures (e.g., Jaccard Index). Advanced measures as proposed in [28], are planned for future work.

Visual Mapping for Redundancy Analysis.

In the MDS projection, each subspace cluster is represented by a single point/glyph. In order to compare the clusters with in the corresponding counter-MDS plot we use a 2-dimensional color schema [8, 27] that links position with color (similar position = similar color; see Figure 1 (1) and (2)). The basic intuition is that in the left MDS projection (object similarity) the cluster member similarity is encoded by the 2D coordinates (position), while the dimension similarity is encoded in color in the same projection. In other words, proximity corresponds to similar/redundant clusters w.r.t. objects and a similar color indicates similar/redundant clusters in dimension similarity. The same is true for the subspace similarity in the right projection: similarity is encoded by the position, while color is used to encode the similarity in cluster member aspect. The interpretation of our interlinked MDS representation is as follows: clusters being close to each other and share a similar color in one MDS projection are similar, hence redundant in both, the cluster member and subspace aspect *(C1.1)+(C1.2)*. Subspace clusters, which are close in the cluster member projection, but different in their coloring are similar in their cluster members, but different in their subspace topology (*C1.3*).

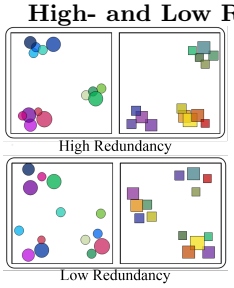
Computation of Coloring in MDS Projections.

The computation of our linked MDS projections is illustrated in Figure 3. First, the two MDS projections for the cluster member and subspace similarity are computed *independently* using a user-defined distance function. Afterwards, both projections are mapped *independently* on top of a perceptual linear 2D colormap [23]. A nearby location in the MDS projection (high similarity) is mapped to a similar color. Up to this point, the visual variables position and color are calculated independently and are not comparable between the two MDS plots. We can now apply the color information of the clusters in one MDS projection on top of the clusters in the opposite projection. By exchanging the semantic color mapping schemes between the two plots, the cluster member MDS can still indicate a (dis-)similarity in their cluster members (visually encoded by the point's location), but the coloring reflects the subspace similarity. Alike, the subspace similarity view reflects the dimension similarity by means of the points' locations, but allows perceiving the cluster membership similarities via the color encoding.

Interpretation of MDS Structures.

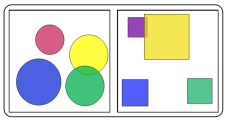
In the following, we give guidelines on how to interpret the visual appearance of the different MDS plots with respect to the presented quality criteria in Section 3.1.

High- and Low Redundancy (C1).



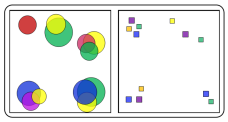
Similar objects have been clustered in similar subspaces: we can see groups of clusters in which colors are similar (top). Opposed to low redundancy (bottom), we can see groups of clusters, too, but either in different subspaces or with different objects. Thus, close clusters have dissimilar colors.

Big (Non-compact) Clusters (C3.1 + C3.4).



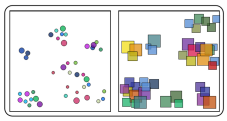
Clusters with many members or dimensions are illustrated by large glyphs in the MDS plots. Compactness can be additionally visualized by a more detailed glyph representation.

Too low-dimensional clusters (C3.3)



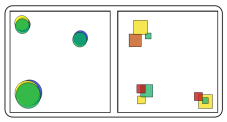
If the relevant subspace is too low dimensional the inferable insights are too trivial and no deeper conclusion about dependencies between the dimensions are possible. Too low- dimensional clusters can be seen by rather small glyphs in the subspace MDS projection. The is especially true for clusters with many cluster members *(C3.4)*.

Small Splinter Clusters (C1.3 + C3.2)



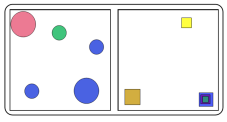
The result contains many small clusters indicated by small glyphs. These clusters do not provide a good generalizations of the data; general conclusions cannot be extracted.

Cluster Splitup in Subspaces (C1.3)



Split of clusters in subspaces: nearly identical object sets are clustered in different subspaces, indicated by largely overlapping cluster circles. Although this does not imply redundancy (colors are different, thus each cluster contains new information), it provides reason to suspect that these objects actually form a cluster in a single high-dimensional subspace.

Cluster Splitup in Objects (C3.2)



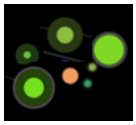
Split of clusters w.r.t. objects: a cluster might be divided into multiple clusters. We can discriminate between two cases: (1) a single cluster is partitioned in a single subspace (rare case) (c.f., blue circles), or (2) a cluster is partitioned and lives in differ-

ent subspaces, which is a typical case for projected clustering algorithms like Proclus [1].

Visual Enhancement and User Interaction.

Further visual encodings can be mapped on top of the enhanced MDS representation to iteratively add more details to the clusters. The additional information adds another level of complexity to the visualization. Therefore, the user can optionally add them, if needed for an analysis purpose.

Glyph Representation: The size of the points in the MDS projection can be mapped to e.g., the cluster- or subspace size. This representation allows assessing the characteristics *C3.3* and *C3.4* of all clusters.



Furthermore, additional cluster characteristics can be added to the glyph representation. For example, the compactness can be illustrated by the size of an inner circle in the glyph. A combination of multiple criteria in a pie-chart like fashion is also imaginable. A mouse over provides additional information for a cluster (e.g., size or members).

Linking and Brushing. We implemented a linking and brushing functionality between the two MDS projections. Moving the mouse over one cluster in the left projection highlights the same cluster in the right projection and vice versa. The user is able to apply a lasso selection and highlight all selected clusters in the opposite plot (c.f. Figure 1).

Selection and Filtering. Selected subspace clusters can be further analyzed by (*L2*) ClustDNA (Section 4.2 and (*L3*) DimensionDNA (Section 4.3). Additionally, the selected clusters can be reprojected into the MDS space to remove outlier-clusters which may distort the projection.

Ground Truth Comparison. Finally, SubEval allows to add potential ground-truth clusters to the projections. Using this feature, external evaluation methods can be enhanced by (1) comparing the similarity of all detected clusters with the ground truth and see for example, that multiple clusters are similar to the benchmark, and (2) the multi-level analysis of SubEval enables the user to visually analyze the structure of a ground truth cluster (c.f. DimensionDNA) to decide whether a ground truth cluster is actually appropriate.

4.2 ClustDNA: Comparison of Cluster Topologies in Subspace Projections

At the second analysis level of SubEval, a user is able to analyze and/or justify the redundancy of a small selection of subspace clusters (e.g., the four selected blue clusters in Figure 1 (1)). Our idea is to show for every selected cluster, both, all data objects and the cluster topology with a visualization, called ClustDNA. To understand the similarity between the different objects and the accordingly generated clustering structures, we rely again on a MDS projection. For every cluster we compute a projection in the respective subspace and assume that redundant subspace clusters result in similar MDS projections. Furthermore, we compare each subspace projection with a MDS projection containing the union of dimensions of all selected subspace clusters. We call the unified combination of dimensions SuperSpace. A comparison with these SuperSpace helps to decide whether a subspace of all dimensions results in a more profound cluster.

An example of ClustDNA can be found in Figure 4. Each selected subspace cluster is represented by a combination of two MDS projections: SuperSpace (left) and subspace of cluster (right). The cluster members are colored whereas

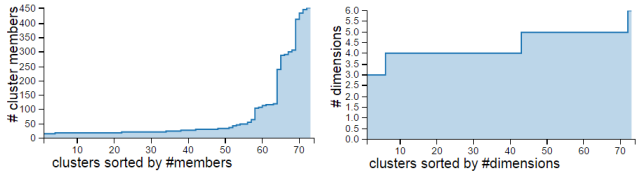


Figure 5: Distribution of the #of cluster members (left) and the #subspaces (right).

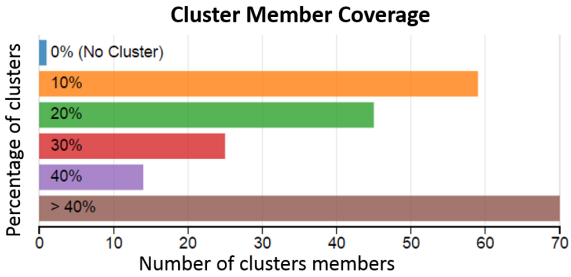


Figure 6: Bar charts to analyze the object coverage: a few objects are not clustered (blue), about 60 objects are a member in $1-10\%$ of the clusters and more than 70 objects are a member in more than 40% the clusters.

non-cluster members are represented in grey. The small glyph at the top indicates the dimensions of each subspace (red = subspace -, grey = SuperSpace dimensions).

4.3 DimensionDNA: In-Depth Analysis

At the third analysis level, a user needs to be able to analyze one particular selected cluster to identify good/bad clustering decisions of an algorithm. SubEval implements an enhanced Parallel Coordinates (PC) [18] visualization called DimensionDNA. Classical PC are combined with a heatmap to illustrate the data density of the entire dataset in each dimension (Figure 1 (right)). Each vertical bar represents one dimension. The minimum value of the dimension is mapped to the bottom of the bar, linearly scaled to the top (maximum value). The white-to-black colormap encodes the number of objects falling into a specific range (dark = many objects; bright = few objects). Records of a selected cluster are visualized as a connected line (red) among all dimensions of the dataset. The subspace dimensions are highlighted.

Using DimensionDNA, a user can analyze the compactness *(C3.1)* of the cluster members in the (subspace) dimensions in order to see whether a subspace cluster is valid. When selecting multiple clusters, the user is able to analyze the cluster's redundancy *(C1)* and separability *(C3.2)*. The underlying distribution of every dimension helps the analyst to inspect outliers or distortions that prevent an algorithm to identify clusters in particular dimensions.

4.4 Cluster Meta-Data Analysis

To provide additional information of detected subspace clusters (or a selection thereof), SubEval comprises several visualizations to analyze the clusters' meta-data:

Cluster- and Subspace Size Distributions: Figure 5 shows a line plot to assess the distributions of the *cluster size* (left) (c.f., *C3.3*) and *subspace size* (right) (c.f., *C3.4*). A user is able to see whether an algorithm produced rather small, large, or different sized subspace clusters.

Object Coverage Analysis: The bar-chart in Figure 6 is targeting *C2.1 -Object Coverage*, where we visualize the relationship between the number of (non-)clustered data objects. The non-clustered objects can be further investigated with the DimensionDNA plot, while the redundancy aspects of the object-to-cluster assignment (*C1*) can be analyzed by interactions on the bar chart. It shows the number of objects (x-axis) which do not belong to any cluster (blue bar), and the number of members being part in p%, of the clusters. The more this bar-chart is shifted to the bottom, the more often specific cluster members occur in multiple clusters.

Dimension Coverage Analysis *C2.2* is targeted with an interactive bar-chart showing how many subspaces a dimension is allocated. The user can subsequently investigate dimensions, which occur frequently or never in any subspace, with the DimensionDNA plot.

Dimension Co-occurrence: Besides the coverage aspect, the user is able to analyze, which dimensions co-occur in the subspaces by choosing one or multiple dimensions. The chart is updated by filtering for subspaces containing the selected dimensions.

All charts can be interactively filtered. A selection of one e.g., dimension in the coverage analysis, or clusters of a specific size will update all other visualizations accordingly, thus allowing an analyst to concentrate on clusters of interest.

5. EXPERIMENTS

We describe two use cases to show the usefulness of SubEval to visually evaluate the quality of subspace clusterings. SubEval is implemented in Java/JavaScript in a server-client fashion using d3.js[1] for the visualizations. In the supplementary material[2] we provide a video and give the user the opportunity to explore the use cases with SubEval.

Use Case 1: Redundancy Analysis.

In the first use case, we want to show the usage of SubEval for the detection and analysis of redundancy. We apply the well-known Clique [2] algorithm to the real-world Glass dataset with 214 objects and 9 dimensions. Clique is a grid-based algorithm which is known to detect many redundant clusters. For the Glass dataset, 132 subspaces[3] are detected.

In the first step, we analyze the cluster member coverage of our result (Figure 6). Except for one outlier (blue bar) we can quickly see that all data objects belong to at least one cluster. However, more than 70 data objects (30% of the dataset) are part of more than 40% of the clusters resulting in a noticeable degree of member overlap in the clusters.

The results of the inter-linked MDS projection can be found in Figure 1. We can see a large group of bigger clusters in the top left corner of the cluster member similarity projection. The clusters of the group share a common clustering topology, but have a different color encoding. This corresponds to similar clusters occurring in subspaces of different dimensions. Besides the smaller splinter clusters that occur in different (larger-dimensional) subspaces, the user is faced with four larger clusters (blue shaded on the left side). These clusters seem to have similar cluster members in similar subspaces and thus can be suspected redundant. We analyze this potential redundancy further with ClustDNA as shown in Figure 4. In the dimension glyph on the top, we can see, that all four clusters share most of their dimensions. Another interesting observation it that the first and second clustering have an almost identical cluster topology which is visible through a similar MDS projection. The cluster on the right comprise only a single dimension in which all cluster members are almost identical. A user can conclude that the selected clusters are truly redundant. It would be sufficient to only report the first cluster without loosing much knowledge about the data.

Finally, we select one of the redundant clusters and investigate the dataset distribution with the DimensionDNA, as shown in Figure 1 (3). We can see that the cluster members are compact in the subspace dimensions *dim1,4,5*, but also in non-subspace dimensions *dim0,2,3,5*, and *dim7*. Hence, an analyst may question, why the aforementioned dimensions are not part of a subspace. In summary, a user can quickly see that the clustering result contains a few larger subspace clusters, but also many smaller splinter clusters and a few redundant clusters as described above. The shown results can be attributed to the bottom-up strategy of Clique, which is known to produce a large number of redundant clusters. An analyst may either change the parameter settings or apply a different subspace clustering algorithm.

Use Case 2: Splinter Cluster Analysis.

In the second use case, we analyze a good performing subspace clustering algorithm (Inscy [4]) on the Vowel dataset as experimentally identified in [24]. The dataset contains 990 object, described by 10 dimensions[4]. Inscy is an algorithm with a redundancy elimination strategy.

According to the experiments in [24], the algorithm performs well on the dataset with good external performance measures (compared to a ground truth). When analyzing the clustering result with SubEval, we made the following observations: The size of the subspaces is homogeneous with a dimensionality between three and six dimensions. However, the number of cluster members varies significantly. Many clusters contain less than 30 members and only a few clusters have more than 300 members as shown in Figure 5. When encoding this information into the inter-linked MDS projection (c.f. Figure 7), we can see that the clustering contains a large number of small splinter clusters with a variety of different colors. This means that in a large number of subspaces, the algorithm detected small, less expressive clusters. The group of bigger clusters on the bottom left is apart from the splinter clusters and contains significantly more cluster members, hence a more general representation of the data. As visible from the similar coloring, there are many redundant clusters, which can be verified in the detail analysis. We select one of the clusters, as shown in Figure 7 (1), and analyze the data distribution with the DimensionDNA (shown in Figure 7 (3)). The subspace contains three dimensions. *dim3*, however, does not seem to be compact and an analyst may question why this dimension is part of the subspace. It is therefore interesting that the algorithm performed well on the dataset according to the experiments in [24]. Based on our findings, an algorithm expert could improve the clustering results by a careful adjustments of the parameters.

[1] https://d3js.org/

[2] http://www.subspace.dbvis.de/idea2016

[3] Parameter of Clique for use case 1: -XI 10 -TAU 0.2

[4] Parameter of Inscy for use case 2: -gS 10 -mS 16 -de 10.0 -m 2.0 -e 8.0 -R 0.0 -K 1

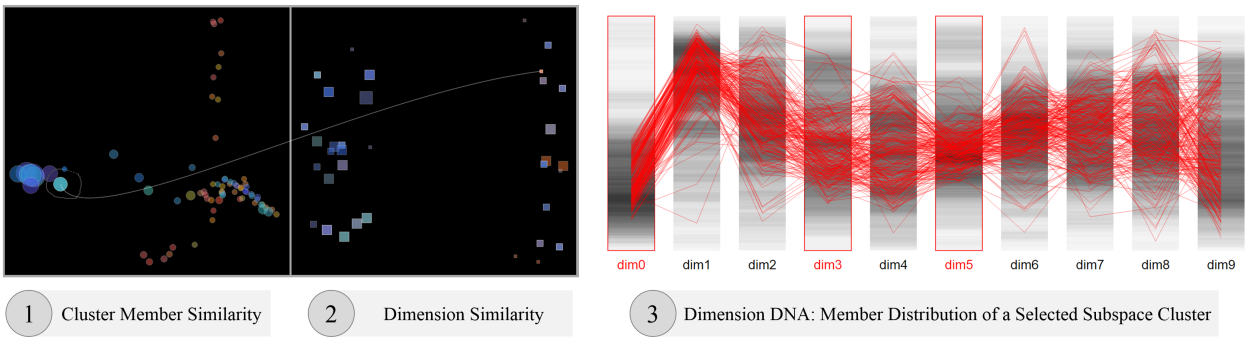


Figure 7: Use Case 2: (1) + (2) Group of large clusters with similar subspaces (blue group left) and many small splinter clusters with different colors (=different subspaces) (left). One cluster is selected for detailed analysis. (3) DimensionDNS: Visualizing the distribution of cluster members of the selected cluster. An analyst may wonder why the outliers in *dim3* and *dim5* are part of the cluster.

6. DISCUSSION AND FUTURE WORK

While our technique has proven useful for an efficient and effective visual comparison of subspace clusters regarding certain quality aspects, we identify areas for further research.

Alternative Visual Design. The inter-linked MDS projection between the cluster member and dimension similarity of subspace clusters may be difficult to read and requires some training for unfamiliar users. The same is true for the ClustDNA visualization. Furthermore, MDS projections face generally the problem of overlapping points and might not show the actual similarity between all combinations of points as discussed below. Therefore, we are planning to improve the MDS projection and also work on different visual representations for the overview of subspace clusterings. Node-link diagrams as introduced in [30] may be an interesting starting point to this end.

MDS projects data points into a 2D space by preserving the pair-wise distances between all data points as well as possible. Depending on the distance distributions, the 2D projection may not reflect the actual relationships correctly. Then, objects appearing close in the projection might be dissimilar in their original space, and far apart objects may be similar. Independent of the quality, a MDS projection is typically interpreted by a user as it is, without considering a possible error which lead to wrong analysis results. SubEval already provides methods for drill-down to justify presumptions in a different view. Later, we also want to address the quality of the MDS projection by visualizing the difference between the similarity in the MDS projection and the real data characteristics, or rely on further techniques for visualization of projection quality [25].

SubEval is designed to analyze one subspace clustering result at a time. A comparative evaluation of several clustering results would be beneficial to compare the influence of minor changes in the parameter settings. We plan to extend SubEval for a comparative analysis of multiple clusterings.

Application to Related Approaches. The analysis goal of subspace clustering differs significantly from other analysis techniques like *subspace outlier detection* (SOD) [33] and *subspace nearest neighbor search* (SNNS) [19]. While SOD tries to identify subspaces in which outliers exist, SNNS identifies nearest neighbor sets to a given query in different subspaces. Although the analysis goal differs, both techniques share the same evaluation challenges like subspace clustering, i.e., redundant subspaces and results (outliers or nearest neighbors). In the future, we want to extend SubEval for the quality assessment of SOD and SNNS. For the inter-linked MDS projection we need to develop quality measures for the redundancy definition. DimensionDNA can be applied to both techniques. Also, we need to develop visualizations to access the meta-data of the respective analysis.

SubEval is designed for the quality assessment of subspace clusterings, however, it can also be used for the evaluation of full-space clusterings, particularly with partially overlapping clusters. For the MDS projection, an appropriate measure is needed to compute the similarity between clusters. One option is to compute the distance between cluster centroids or the pair-wise distances between all cluster members. DimensionDNA and ClustDNA can also be applied to investigate cluster topologies and member distributions.

Open Source Framework. SubEval is part of SubVA (Subspace Visual Analytics), a novel open-source framework for visual analysis of different subspace analysis techniques. Besides providing implementations of recently developed visualizations, such as SubVis [20], SubVA integrates the well-known OpenSubspace framework [24] as a module, allowing analysts to apply the most commonly used subspace clustering algorithm to a given dataset. We will distribute the framework on our website[5] and provide the source code in the supplementary material.

7. CONCLUSION

This paper presented SubEval, a subspace evaluation framework for the simultaneous assessment of several quality characteristics of one subspace clustering result. SubEval combines expressive visualizations with interactive analysis and domain knowledge, and complements, potentially advancing standard evaluation procedures with a more comprehensive, multi-faceted approach. We summarized state-of-the-art evaluation methods for subspace clustering algorithms and showed that, besides classical measures, visualizations can be an insightful approach to the evaluation and understanding of subspace clustering results. We also outlined ideas for extensions of our approach.

Acknowledgments

The authors are grateful for the valuable discussion and work that contributed to the underlying framework of J. Kosti, F.

[5]http://www.subva.dbvis.de

Dennig, M. Delz, and S. Wollwage. We wish to thank the German Research Foundation (DFG) for financial support within the projects A03 of SFB/Transregio 161 "Quantitative Methods for Visual Computing" and DFG-664/11 "SteerSCiVA: Steerable Subspace Clustering for Visual Analytics".

8. REFERENCES

[1] C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast Algorithms for Projected Clustering. *SIGMOD Rec.*, 28(2):61–72, 1999.

[2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec.*, 27(2):94–105, 1998.

[3] I. Assent, R. Krieger, E. Müller, and T. Seidl. VISA: visual subspace clustering analysis. *SIGKDD Explor. Newsl.*, 9(2):5–12, 2007.

[4] I. Assent, R. Krieger, E. Müller, and T. Seidl. INSCY: Indexing subspace clusters with in-process-removal of redundancy. In *Proc. of ICDM*, pages 719–724, 2008.

[5] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[6] F. Bendi, R. Kosara, and H. Hauser. Parallel sets: visual analysis of categorical data. In *Proc. of InfoVis*, pages 133–140, 2005.

[7] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *Database theory — ICDT'99*, pages 217–235, 1999.

[8] S. Bremm, T. von Landesberger, J. Bernard, and T. Schreck. Assisted descriptor selection based on visual comparative data analysis. *CGF*, 30(3):891–900, 2011.

[9] S. Bremm, T. von Landesberger, M. Heß, T. Schreck, P. Weil, and K. Hamacher. Interactive Comparison of Multiple Trees. In *Proc. of VAST*, 2011.

[10] N. Cao, D. Gotz, J. Sun, and H. Qu. Dicon: Interactive visual analysis of multidimensional clusters. *TVCG*, 17(12):2581–2590, 2011.

[11] T. F. Cox and M. A. Cox. *Multidimensional scaling*. CRC press, 2000.

[12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of SIGKDD*, pages 226–231, 1996.

[13] I. Färber, S. Günnemann, H.-P. Kriegel, P. Kröger, E. Müller, E. Schubert, T. Seidl, and A. Zimek. On Using Class-Labels in Evaluation of Clusterings. In *Workshop at SIGKDD*, 2010.

[14] S. Günnemann, I. Färber, E. Müller, I. Assent, and T. Seidl. External Evaluation Measures for Subspace Clustering. In *Proc. of CIKM*, pages 1363–1372, 2011.

[15] S. Günnemann, I. Färber, H. Kremer, and T. Seidl. CoDA: Interactive cluster based concept discovery. *Proc. of VLDB Endowment*, 3(1-2):1633–1636, 2010.

[16] S. Günnemann, H. Kremer, I. Färber, and T. Seidl. MCExplorer: Interactive Exploration of Multiple (Subspace) Clustering Solutions. In *Data Mining Workshops at ICDMW*, pages 1387–1390, 2010.

[17] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 3rd edition.

[18] J. Heinrich and D. Weiskopf. State of the art of parallel coordinates. 2013:95–116, 2013.

[19] M. Hund, M. Behrisch, I. Färber, M. Sedlmair, T. Schreck, T. Seidl, and D. Keim. Subspace Nearest Neighbor Search - Problem Statement, Approaches, and Discussion. In *Proc. of SISAP*, pages 307–313. 2015.

[20] M. Hund, D. Böhm, W. Sturm, M. Sedlmair, T. Schreck, T. Ullrich, D. A. Keim, L. Majnaric, and A. Holzinger. Visual analytics for concept exploration in subspaces of patient groups. *Brain Inf.*, pages 1–15, 2016.

[21] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering pattern-based clustering, and correlation clustering. *ACM TKDD*, 3(1), 2009.

[22] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. In *Proc. of ICDM*, pages 911–916, 2010.

[23] S. Mittelstädt, J. Bernard, T. Schreck, M. Steiger, J. Kohlhammer, and D. A. Keim. Revisiting Perceptually Optimized Color Mapping for High-Dimensional Data Analysis. In *In Proc. of EuroVis*, pages 91–95, 2014.

[24] E. Müller, S. Günnemann, I. Assent, and T. Seidl. Evaluating Clustering in Subspace Projections of High Dimensional Data. *Proc. of VLDB Endowment*, 2(1):1270–1281, 2009.

[25] T. Schreck, T. von Landesberger, and S. Bremm. Techniques for precision-based visual analysis of projected data. *Information Visualization*, 9(3):181–193, 2010.

[26] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of Visual Languages*, pages 336–343. IEEE, 1996.

[27] M. Steiger, J. Bernard, S. Mittelstädt, S. Thum, M. Hutter, D. A. Keim, and J. Kohlhammer. Explorative Analysis of 2D Color Maps. In *Proc. of Computer Graphics, Visualization and Computer Vision*, volume 23, pages 151–160, 2015.

[28] A. Tatu, F. Maaß, I. Färber, E. Bertini, T. Schreck, T. Seidl, and D. A. Keim. Subspace Search and Visualization to Make Sense of Alternative Clusterings in High-Dimensional Data. In *Proc. of VAST*, pages 63–72, 2012.

[29] A. Tatu, L. Zhang, E. Bertini, T. Schreck, D. A. Keim, S. Bremm, and T. von Landesberger. ClustNails: Visual Analysis of Subspace Clusters. *Tsinghua Science and Technology*, 17(4):419–428, 2012.

[30] C. Vehlow, F. Beck, P. Auwärter, and D. Weiskopf. Visualizing the evolution of communities in dynamic graphs. *Comput. Graph. Forum*, 34(1):277–288, Feb. 2015.

[31] M. O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3-4):194–210, 2002.

[32] M. O. Ward, G. Grinstein, and D. Keim. *Interactive Data Visualization: Foundations, Techniques, and Applications*. A. K. Peters, Ltd., 2010.

[33] A. Zimek, E. Schubert, and H.-P. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5):363–387, 2012.

Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models

Josua Krause
New York University
New York, NY, USA
josua.krause@nyu.edu

Adam Perer
IBM T.J. Watson
Research Center
Yorktown Heights, NY, USA
adam.perer@us.ibm.com

Kenney Ng
IBM T.J. Watson
Research Center
Yorktown Heights, NY, USA
kenney.ng@us.ibm.com

ABSTRACT

Understanding predictive models, in terms of interpreting and identifying actionable insights, is a challenging task. Often the importance of a feature in a model is only a rough estimate condensed into one number. However, our research goes beyond these naïve estimates through the design and implementation of an interactive visual analytics system, *Prospector*. By providing interactive partial dependence diagnostics, data scientists can understand how features affect the prediction overall. In addition, our support for localized inspection allows data scientists to understand how and why specific datapoints are predicted as they are, as well as support for tweaking feature values and seeing how the prediction responds. Our system is then evaluated using a case study involving a team of data scientists improving predictive models for detecting the onset of diabetes from electronic medical records.

Keywords

interactive machine learning; predictive modeling; partial dependence; visual analytics; model visualization

This paper was published before. The original manuscript can be found at:
http://perer.org/papers/adamPerer-Prospector-CHI2016.pdf.

IDEA '16 San Francisco, California, USA

DOI: 10.475/123_4

Interactive Exploration for Domain Discovery on the Web

Yamuna Krishnamurthy
yamuna@nyu.edu

Kien Pham
kien.pham@nyu.edu

Aécio Santos
aecio.santos@nyu.edu

Juliana Freire
juliana.freire@nyu.edu

Tandon School of Engineering
New York University

ABSTRACT

As the volume of information on the Web grows, it has become increasingly difficult to find web pages that are relevant to a specific domain or topic. In this paper, we explore the general question of how to assist users in the domain discovery process. Domain discovery entails the translation of a user's information needs and conceptual view of a domain into a computational model that enables the identification and retrieval of relevant content from the Web. We discuss the challenges and propose an initial approach based on exploratory data analysis that combines techniques from information retrieval, machine learning and data mining to streamline domain discovery. We implemented the approach in an open-source tool and present the results of a preliminary evaluation.

Keywords

Exploratory data analysis, exploratory search, human interaction, visualization, information retrieval, text mining, focused crawling

1. INTRODUCTION

Domain discovery is the process through which a user identifies and retrieves information and sources from the Web that are relevant for a specific information need. Consider the following scenario. Analysts at a law enforcement agency that is tasked with investigating and preventing the illegal use and trafficking of firearms regularly search the Web to discover and track potentially illicit activities. They want to find suspicious brokers and online stores, forbidden weapons for sale, reports of stolen weapons, and leads into trafficking activities. While they have a clear idea of the information they need, finding this information on the Web is challenging. They often start by issuing queries to Google or Bing using keywords such as "AR15" or "no paperwork", which based on their prior knowledge, provide a good indication of illegal weapon sales. While search engines provide broad coverage of the Web, for domain specific searches they have an important drawback: they return a very large number of irrelevant results. Figure 1 shows results from Google for the queries `ar15 no paperwork` and `sell ar15 no paperwork`. Most of these results are not related to the sale of the weapons with no paperwork.

The experts need to analyze the results of the search either by reading the snippets returned by the search engine or the actual pages. When they identify a relevant page which contains information like phone numbers, user ids in forums and images, they bookmark or save it locally. As they perform multiple investigations, it is easy to lose the search context. Moreover, content on the Web is very dynamic: existing pages change or are deleted, and new pages are added at a high rate. Thus, just keeping track of URLs visited is not sufficient. To maintain the information up-to-date and discover new relevant content, the expert must continuously query the search engine. This process is time consuming and tedious.

Another challenge lies in formulating keyword queries. While these queries are simple, selecting the right terms for a domain-specific search can be daunting. The representation of a given domain on the Web can differ from what an analyst expects, and the analyst may not be aware of certain nuances. During exploration, by examining pages returned by a search engine, the analyst can discover other related terms. For example, as illustrated in Figure 1, another term that appears in the search results that is potentially related to illegal activity is `background check`. Currently, analysts have to read the pages, manually record the keywords of interest they discover, and later use these keywords as additional queries. This clearly does not scale.

The simplicity of keyword queries is a strength and also a limitation. In theory, an analyst could improve the relevance of the results by issuing more specific queries. For example, an analyst could search all forums and user ids associated with the sale of a particular illegal weapon. Or when she finds a user in a forum who posted an ad for a gun without paperwork, she would like to check whether this user is active in other forums. Such queries cannot be expressed using the interfaces supported by search engines.

These challenges are commonplace in many different tasks, from tracking criminal activities to understanding how research areas evolve over time.

Contributions. To address these challenges, we developed a visual analytics framework for interactive domain discovery that augments the functionality provided by search engines to support analysts in exploratory search. The framework (1) supports exploratory data analysis (EDA) [27] of web pages, and (2) translates the analyst's interactions with this data into a computational model of the domain of interest. By organizing and summarizing the search results,

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA '16)
August 14th, 2016, San Francisco, CA, USA.

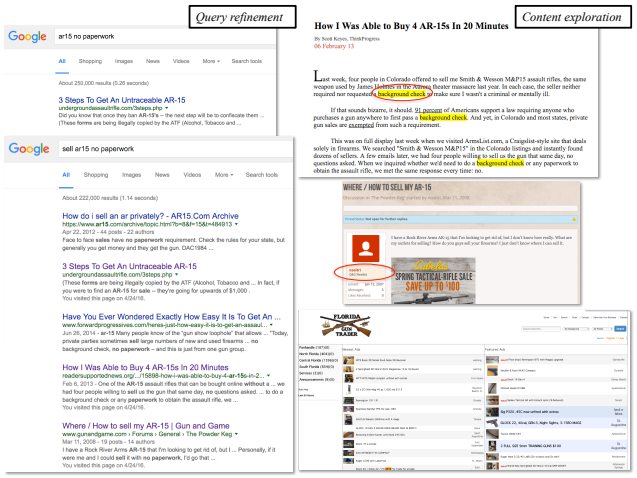


Figure 1: Searching for assault rifles sold with out proper paperwork

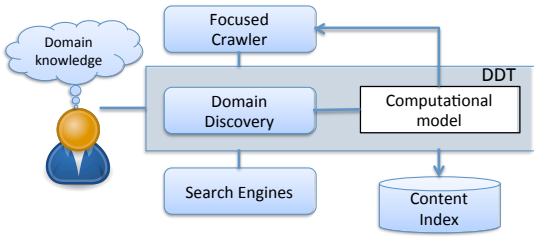


Figure 2: Interactive Domain Discovery

the framework helps users better understand and analyze the retrieved content as well as provide feedback. By clustering the retrieved pages and grouping similar pages together, it simplifies the process of selecting and annotating the pages. It also automatically extracts important keywords and phrases in the pages. These not only serve as a summary of the content, but also as suggestions for new queries to be issued. In the course of exploration, the search context is maintained: queries and their results are persisted, allowing users to revisit and analyze the content. The annotations provided by the users regarding the relevance of pages is used to build the domain model, a computational specification of the domain, which can then be used to configure a focused crawler [2, 4]. The focused crawler, in turn, provides a scalable mechanism to retrieve additional pages which feeds back into the domain discovery process. Figure 2 illustrates the interactive domain discovery process.

We have implemented the framework in the Domain Discovery Tool (DDT),[1] an open-source system that implements these mechanisms. We also report the results of a preliminary user evaluation.

2. RELATED WORK

Search user interfaces have been extensively studied and implemented. Hearst [8] provides a comprehensive summary of work on search interface design. She also discusses the broader problem of *sensemaking* [16, 20, 21], the "iterative process of formulating a conceptual representation from a large volume of information", and argues that "the standard Web search interface does not do a good job of supporting the sensemaking process". Tools such as Sandbox [28] provide an interface for advanced analysis of the information gathered, from various sources, by allowing free-form organization of retrieved results. However, it misses an important step in sensemaking: the collection of a good set of resources, representative of the domain. Search is an integral part of this domain discovery process which enables analysis and information extraction. The framework we propose provides this missing step towards sensemaking.

Vertical search engines focus on a specific segment of online content. For example, Google Scholar[2] stores information about scientific publications and Yelp[3] helps users find information about local businesses. These systems have several benefits over general search engines. Notably, because of their limited scope, they return more relevant results and lead to higher precision. In addition, they support domain-specific tasks. For example, in Google Scholar, it is possible to search for papers written by a given author. These vertical engines, however, are expensive to build and hence they are available only for broad topics of wide interest. Our framework allows exploration of any given domain, and is a step towards lowering the costs of building vertical search engines for any domain available on the Web.

Focused crawling [2, 4] has been proposed as a scalable mechanism to gather data about specific domains from the Web. In order to bootstrap a focused crawler, it is necessary to provide a set of seed URLs that serve as the starting points for the crawl, and a page classifier that can decide whether a retrieved page is relevant or not. While these systems are effective and address many of the challenges discussed previously, they require substantial human input. Collecting a set of positive and negative examples to train classifiers that recognize the target concept is time consuming; and as new pages are obtained by the crawler, the classifier needs to be iteratively refined. Not surprisingly, focused crawlers have not been widely adopted. The framework proposed in this paper helps to solve the crawler bootstrapping problem by helping the user to acquire seed URLs and build models to classify Web pages.

Domain discovery requires exploration of text corpora gathered from the Web. *Interactive text mining* techniques help address this problem. Over the years there has been substantial research on various aspects of interactive text mining, for both web and other documents, such as clustering [5, 11, 12, 15, 17, 23, 30], topic modeling [10, 29], and semantic analysis [25]. Interactive applications like STREAMIT [1] and i-GNSSMM [14] have attempted to bring some of this work together to analyze web documents. However, STREAMIT assumes the existence of an external continuous source of documents. The user cannot add documents to this source during exploration using STREAMIT. It does not allow users to annotate the documents and create their own clusters – users can only tweak certain parameters to adjust the system's clustering algorithm. i-GNSSMM extracts the topic graph from a collection of web pages. Although this could be a useful representation of the content it is not always appropriate for a user's information seeking needs.

[1]DDT is available at https://github.com/ViDA-NYU/domain_discovery_tool. For demos see: https://youtu.be/XmZUnMwI10M, https://youtu.be/YKAI9HPg4FM, https://youtu.be/HPX8lR_8QS4.

[2]http://scholar.google.com
[3]http://www.yelp.com

There are also a number of text mining software packages.[4] Since these require technical expertise to configure and use them, they are out of reach for domain experts without training in computing.

Recent work in *dynamic search* [31] improves search over time by learning from the user's interaction with the system. This work is complementary to our effort. We may leverage dynamic search to improve the search and filtering of documents in our framework. More closely related to our framework is the *intent modeling* work by Ruotsalo et al. [19]. But this work uses only the feedback of important keywords to model the intent of the user. It does not allow the users to provide feedback on the relevance of documents or group them as they see fit.

3. DESIDERATA OF INTERACTIVE DOMAIN DISCOVERY

This work was originally motivated by challenges of domain specific search that were encountered as part of the DARPA Memex program.[5] In this project, we have interacted with experts with a wide range of information needs in different domains, including human trafficking, sale of explosives, illegal weapons and counterfeit electronics, micro cap fraud and patent trolls. In what follows, we discuss the desiderata for domain discovery based on our interactions with these experts, their feedback on existing state-of-the-art tools, and information needs.

Translation of conceptual definition of a domain into a computational model. Domain definition and discovery can be viewed as the iterative process of mapping an expert's conceptual view of a domain into a set of artifacts available on the Web (e.g., web sites, web pages, terms, phrases and topics). Since the human is clearly the biggest bottleneck in this process, we need usable and scalable mechanisms that guide and support the user. Capturing the domain definition as a computational model enables this process to scale: with such a model, automated processes can be deployed to retrieve relevant information.

Data gathering. Analysts use various information retrieval mechanisms to collect relevant data for subsequent analysis. Some of the common mechanisms include, but are not limited to, web searches to locate new information and uploading already known relevant web pages. As they identify relevant pages, they often crawl forward and backward in an attempt to find additional content. A tool for domain discovery should support these mechanisms and make them easy to use.

Maintaining search context and capturing user feedback. Search engines treat each query independently. While there is a notion of *session* which either refers to a specific period of time or the linear chain of links followed, in domain discovery the context should take the domain into account. This would be an aggregation of sessions of exploration of that domain. The history and bookmarking mechanisms allow users to save some of the search context, but it is hard to reason about and revisit previously viewed content. This is a major roadblock for domain discovery. The search context should include queries issued, pages retrieved, indications provided by users regarding the relevance of both pages and keywords extracted from them. This information should be readily available and easily interrogated.

Summarizing search results. The simple list of links with snippets provided by existing search engines is not sufficient for quick analysis and annotation of pages especially when the number of results returned is large. The list fails to provide an overview of the results. As we discuss in Section 4, we explored different techniques to better summarize the information.

Streamlining annotations. An important component of domain discovery is user feedback regarding the relevance of pages and sites. This feedback is essential to: Guide users in the process of understanding a domain and help them construct effective queries to be issued to a search engine; and configure focused crawlers [4] that efficiently search the Web for additional pages on the topic by using the feedback to build page classifier models and gather seed URLs.

Exploring and Filtering Results. Once a set of pages is gathered for a particular domain, the experts, as part of their investigation, benefit from exploring subsets of these results. Useful filtering mechanisms include, for example, filter by keywords or specific time period.

Minimal setup and configuration. Analysts working on domain discovery do not necessarily have technical expertise to setup and configure tools and applications. They usually require systems that have a simple, intuitive, visual and interactive interface that has a very low learning curve with minimal or no configuration required.

4. DOMAIN DISCOVERY TOOL

Informed by the desiderata described in Section 3, we designed a framework to support domain discovery. The framework aims to support users in the construction of a computational model of their conceptual view of the domain. To achieve this, it includes several mechanisms that aid analysts to explore, interact with and learn about the domain from the Web content retrieved, and that also gather user feedback . The mechanisms, which we describe below, combine techniques from data mining, machine learning and information retrieval, and their results are presented to the expert through interactive visualizations. They were implemented in Domain Discovery Tool (DDT), whose user interface is shown in Figure 3.

4.1 Data Gathering and Persistence

Search context is maintained by persisting it in an index[6], created for each domain, where all the domain specific exploration activities are stored. Domain experts can use a variety of methods to make pages of interest available for analysis through DDT.

Querying the Web. DDT allows users to query the Web using Google or Bing. They can leverage the large collections already crawled by the search engines to discover interesting pages across the Web using simple queries. Since search engines only return the URLs and associated snippet, DDT downloads the HTML content given the URLs and stores it in the selected domain's index. This content can be used later for analysis of the domain and also as seeds for fo-

[4]https://en.wikipedia.org/wiki/List_of_text_mining_software

[5]http://www.darpa.mil/program/memex

[6]Our prototype makes use of an elastic search index: https://www.elastic.co/products/elasticsearch

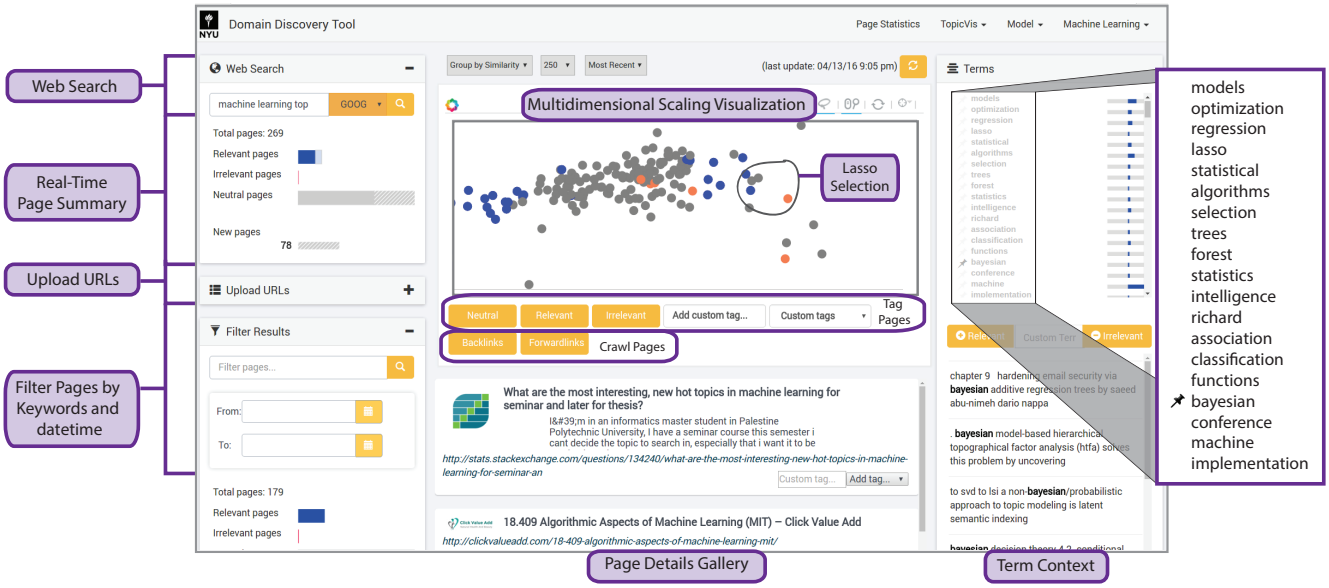


Figure 3: Domain Discovery Tool Interface Components

cused crawlers. Since downloading a large number of pages (including the raw HTML content) takes significant time, DDT performs this operation in the background.

Uploading URLs. In our interviews with experts and use cases we explored, experts often have a set of sites (or pages) they know are relevant. Therefore, it is important to provide a mechanism for incorporating this background knowledge. DDT allows users to provide URLs either through the input box provided or by uploading a file containing a list of URLs. DDT then downloads the pages corresponding to these URLs and makes them available through its interface.

Forward and Backward Crawling. While users can manually follow links forward and backward from the pages they explore, this process is tedious. DDT automates these tasks. Given a page, crawling backwards retrieves the backlinks (the resources that contain a link to the selected page) of that page and then downloads the corresponding pages. Forward crawling from a selected page retrieves all the pages whose links are contained in that page. The intuition behind the effectiveness of these operations is that there is a high probability that the backlink of a page and the page itself will contain links to other pages that are relevant.

4.2 Visual Summary of Search Results

To provide the analyst an overview of the pages they have explored, DDT summarizes them visually in different ways.

4.2.1 *Multidimensional Scaling*

DDT uses multidimensional scaling (MDS) (see Figure 3) for visualizing the retrieved content. Instead of displaying just a list of snippets, DDT applies MDS to create a visualization of the retrieved pages that maintains the relative similarity and dissimilarity of the pages. This allows the user to more easily select (e.g., using lasso selection), inspect and annotate a set of pages.

MDS is currently achieved by principal component analysis (PCA) [26] of the documents. Since initially all pages are unlabeled we need an unsupervised learning method to group the pages by similarity. Note that other unsupervised clustering methods such as K-Means [7] and hierarchical clustering [18] can be used, and we plan to explore these in future work.

To improve scalability, instead of using the sparse *document*×*term* (words in a document) matrix of TF-IDF [22] as the input to the scaling algorithms, we use Google's word2vec [13] pre-trained vectors that were trained on part of Google News dataset (about 100 billion words). The model contains a 300-dimensional vector for each word in a set $W2V$ of 3 million words and phrases. The archive is available online as GoogleNews-vectors-negative300.bin.gz[7].

We convert each document using the word vectors as follows. Let $D = \{d_1, d_2, ..., d_n\}$ be the set of documents to be scaled. $\forall d \in D$ *let* $W_d = \{w_1, w_2..., w_m\}$ where W_d is the set of all words in the document (after removing stopwords). The word2vec archive provides a 300 dimension vector $V = \{v_1, v_2, ..., v_{300}\}$, $\forall w \in \{W_d \cap W2V\}$. So $\forall d \in D$ the vector corresponding to $d = \frac{\sum V_w,\ \forall w \in \{W_d \cap W2V\}}{|\{W_d \cap W2V\}|}$. This generates an input matrix of dimension $n \times 300$, where n is the number of documents, which is much smaller than the original *document* × *term* matrix. By mapping words to word vector representations, we saw a significant improvement in the speed of scaling computation, and also got the benefits of a word vector representation trained on a large text corpus.

4.2.2 *Descriptive Statistics*

Real-time Page Statistics. As new pages are retrieved, DDT dynamically updates the following statistics:

- *Total pages* - total number of pages in the domain
- *Relevant pages* - number of pages marked as relevant
- *Irrelevant pages* - number of pages marked as irrelevant
- *Neutral pages* - pages that have yet to be annotated

[7]https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing

- *New pages* - number of pages downloaded in the background since last update. This indicates that there are new pages yet to be analyzed.

Page Statistics Dashboard. This dashboard, shown in Figure 4, displays various statistics over the entire content in the domain such as the distribution summary of sites, the distributions and intersections of the search queries issued, summary of page tags and their intersections and number of pages added to the domain over time. This provides the user a map of the domain represented by the retrieved pages.

Topic Distribution Dashboard. This dashboard, shown in Figure 5, visualizes the various topics contained in the domain. The topics are generated using the Topik[8] topic modeling toolkit. Topics can be generated using either LDA [3] or PLSA [9]. Visualization of the topics is done with LDAvis [24]. It shows the topics, the overlap of topics and the most frequent words contained in each topic.

4.2.3 *Keywords and Phrases Extraction*

The keywords and phrases extracted from the pages displayed in the MDS window are shown in the Terms window. They provide a summary of the content of the result pages from which the analyst can learn new information about the domain and use some of the keywords and phrases as new search queries to retrieve additional pages from the Web.

An example is shown in the zoomed region in Figure 3, which shows important terms for the "*Machine Learning*" domain. The initial set of keywords and phrases (bi-grams and tri-grams) displayed are the ones with high TF-IDF [22] in the retrieved pages. But as the pages are annotated, the keywords and phrases are selected from the pages that are annotated as relevant.

When the user hovers the mouse over a term, snippets of result pages that contain the corresponding keyword or phrase are shown below the Terms window. This helps to better understand the context in which the keyword and phrase appear.

4.3 User Annotations

DDT allows users to provide feedback for both pages and terms extracted. In addition to marking individual pages, users can select a group of documents for analysis and mark them as relevant (or irrelevant). Users may also annotate pages with user-defined tags. These tags are useful to define sub-domains, for example, in the "*Machine Learning*" domain we can have sub-domains like "*Deep Learning*" and "*Generative Models*"

Users can also mark the keywords and phrases extracted by DDT as relevant or irrelevant. Based on the relevant terms, the system re-ranks the untagged keywords and phrases, by relatedness to the relevant terms using Bayesian sets [6]. This brings in more related terms and phrases that help the user both understand the domain further and formulate new search queries. Given a query consisting of a few items of the cluster, the Bayesian sets algorithm retrieves more items belonging to that cluster. It achieves this by computing a score for each item, that indicates how related it is to the query cluster. We modeled our ranking of untagged terms and phrases, based on a few tagged terms and phrases, to this setting. The terms and phrases that are marked as relevant by the user make the query cluster. Each term or phrase is represented by a binary vector of all documents in the corpus. The binary value 1 indicates that the term or phrase occurs in the corresponding document and 0 otherwise. So we have two sparse binary matrices as inputs to the Bayesian sets algorithm, (1) *query terms* $\times$ *documents* and (2) *untagged terms* $\times$ *documents*. The output is a list of the untagged terms ranked in the decreasing order of their score. We chose to use Bayesian sets as it computes the score exactly using a single sparse matrix multiplication, making it possible to apply the algorithm to very large datasets, which in our case is the large vocabulary of the corpus.

[8]http://topik.readthedocs.io/en/latest/

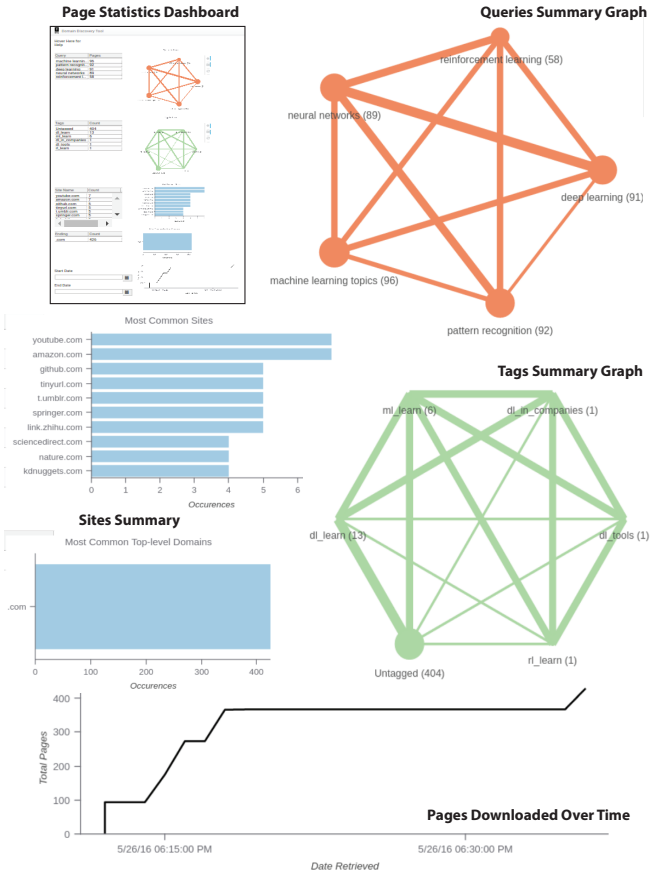


Figure 4: Page Statistics Dashboard

Users may also integrate background knowledge by adding custom keywords and phrases. To guide users and provide them a better understanding of the importance and discriminative power of the extracted terms, DDT shows the percentage of relevant and irrelevant pages the keyword or phrase appears in.

4.4 Domain Model and Focused Crawling

By using the pages marked relevant and irrelevant as positive and negative examples, respectively, DDT supports the construction of a page classifier which serves as a model for the domain. This classifier together with a set of seeds (relevant pages) can be used to configure a focused crawler. In DDT, we support the ACHE [2] crawler.

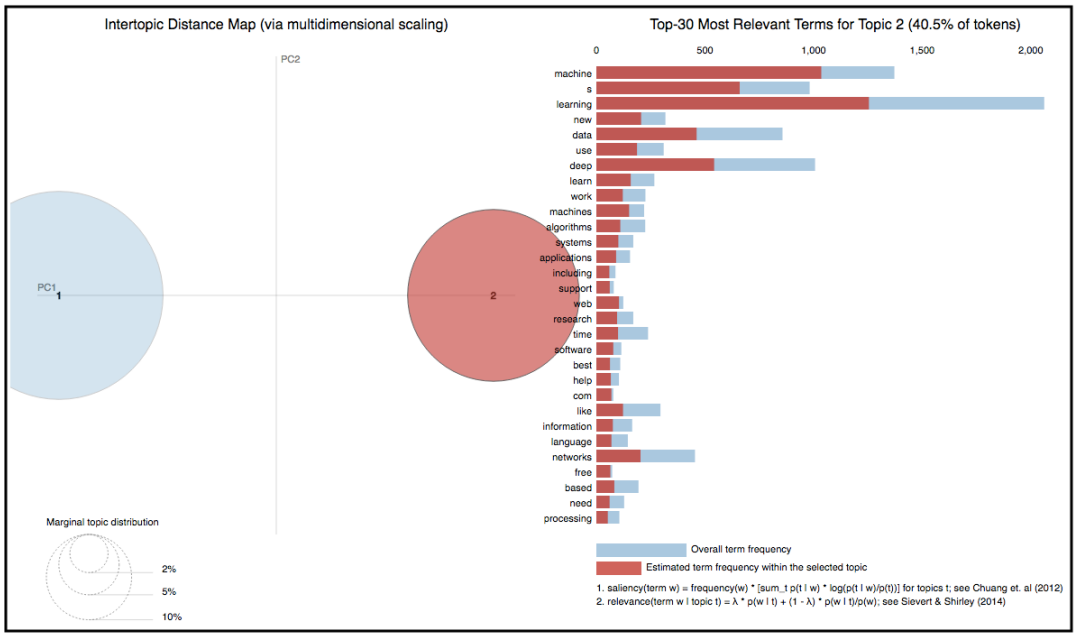


Figure 5: Topic Distribution Dashboard

4.5 Implementation

DDT is designed as a client-server model. The client is a web-based Javascript interface. This ensures that there is no client side setup as the analysts using the system could have a non-technical background.

The server is a Java and Python based cherrypy server that supports multiple clients simultaneously. The core features and functionality of DDT's domain search interface are shown in Figure 3. DDT is also packaged as a docker[9] container for easy deployment on the server.

5. USER EVALUATION

As an initial validation for our design decisions, we carried out a small-scale study. Since search engines are the most common tool used for gathering information on the Web, our study compares the effectiveness of DDT with that of Google for gathering information in a specific domain.

5.1 Experimental Setup

The evaluation involved six participants. The participants were graduate students or research associates with background in computer science. The two primary criteria for their selection was (1) that they should be very familiar with using search engines, especially Google and Bing, and (2) they should be capable of exploring information about a given topic on the Web. The users were given a demo of DDT and all its features, and they were allowed to use DDT to get familiar with it before the actual evaluation.

In order to keep the topics easy to understand and to ensure that the participants were not experts in the domain (as the goal here is for them to discover the topics), we selected topics from the *Ebola* domain in the TREC Dynamic Domain (DD) Track 2015 dataset[10]. The dataset for the *Ebola* domain consists of $\sim 143,044$ pages of which $\sim 5,832$ pages are labeled by humans into 20 topics.

Each user was then given the same 2 topics, in the same domain, and asked to find as many pages as they could for each of those topics using Google and DDT. While using Google the users annotated pages relevant to each topic by bookmarking them under corresponding folders. For DDT, the users annotated the pages for each topic with a custom tag corresponding to that topic. They were allowed 15 minutes for each topic on Google and DDT.

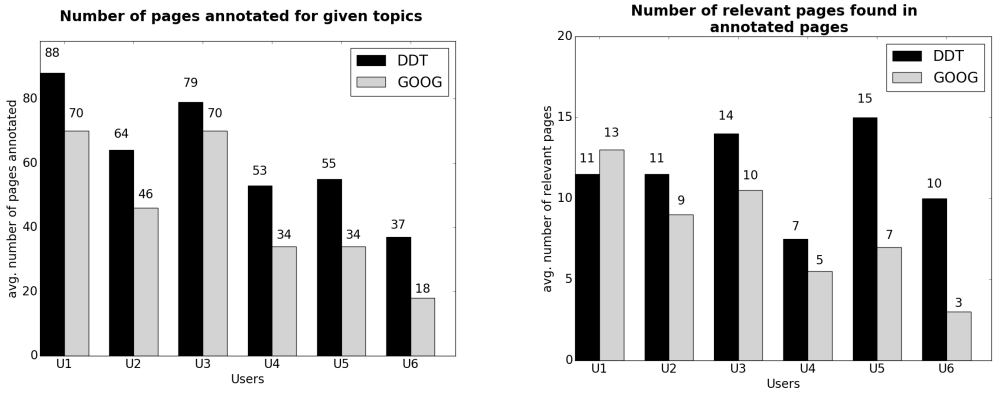
Since we used Google as a search engine in our experiments, we needed a "domain expert" that could consistently judge whether a page annotated by a user belonged to the given topic or not. Since we did not have access to such an expert directly, we instead built a multiclass SVM classifier[11], using the TREC DD data that was labeled by humans. The words (excluding stopwords) in the pages were used as features and the topic a page belonged to was the output class. The model was tested using cross validation which produced an average accuracy of 74.6%. Given the topic distribution, where the most frequent topic consisted of 700 pages, the model is still quite good, as a max baseline accuracy, if we labeled all samples with the most frequent topic label, would be $(700/5832) * 100 = 12\% << 74.6\%$.

5.2 Results

We measured the total number of pages that the users were able to annotate with Google and DDT. We executed the model on the annotated pages to find how many of them were actually relevant to the given topics. The results are shown in Figure 6.

Figure 6a plots the average number of pages annotated for the topics by each user. Users were able to annotate more pages using DDT than Google. Users reported that visualization and grouping of the pages by similarity made it easier for them to select and annotate a set of pages. Whereas on Google, they had to go through the list of results on multiple pages to be able to find the relevant pages and then bookmark them individually.

[9]https://www.docker.com/what-docker
[10]http://trec-dd.org/2015dataset.html
[11]We used LinearSVC from the scikit-learn library



(a) (b)

Figure 6: Evaluation: Google vs. DDT

Figure 6b shows the average number of relevant pages found by each user. The plot shows that the majority of the users were able to find more relevant pages with DDT than Google – in some cases 2-3 times more pages. This indicates that the features provided by DDT do help streamline domain discovery. The only exception was user $U1$. This user used the least number of features of DDT, which could explain the lower relevant pages found.

5.3 User Feedback

Users also completed a questionnaire about their experience with DDT. The following are the summarized positive and negative feedback we received. Given the duration of 15 minutes for each topic the users were not able to use all the features of DDT. The union of the set of features used by each user for this experiment were web search, MDS visualization window, backlinks and forward links, various filtering options and page tagging.

Positive.

- The users found the MDS visualization of the pages useful to see the similarity between the pages, analyze and annotate a group of pages
- The various methods to filter pages, such as by queries, tags and "more like this" (pages similar to a selected set of pages), facilitated finding and bringing in more pages related to the domain for analysis
- Ability to add user defined tags to annotate a set of pages allowed grouping them by topic
- Avoiding annotating the same pages multiple times as they are brought in through different queries
- Though none of the users was able to use the terms extracted due to the limited time of the test, the consensus was that the extracted terms were relevant to the domain and improved with page annotations

Negative.

- The feature for crawling forward and backwards from selected pages was difficult to use and led to a large number of irrelevant pages. This was especially true for the *Ebola* domain as most of the pages for this domain were news articles with links to different unrelated topics
- Although DDT was easy to use with little training, some aspects like the need for tagging extracted terms, the workflow (the sequence in which data gathering and analysis should be done) were not clear.

6. CONCLUSION AND FUTURE WORK

In this paper, we discussed the challenges in domain discovery on the Web and presented our first step towards building a solution to this problem. We proposed a new exploratory data analysis framework that combines techniques from information retrieval, data mining and interactive visualization to guide users in exploratory search. This framework was implemented and has been released as an open source system. We have also carried out a preliminary evaluation whose results are promising and indicate that the framework is effective.

The preliminary evaluation suggests that a framework like DDT can considerably improve the quality and speed of domain discovery. We have been able to achieve these results by using fairly simple mechanisms. In future work, we plan to explore more sophisticated interactive data mining techniques to leverage all the user feedback available to further improve the performance and accuracy of DDT, including interactive document clustering [11] and interactive topic modeling [10, 29].

An important feedback we received as part of the evaluation was the difficulty in using forward and backward crawling. This was because many documents, irrelevant to the domain of interest, were downloaded. We plan to use a classifier, created in an online fashion using the pages labeled by the user, to filter the downloaded pages and thereby considerably reduce the number of irrelevant documents that the analyst must analyze.

While our results are promising, we need to perform a comprehensive user study with a larger number of participants of diverse background. We would also like to conduct various evaluations of the effectiveness of DDT for non-Web text corpora.

Acknowledgments. We would like to thank the Continuum Analytics team for their help with packaging the DDT software for easy deployment. We would also like to thank Cesar Palomo who designed the original interface of the system, Jean-Daniel Fekete for his valuable inputs and Ritika Shandilya for her contributions to the DDT client interface. We also thank the Memex performers and collaborators for their feedback and suggestions to improve DDT. This work was funded by the Defense Advanced Research Projects Agency (DARPA) MEMEX program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

7. REFERENCES

[1] J. Alsakran, Y. Cen, Y. Zhao, Y. Jing, and D. Luo. STREAMIT: Dynamic visualization and interactive exploration of text streams. In *IEEE Pacific Visualization Symposium*, pages 131 – 138, 2011.

[2] L. Barbosa and J. Freire. An adaptive crawler for locating hidden-web entry points. In *Proceedings of WWW*, pages 441–450, 2007.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.

[4] S. Chakrabarti. Focused web crawling. In *Encyclopedia of Database Systems*, pages 1147–1155. Springer, 2009.

[5] A. Dayanik, D. D. Lewis, D. Madigan, V. Menkov, and A. Genkin. Constructing informative prior distributions from domain knowledge in text classification. In *Proceedings of ACM SIGIR*, pages 493–500, 2006.

[6] Z. Ghahramani and K. A. Heller. Bayesian sets. In *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[7] J. A. Hartigan and M. A. Wong. A K-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

[8] M. A. Hearst. *Search User Interfaces*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[9] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of ACM SIGIR*, pages 50–57, 1999.

[10] Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith. Interactive topic modeling. *Machine Learning Journal*, 95:423–469, 2014.

[11] Y. Huang and T. M. Mitchell. Text clustering with extended user feedback. In *Proceedings of ACM SIGIR*, pages 413–420, 2006.

[12] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 200–209, 1999.

[13] T. Mikolov, W. tau Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*, pages 746–751, 2013.

[14] G. Neumann and S. Schmeier. A mobile touchable application for online topic graph extraction and exploration of web content. In *Proceedings of the ACL System Demonstrations*, pages 20–25, 2011.

[15] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning Journal*, 39(2-3):103–134, May 2000.

[16] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of International Conference on Intelligence Analysis*, volume 5, pages 2–4, 2005.

[17] H. Raghavan, O. Madani, and R. Jones. Interactive feature selection. In *Proceedings of IJCAI*, pages 841–846, 2005.

[18] L. Rokach and O. Maimon. *Chapter 15, Clustering Methods in Data Mining and Knowledge Discovery Handbook*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[19] T. Ruotsalo, J. Peltonen, M. Eugster, D. Glowacka, K. Konyushkova, K. Athukorala, I. Kosunen, A. Reijonen, P. Myllymäki, G. Jacucci, and S. Kaski. Directing exploratory search with interactive intent modeling. In *Proceedings of ACM CIKM*, pages 1759–1764, 2013.

[20] D. M. Russell, M. Slaney, Y. Qu, and M. Houston. Being literate with large document collections: Observational studies and cost structure tradeoffs. In *Proceedings of IEEE HICSS*, volume 3, pages 55–, 2006.

[21] D. M. Russell, M. J. Stefik, P. Pirolli, and S. K. Card. The cost structure of sensemaking. In *Proceedings of the INTERACT and CHI*, pages 269–276, 1993.

[22] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, Aug. 1988.

[23] U. Scaiella, P. Ferragina, A. Marino, and M. Ciaramita. Topical clustering of search results. In *Proceedings of WSDM*, pages 223–232, 2012.

[24] C. Sievert and K. E. Shirley. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, 2014.

[25] S. Soliman, M. F. Saad El-Sayed, and Y. F. Hassan. Semantic clustering of search engine results. *The Scientific World Journal*, 2015.

[26] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

[27] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[28] W. Wright, D. Schroh, P. Proulx, A. Skaburskis, and B. Cort. The sandbox for analysis: concepts and methods. In *Proceedings of SIGCHI*, pages 801–810, 2006.

[29] Y. Yang, D. Downey, and J. Boyd-Graber. Efficient methods for incorporating knowledge into topic models. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 308–31, 2015.

[30] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma. Learning to cluster web search results. In *Proceedings of ACM SIGIR*, pages 210–217, 2004.

[31] A. J. Zhou, J. Luo, and H. Yang. DUMPLING: A Novel Dynamic Search Engine. In *Proceedings of ACM SIGIR*, pages 1049–1050, 2015.

Peekquence: Visual Analytics for Event Sequence Data

Bum Chul Kwon
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
bumchul.kwon@us.ibm.com

Janu Verma
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
jverma@us.ibm.com

Adam Perer
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
adam.perer@us.ibm.com

ABSTRACT

Exploring event sequences in big data is challenging. Though many mining algorithms have been developed to derive the most frequently occurring and the most meaningful sequential patterns, it is yet difficult to make sense of the results. To tackle the problem, we introduce a visual analytics approach, *Peekquence*. In this paper, we describe the design of *Peekquence*, which aims to increase the interpretability of machine learning-based sequence mining algorithms.

CCS Concepts

•**Human-centered computing** → **Visual analytics;**

Keywords

Event Sequence; Sequence Mining; Healthcare; Electronic Health Records

1. INTRODUCTION

Finding temporal patterns in longitudinal event sequences is a challenging task, as the volume and variety of events often make it difficult to extract salient patterns. In response to this challenge, data scientists have turned to machine learning, known as frequent sequence mining (FSM) techniques, to automatically detect the most common sequences of events to unearth interesting patterns. However, these algorithms often require users to specify a support threshold that, if too high, will yield only a few patterns, or if too low, will yield numerous patterns that may be difficult for data scientists to determine the interesting sequences from the mundane. In this work, we aim to make the results of frequent sequence mining algorithms more interpretable by giving end-users powerful ways to explore the data.

In particular, we propose several new techniques that include: 1) powerful ways to navigate the patterns by sorting with metrics relevant to users (variability, correlation to outcome, etc), 2) integration of patterns with patient time lines, so users can understand where the patterns occur in the actual data, and 3) overviews the summarize the most common events in the patterns.

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16) August 14th, 2016, San Francisco, CA, USA.

ACM ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

2. RELATED WORK

There are a large number of visual analytics tools designed to making temporal event sequences more interpretable. However, as a recently survey points out, many of them have difficulty handling the volume and variety of data [2].

Recently, there have been several approaches that integrate visualization with machine learning algorithms to surface the most interesting patterns, so only a manageable subset of patterns need to be visualized. Frequent Sequence Mining (FSM) is a popular data mining technique for finding sets of frequently occurring subsequences from a larger set of temporal event sequences. *Peekquence* uses SPAM (Sequential Pattern Mining) [1] as its FSM algorithm, which uses a bitmap-based representation for event sequences for efficiency reasons. Integrating visualization with the data mining algorithms is a promising approach, as it can help users understand algorithmic uncertainties, as well as trust the results of algorithms [9].

There have been other visualization systems that have integrated with FSM techniques. For instance, Frequence [5] integrates SPAM with visualization to support finding frequent patterns from longitudinal event sequences. This work was later extended and adapted to a medical context as Care Pathway Explorer [6]. However, the visualizations are similar to Sankey Diagrams [8], which have scalability issues when there are many patterns and large event dictionaries. Another system, TimeStitch [7], relies on the PrefixSpan [4] algorithm, which has several limitations, and is demonstrated on only small event sequences, generally composed of 2 or less events. *Peekquence* addresses these issues by having interactive sorting, clustering, and overviews to visualize thousands of patterns with large event dictionaries.

3. PEEKQUENCE

Peekquence is designed to make the results of the SPAM frequent sequence mining algorithm [1] more interpretable. To achieve the goal, the system has four views that present visual representations of the mining results. Figure 1 illustrates the four views: (A) the sequence network view, (B) the event co-occurrence histogram view, (C) the pattern list view, and (D) the patient timeline view. Using four coordinated views, users can interactively explore commonly occurring event sequences as well as their occurrences within patients' records.

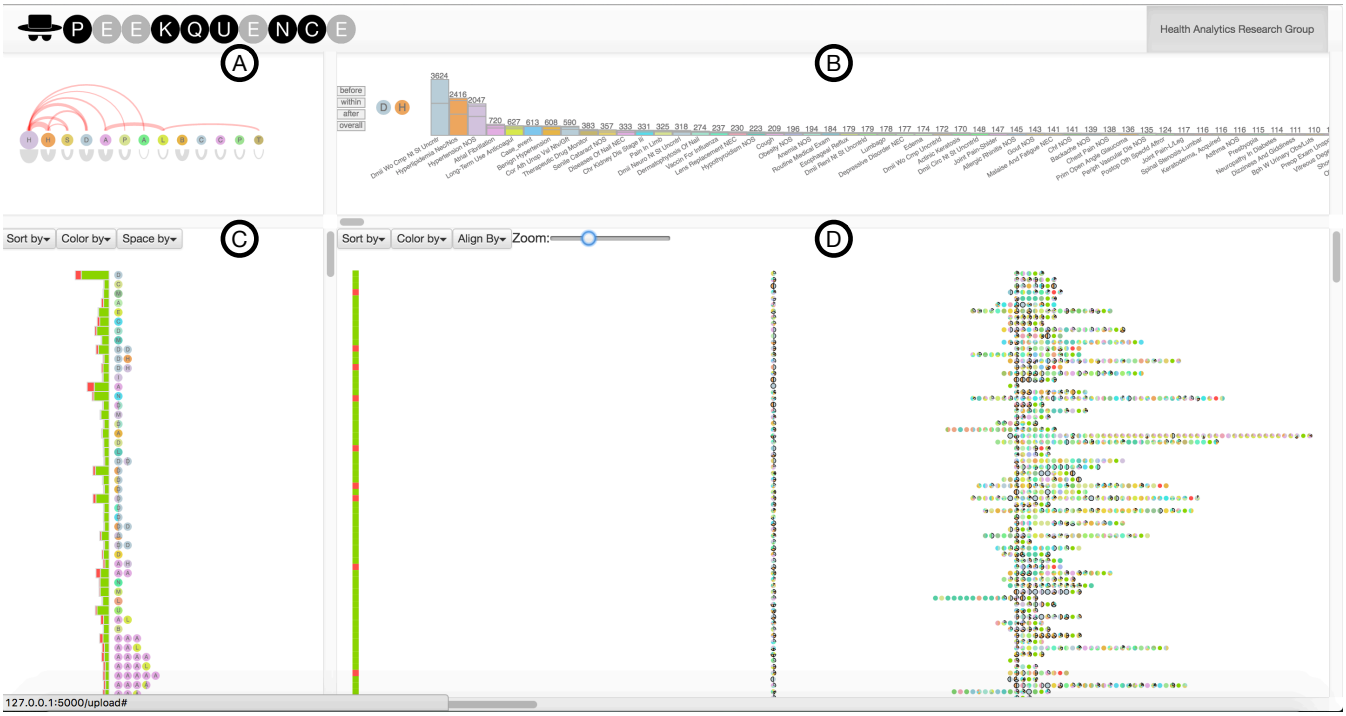


Figure 1: Peekquence consists of four views: (A) the sequence network view showing the frequency of event sequence occurrences within patterns mined from SPAM; (B) the event co-occurence histogram view showing the frequency of events co-occuring with a pattern selected ("S", "H" in this example); (C) the pattern list view showing patterns mined from SPAM with event sequences (colored circles with letters) as well as bars of patients with the ratio of case and control labels (diagnosis of a disease); (D) the patient timeline view showing patients' event sequences aligned with respect to the pattern selected ("S" and "H" events are vertically algined in this example).

Figure 2: The design of visual elements: circle for unit time duration, pie for event, color and letter for event type.

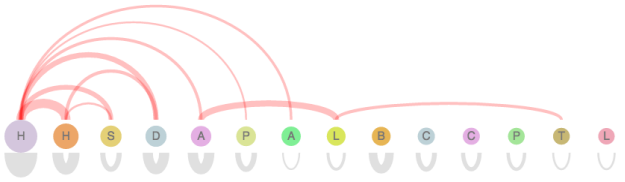


Figure 3: The sequence network view showing the most frequently occurring event types and their co-occurrences.

All four views use a common visual element to visualize event sequences: an *event glyphs* that visually encodes each unique event type that occurs in the mined data. The event glyphs are visually encoded as circles, colored according to an categorical ontology, and labeled with an abbreviation of the event type's name.

In the situation where multiple event types occur concurrently, the glyph is divided into multiple slices, similar to a pie chart, where each slices represents an event type. For example, Figure 2 shows a pattern consisting of three event types occurring sequentially: L, A, and L & A. In this medical dataset used throughout this paper, the colors represent the category of the clinical event according to ICD-9 (International Classification of Diseases) codes for classifying medical events.

The sequence network view in Figure 1 (A), also shown in Figure 3, acts as an overview, and shows the frequency of co-occurring event types within patterns mined from SPAM. The nodes indicate the types of events, and edges indicate that two nodes co-occur within patterns. The size of nodes and the thickness of edges show the number of patients that include events and event sequences within their records, respectively. For example, the purple "H" event, representing Hypertension events (a clinical event type indicating high blood pressure), has the largest size and the most edges to other events, showing that many event sequences in mined patterns contain the event. Users can click on a node or an edge to filter the pattern list view (Figure 1 (C)).

The pattern list view in Figure 1 (C) shows all patterns mined from SPAM, aligned vertically. Each row shows a pattern, visualized as a sequence of circular event glyphs that describe the sequence of the mined pattern. In addition, the pattern's association with outcome is represented by the stacked bar chart to the left of the sequence. In this medical example, the bars are divided into red and green, indicating the proportion of the case patients (patients diagnosed with the disease) and control patients (patients without the disease). This synchronization between pattern and outcome

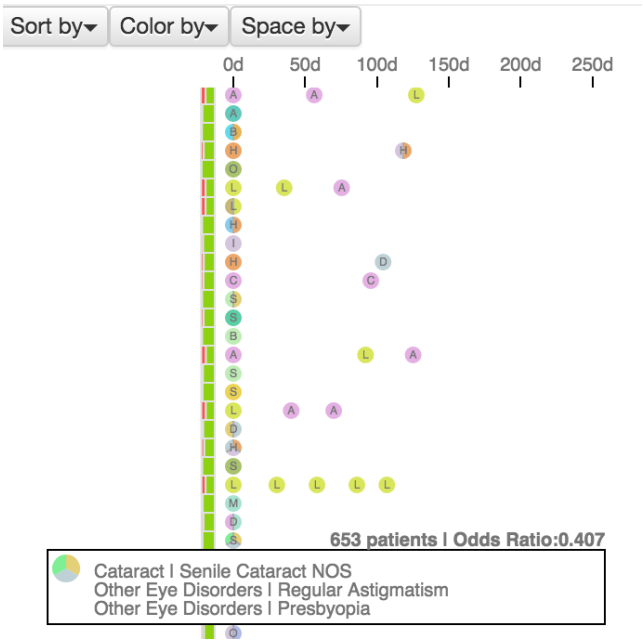
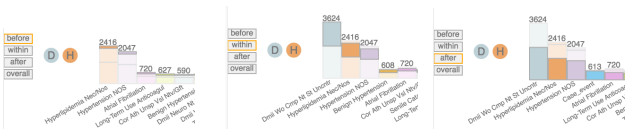


Figure 4: The pattern list view showing patterns of events spread out based on average time duration between events.

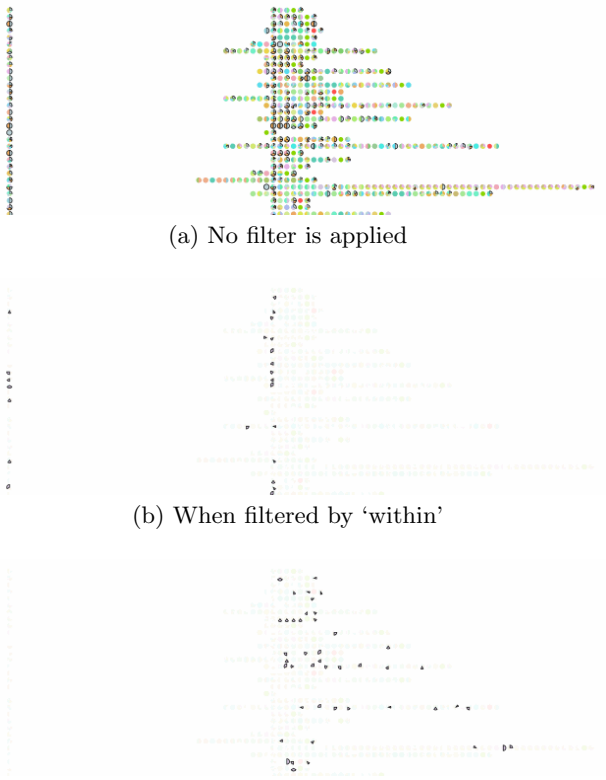
allows users to understand the impact of each pattern. This view is interactive, so users can sort the pattern list view by various attributes of the pattern: 1) the number of patients that have the pattern; 2) length of the pattern; 3) odds ratio of outcome; 4) variability of events in sequences; 5) clusters based on sequence similarity. Users can choose to horizontally spread event glyphs so that spaces between events indicate the average duration of occurrences of the events within patient records. Figure 4 shows a list of patterns, in which events are spread out to show average duration between the events. Users can click on a pattern to populate patient information in the event co-occurrence histogram view (Figure 1 (B)) and the patient timeline view (Figure 1 (D)).

The event co-occurrence histogram view in Figure 1 (B) shows the summary of patient records which contain the selected pattern from the pattern list view. The summary is the histogram of events co-occurring with the selected pattern within patients. Each bar indicating a event type is divided into three blocks that show events occurring 1) before, 2) within, and 3) after the selected pattern, respectively. For example, Figure 1 (B) shows the histogram of the selected pattern of "D" and "H" events. The top block of each bar in indicates the number of occurrences of the corresponding event before the "D" event within patient records. Subsequently, the second block shows the number of co-occurring events on the same day of or later than the "D" event and before or on the same day of the "H" event. Lastly, the bottom



(a) Before (b) Within (c) After

Figure 5: The histogram view sorted by before, within, and after the pattern.



(a) No filter is applied

(b) When filtered by 'within'

(c) When filtered by 'after'

Figure 6: The patient timeline view 6a before filter, 6b filtered by within pattern, and 6c filtered by after pattern.

block indicates the number of co-occurring events later than the "H" event. Using the view, users can find the most commonly co-occurring events with the selected pattern. The view allows users to sort the histogram horizontally by the frequency of events before, within, and after the pattern as shown in Figure 5. In this view, users can select a block of histogram bar to highlight the events within patient records shown in the the patient timeline view (Figure 1 (D)).

The patient timeline view shows individual patient's entire event sequences per row in Figure 1 (D). The sequences are aligned horizontally so that the selected pattern occurs at the same horizontal location. To do so, we shift patients' records horizontally, which sometimes creates empty space between events. Thus, in Figure 1 (D), the horizontal distance between events of "D" and "H" indicates the maximum days of events that occurred between the "D" and "H" events within a patient's record. As mentioned earlier, by clicking a block of the event co-occurrence histogram view, users can filter the patient timeline view. Figure 6 shows the patient timeline view filtered by the event "H", shown as purple pies, 6b within and 6c after the selected pattern of "D" and "H".

In Peekquence, the four views independently show information about patterns mined from an algorithm, and they also connect to each other by highlighting and filtering other views. The divided views ensure participants to gain new insights in different levels. At the same time, the interactive exploration enables users to progressively investigate event sequences from overview (top views) to details (bottom views) and vice versa. The sections also let users smoothly switch back and forth between pattern-level investigation (left views) and patient-level investigation (right

views). The design of Peekquence captures information in different granularities providing users with appropriate interpretation layers, which confirm the importance of paving the cow path of users' analysis pattern [3]. To increase the transparency of complex pattern mining algorithms like SPAM, we believe that it is important to provide users with visual channels to different modalities and depths of information through divided-but-connected views.

The current status of Peekquence shows the potential of visual analytics approach to make frequent sequence mining algorithms more interpretable. At the same time, we believe that much work remains to be done to improve the prototype. First, we need to allow users to run the SPAM algorithm with a subset of data as well as user-specified parameters. By doing so, users will have an ability to detect user-defined patterns. Second, we are investigating new methods for visually summarizing event sequences that share common events within them. As the number of patterns grow, it is difficult for users to explore patterns. Thus, visual aggregation will help users understand the difference and similarity between event sequences. Lastly, we are investigating ways to incorporate predictive models so that the model can provide the probability of having certain diseases based on event sequences of users.

4. CONCLUSION

In this paper, we presented our visual analytics approach, called Peekquence, which aims to increase the interpretability of frequent sequence mining algorithm such as SPAM. The four views combined with interactions provide useful functionalities for users to make sense of patterns as well as their occurrences within patients' records. In future work, we aim to integrate the visual representation with the algorithm so that users can iteratively run the algorithm with new parameters based on insights gained from previous runs. Work is also in progress to exploit the hierarchy of events and provide users the ability to run SPAM at different levels of detail.

5. ACKNOWLEDGMENTS

We would like to thank our colleagues who provided constructive feedback for the research.

6. REFERENCES

[1] J. Ayres, J. E. Gehrke, T. Yiu, and J. Flannick. Sequential pattern mining using bitmaps. pages 429–435, 2002.

[2] F. Du, B. Shneiderman, C. Plaisant, S. Malik, and A. Perer. Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus. *IEEE Transactions on Visualization and Computer Graphics*, In press.

[3] B. C. Kwon, S.-H. Kim, S. Lee, J. Choo, J. Huh, and J. S. Yi. Visohc: Designing visual analytics for online health communities. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of the Visual Analytics Science and Technology)*, 22(1):71–80, 2016.

[4] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1424–1440, 2004.

[5] A. Perer and F. Wang. Frequence: Interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, pages 153–162, New York, NY, USA, 2014. ACM.

[6] A. Perer, F. Wang, and J. Hu. Mining and exploring care pathways from electronic medical records with visual analytics. *Journal of Biomedical Informatics*, 56(C):369–378, Aug. 2015.

[7] P. J. Polack Jr, S.-T. Chen, M. Kahng, M. Sharmin, and D. H. Chau. Timestitch: Interactive multi-focus cohort discovery and comparison. In *IEEE Proceedings of the Visual Analytics Science and Technology (VAST)*, pages 209–210. IEEE, 2015.

[8] P. Riehmann, M. Hanfler, and B. Froehlich. Interactive sankey diagrams. In *IEEE InfoVis*, pages 233–240, 2005.

[9] D. Sacha, H. Senaratne, B. C. Kwon, G. Ellis, and D. A. Keim. The Role of Uncertainty, Awareness, and Trust in Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of the Visual Analytics Science and Technology)*, 22(01):240–249, Jan. 2016.

Human-guided Flood Mapping on Satellite Images

Jiongqian Liang
Department of Computer Science and Engineering
The Ohio State University
liangji@cse.ohio-state.edu

Peter Jacobs
Data Analytics
The Ohio State University
jacobs.269@osu.edu

Srinivasan Parthasarathy
Department of Computer Science and Engineering
The Ohio State University
srini@cse.ohio-state.edu

ABSTRACT

Flooding is responsible for substantial loss of life and economy. Flood mapping, the process of distinguishing flooded areas from non-flooded areas during and after a disaster, can be very useful in guiding first response resources in a disaster situation, and in assessing flood risk in future disaster scenarios. This paper involves the use of image segmentation methods and human guidance to provide a mechanism for flood mapping. Previous image segmentation methods do not work well in flood mapping because they are designed to segment objects out of an image, where there are only a few objects, e.g., foreground-background segmentation. However, satellite images of flooded areas often contain hundreds to thousands of large and small water areas that need to be identified. Therefore, we design a semi-supervised learning algorithm specifically to tackle the flood mapping problem. We first divide the satellite image into patches using a graph-based approach depending on the proximity and intensity of pixels. We then classify each of the patches in an interactive and incremental way, where each time the user is asked to label a few patches and we learn a classifier to automatically classify other patches into water area or land area. We run our algorithm on satellite images of Chennai, India during the 2015 Chennai flood period. The results show that our algorithm can robustly and correctly detect water areas compared to baseline methods. We compare the segmentation results of post-flood with pre-flood and conduct an effective flood evolution analysis.

Keywords

Flood mapping; Graph-based approach; Semi-supervised; Image Segmentation

1. INTRODUCTION

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16) August 14th, 2016, San Francisco, CA, USA.

According to Hallegatte et al. [11], if worldwide flood probabilities remain constant over the next 35 years, rising sea levels, sinking land areas, and growing urban coastal populations are expected to drive annual global flood losses from 6 billion U.S. Dollars per year to upwards of 60 Billion US Dollars per year by 2050.

It is of paramount importance to identify ways to reduce the probability of flooding in coastal urban areas. A first step in achieving this goal is the reliable identification of regions in coastal cities that are most susceptible to flooding. If these regions can be identified, action can be taken to better protect these areas from flooding, and public policy can prevent development in areas that have a high risk of flood damage. Flood mapping allows for identification of areas of high, medium, and low risk of flooding, which can help prevent serious flooding from happening. Another application of flood mapping is the quick identification of areas that have been severely flooded during or immediately after a storm. This information can be utilized to guide first responders to where they are most needed.

In order to conduct flood mapping, satellite images promise tremendous potential in monitoring flood disasters due to their low cost and consistent and repetitive data acquisition capability over large spatial areas [17, 20]. Compared to the sparse *in situ* physical sensing data (e.g., river gauge data and weather station records), satellite images offer a synoptic view of the landscape and provide a comprehensive geo-spatial perspective on flood events. The problem here is how to correctly identify areas flooded areas given high-resolution satellite images.

This problem can be regarded as an image segmentation task, where one wants to segment flooded areas out of the whole region. While image segmentation has been widely studied in the image processing community [19, 7, 5, 1], these approaches cannot directly be applied in flood mapping. On one hand, they mostly focus on background and foreground segmentation and the total number of segments is relatively small. On the other hand, these approaches are usually not scalable on large datasets and cannot work on high-resolution satellite images. Moreover, the difference between flooded regions and other regions can be so subtle that human guidance is required in order to correctly locate floods. To address these difficulties, we propose a semi-

supervised learning method that can interact with humans and incrementally conduct flood mapping efficiently.

In this paper, we explore novel ideas to integrate network-analysis and human guidance for flood mapping, where we use network clustering approaches to divide images into patches and adopt human guidance to interactively label the patches as water and land areas. The method for flood mapping involves segmentation of satellite images of the given city before and after a flood occurred to identify land and water areas. This is followed by a comparison of these pre and post disaster segmentations to identify flooded vs. non-flooded areas. The experiments on satellite images of Chennai, India during the 2015 floods show that our method can more effectively identify flooding areas compared to state-of-the-art approaches. Our method is also much more efficient, which enables real-time incremental learning and provides instant information to help prioritize post disaster repair and relief activities.

The rest of the paper is organized in the following way. Section 2 reviews related literature. Section 3 presents our methodology for flood mapping. Section 4 describes an extensive experiment conducted to show the efficiency and efficacy of our method. We describe ongoing and future work in Section 5. Finally, we provide a summary in the last section.

2. RELATED WORK

The problem of flood mapping satellite images is related to the field of image segmentation while the incorporation of human guidance is connected to semi-supervised clustering. Therefore, we review some existing work on image segmentation and semi-supervised clustering in this section. We also discuss past work from the flood mapping domain.

Image segmentation is a long-standing problem and a wide range of techniques has been developed to attempt to segment an image [19, 7, 1, 5, 3, 2]. Some classic methods for image segmentation involve thresholding. Thresholding-based techniques for grayscale image segmentation pick a pixel intensity T and force all pixels with intensity above T to be one color, while all pixels with intensity below T become another color [2]. Thresholding produces a binary image, and if the threshold T is selected carefully, this binary image can isolate foreground objects from the image background, which can be an effective mechanism for image segmentation [2]. Picking the value of T is the main challenge in thresholding. Many methods have been developed for selecting the threshold pixel intensity T [1, 15, 18, 21, 13]. One common method for picking T, known as Otsu thresholding [1], involves finding the pixel intensity that creates the greatest separation and least overlap between the modes in the pixel intensity histogram; this method works best for images with bi-modal pixel intensity distributions. However, Otsu thresholding is not robust when applied to images with noise because the segmentation produced is merely based on the intensity of each pixel without looking at the pixels nearby. If applied to satellite images, it will generate many tiny spots that do not represent relevant higher level structure in the image.

Other methods of image segmentation include the region merging technique proposed by Baatz et al. [3]. They treat pixels as objects initially, and at each iteration, the two objects are merged that lead to the smallest increase in heterogeneity. More recently, graph-based methods have been introduced for image segmentation. Graph-based techniques formulate the image as a graph, and then use some form of community detection to find a segmentation. Shi et al. [19] create a graph with weighted edges and use the normalized cut criterion to segment the image. Browet et al. [7] also formulate the image as a graph; they use modularity as a criterion to find a segmentation for the image. However, these methods are computationally expensive and are not scalable on large satellite images.

Furthermore, there are some semi-supervised learning approaches for image segmentation [5, 14, 4]. One influential semi-supervised method for image segmentation is the watershed algorithm, developed by Beucher and Meyer [5]. The watershed algorithm allows the user to mark different segments in the image. The algorithm then performs a region growing technique, starting from the user placed marks, that operates on the gradient of the original image. While it allows interaction with users and can conduct image segmentation incrementally, the watershed algorithm requires the user to place at least one marker for each segment, which is inefficient in the scenario of flood mapping on satellite images.

Beyond image segmentation, our problem is also relevant to semi-supervised clustering [10]. Semi-supervised clustering involves the addition of "must-link" and "cannot-link" information into the clustering process. "Must-link" information indicates that two objects "must" be in the same cluster. "Cannot-link" information indicates two objects "cannot" be in the same cluster. Wagstaff et al. [22] show that insertion of "must-link" and "cannot-link" information into the clustering process can lead to improved accuracy and efficiency in clustering. However, our problem on satellite images is quite different from the traditional setting of semi-supervised clustering and we need a more convenient way than "must-link"/"cannot-link" for human to provide supervision.

Flood mapping itself has been the subject of previous work. Wang et al. [23] use Thematic Mapping, a type of earth observing sensor, to identify land and water areas before and after flooding, followed by the use of a classification algorithm to identify flooded and non-flooded areas. Henry et al. [12] use Advanced Synthetic Aperture Radar (ASAR) data for flood mapping. These methods both rely on data sources from earth-observing satellites (landsat 7 and Envisat respectively). These data sources are not always available at the time of a disaster. For example, Envisat, the satellite that provided the ASAR data used in the paper by Henry et al., is no longer in operation. Moreover, these methods do not support interactions with ordinary users and cannot update the results incrementally.

3. METHODOLOGY

To effectively solve our problem and overcome limitations in prior work, we state the following desiderata:

- **Fast flood mapping:** Conduct efficient/scalable flood mapping for large satellite images. Efficiency is necessary to facilitate interactive learning and it is also vital if the method is used to help guide emergency first responders in a flood disaster.

- **Guided by human:** Incorporate guidance from humans to achieve better results.

- **Easy to use:** Ordinary users can easily use the method and conveniently provide supervision.
- **High quality results:** Generate effective flood mappings that can be easily interpreted.

Building on these desiderata, we propose a novel method for flood mapping. Our method first preprocesses the satellite images and then detects water areas from satellite images in an interactive fashion using human guidance. Then by comparing the water areas pre and post disaster, it can identify the flood areas. We describe the method in detail below.

3.1 Preprocessing

To label areas of a satellite image as either land or water, we need to decide on a primary unit for labeling. A straightforward way is to treat each pixel as a unit and conduct pixel-based labeling. The drawback is that we lose the information derived from geographic correlations, and the results will tend to be noisy. The labeling results will involve fuzzy and blurring boundaries, and there might be many small spots. Also, it is difficult to label one pixel manually. Another alternative is to conduct uniform grouping, which involves treating the image as a grid with squares of uniform size. However, without using the intensity information of the image, this grouping can go across boundaries (many patches include both water and land), which is not desirable. In this paper, we adopt a graph-based approach for patch generation, which is efficient and can not only effectively detect regions of different sizes, but can also avoid generating regions across land-water boundaries.

3.1.1 Graph Construction

Graph-based segmentation has been widely studied in the literature [19, 9, 6, 7]. In this paper, we convert the image into an undirected graph following the approach proposed by Cour et al. [8]. Each pixel of the image is treated as one node and each pixel has edges to nearby pixels within the distance of d_{max}, where d_{max} is a user-defined parameter. The weight of the edge between pixel i and pixel j is defined using the following approach.

$$w_{ij} = \begin{cases} e^{-\frac{d(i,j)^2}{\sigma_x^2} - \frac{|F(i)-F(j)|^2}{\sigma_i^2}} & \text{if } d(i,j) < d_{max} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

where $d(i,j)$ is the Euclidean distance between pixels i and j and $F(i)$ is a feature vector evaluated at pixel i. The feature vector can be the scalar intensity value or the RGB values of an image. σ_x, σ_i and d_{max} are parameters controlled by the user.

Note that the number of nodes in the constructed graph n is equal to the number of pixels in the image, and the number of edges is $m = k * n$, where k is a small constant factor depending on the setting of d_{max}.

3.1.2 Graph Clustering to Generate Patches

After we construct the graph from the image, we cluster the graph. Since the satellite image usually contains hundreds of millions of pixels, we need a highly scalable graph clustering algorithm. In this paper, we leverage the off-the-shelf tool Multi-level Regularized Markov Clustering (MLR-MCL) [16], which is an efficient multi-level graph clustering software.

Since the goal of graph clustering is to generate basic units for labeling, we tend to produce a large number of clusters. Empirically, we find the method works well when the average size of clusters is a few hundred pixels. Once we have the graph clustering results, pixels in the same cluster are considered to be in the same patch.

There are many advantages to producing patches using this graph-based approach. This approach can avoid the edges/boundaries of a segment being in a patch (e.g. land and water areas being in the same patch). Also, it becomes very easy to control the number of patches using MLR-MCL. Moreover, MLR-MCL has time complexity linear to the number of edges and is very efficient to run in our graph (the number of edges is proportional to the number of nodes).

3.2 Human-guided Labeling

After generating patches, the next step is to ask the user for a couple of labels. The user will place a few markers in the image to label a few patches that they identify as land/water. To utilize this user-provided supervision, a binary classifier is learnt and then applied to the rest of the unlabeled patches.

3.2.1 Learning the Binary Classifier

In this paper, we use k-NN as the classifier since there are only a few features and they are interpretable. In particular, we define the distance function between two patches i and j as follows.

$$D(i,j) = \left\| \bar{F}(i) - \bar{F}(j) \right\|_2 * \log\left(dist(i,j)\right) \quad (2)$$

Eq. 2 contains two factors. The first factor compares the features of the two patches while the second factor calculates the Euclidean distance between the two patches. Specifically, to calculate the first factor, we average the feature vectors of the two patches respectively and compute the L2-norm of their difference. For the second factor, we compute the centroids of both patches and compute the Euclidean distance between the centroids. To decrease the effect of geographical distance, we take the logarithm of the Euclidean distance.

To classify an unlabeled patch, we find the k most similar labeled patches based on the distance function in Eq. 2. The classification of the patch is then decided by a vote conducted using the labels of these k most similar labeled patches.

3.2.2 Interactive Labeling And Incremental Update

Instead of asking the user to label the patches at one time, we create an interactive environment for labeling. The user is asked to label one patch at one time and our method generates classification results based on the labels currently available. The results are presented to the user in real-time and the user decides whether to label more patches or not. If/when the user provides a new label, the method will incrementally update the results. Our algorithm terminates only when the user does not plan to label more patches; at this point, the result is saved. In practice, we find out that the user usually only needs to mark 2 to 6 patches to generate reasonably good results.

Image Date	Size of Image	σ_x^2	σ_i^2	d_{max}	# patches
11/24/2015	800x444	3	16	2	12946
10/19/2015	4500x2500	2	16	2	69674
10/31/2015	4500x2500	2	16	2	69674
11/12/2015	4500x2500	2	16	2	69674
11/24/2015	4500x2500	2	16	2	69674
12/06/2015	4500x2500	2	16	2	69674
12/18/2015	4500x2500	2	16	2	69674

Table 1: Parameter settings used to construct the graph and generate patches

3.3 Flood Mapping

After obtaining segmentations of an urban area before and after a flood, flood mapping can be performed through comparison of the segmentations. We treat the satellite images before the flood as a baseline and compare satellite images during and after the flood with this baseline. Areas that are not segmented as water before the flood, but are segmented as water after the flood are considered flooded areas.

4. EXPERIMENTS AND ANALYSIS

We run our algorithm on real-world satellite images and conduct analysis in this section.

4.1 Dataset and Baseline

We use satellite images of Chennai, India during the 2015 South Indian Floods [1]. In total, we collect six satellite images during the flood, one for every twelve days, shown in Figure 1.

We compare our algorithm with some state-of-the-art algorithms for image segmentation: 1) Watershed algorithm [5]; 2) Normalized cut algorithm [19]; 3) Graph-based image segmentation with post-processing. The method for generating patches in the 3rd baseline is the same method used to generate patches in our method. However, the second step of the 3rd baseline method is purely unsupervised; the step involves continued merging of nearby patches based on the similarity of pairs of patches until the designated number of patches has been generated.

Considering the fact that some baselines (e.g., Normalized cut algorithm and Watershed algorithm) are very computationally expensive and cannot finish on the large satellite images in a reasonable amount of time, we divide our experiment into two parts. In the first part, we downscale the satellite images and run our method and baselines on them for comparison. As an example, we run all the algorithms on the satellite image of Chennai on 11/24/2015, which is re-sized from 4,500×2,500 to 800×444. We then run our method on the full-size satellite images and conduct further performance analysis.

For the experiments, we implement our algorithm using `Python`. We use the OpenCV API for the Watershed algorithm. For the Normalized cut, we obtain the source code from authors [2]. We also implement the graph-based method with post-processing. Basic information about the datasets and parameter settings for our algorithm are displayed in Table 1.

4.2 Comparing Different Methods

[1]https://en.wikipedia.org/wiki/2015_South_Indian_floods
[2]https://www.cis.upenn.edu/~jshi/software/

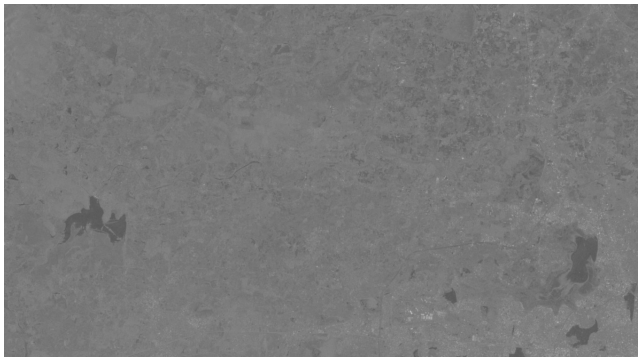
Method	# Markers	Total Time (s)	Interactive Labeling Time (s)
Our Approach	2	30.551	0.057
Watershed Algorithm	11	0.225	0.225
N-cuts Algorithm	0	538.615	0.000
Graph method w. post-process	0	558.220	0.000

Table 2: Running time comparisons of different methods. # markers is the number of markers the human provides for the algorithm.

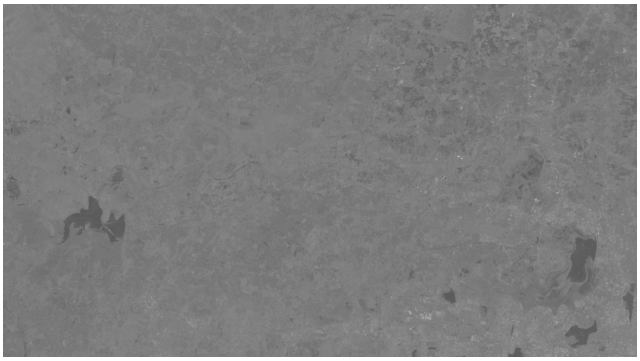
We now compare the performance of our algorithm with the baselines on the downscaled image as shown in Figure 2 (Chennai area on 11/24/2015). The results of segmenting water areas are shown in Figure 3 while the execution time is listed in Table 2. We hereby highlight the following observations:

- Among all the approaches, our method is apparently the best. Our method can clearly identify most of the water areas and even long thin rivers while all other methods fail to do this. Particularly, our method is good at identifying regions of arbitrary shape and it does not limit the size of each segment. We notice that Otsu thresholding [1] might also have similar advantages, but it tends to generate more tiny partitions because its segmentation results only depend on the intensity of each pixel and one pixel can be an individual partition if its pixel intensity is far different from its neighboring pixels [3].
- Compared to the Watershed algorithm, our method produces better results while requiring less effort from humans. The Watershed algorithm seems to correctly capture some boundaries but could not segment out small water areas, including the long thin rivers. In the example shown in Figure 3, we place nine markers in different water areas and two markers in the land areas (see Figure 4). But the segmentation results are still not desirable. On the other hand, using our method, we only need to place one marker in water and one in land respectively (see Figure 4) and the results are much better than the Watershed algorithm. One of the reasons for this difference is that labeling of the Watershed Algorithm grows from the user marked regions in a local fashion and therefore requires much more manual labels for it to work reasonably well.
- The Normalized Cut algorithm tends to generate over-balanced segments and cannot extract segments of long thin shape (shown in Figure 3(c)). Though it performs well in detecting most of the boundaries, it breaks large areas into pieces that should be in one partition. This can be seen from the split of some large lakes. As a whole, the results are much worse than our algorithm.
- Graph-based segmentation with post-processing in general works well in detecting some large areas (see Figure 3(d)). However, similar to the Normalized Cut algorithm, it cannot detect small water regions, especially those long thin rivers.

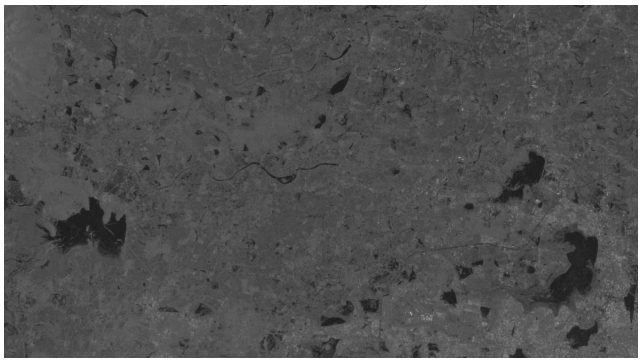
[3]Further investigation shows that Otsu generates two times as many segments as our method on the image.



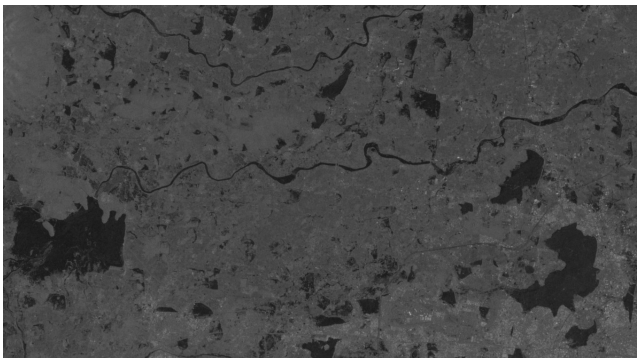
(a) 10/19



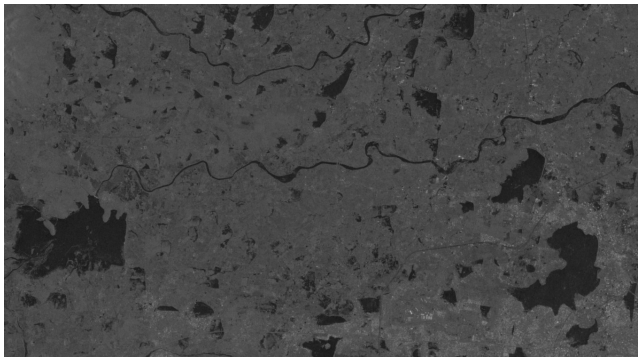
(b) 10/31



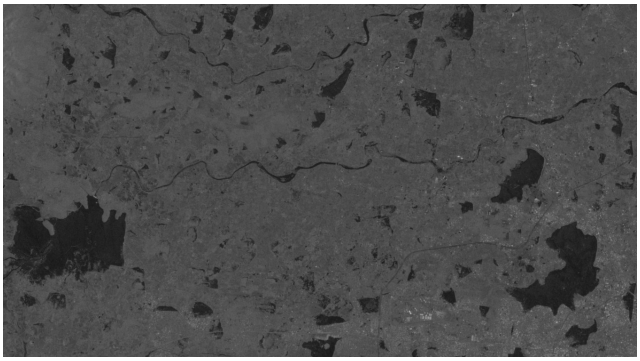
(c) 11/12



(d) 11/24



(e) 12/06



(f) 12/18

Figure 1: Satellite images of Chennai from 10/19/2015 to 12/18/2015. One image for every 12 days.

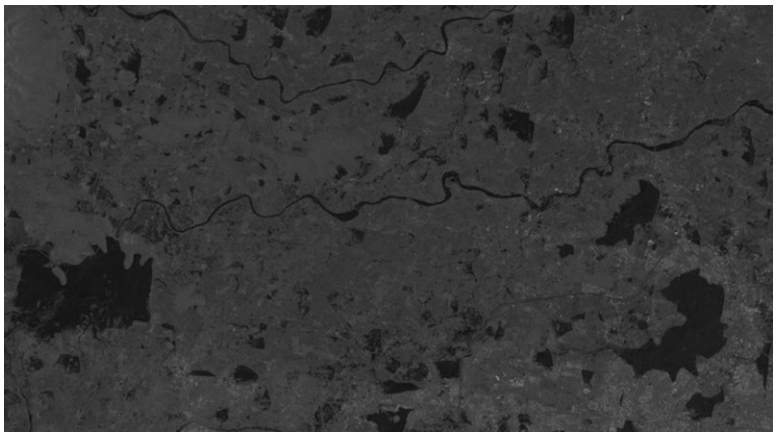
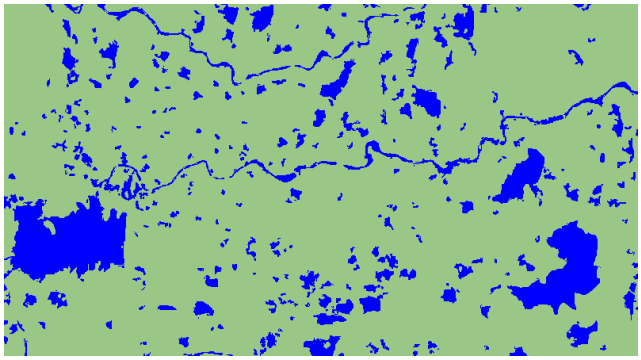
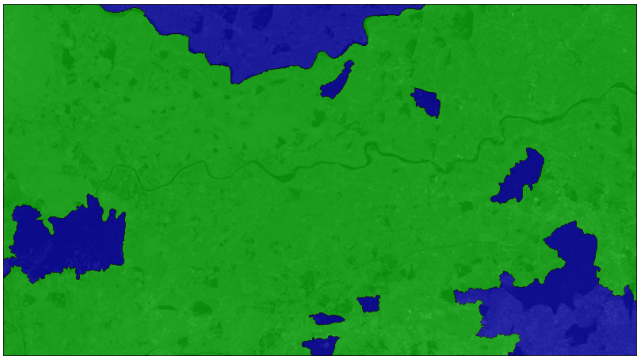


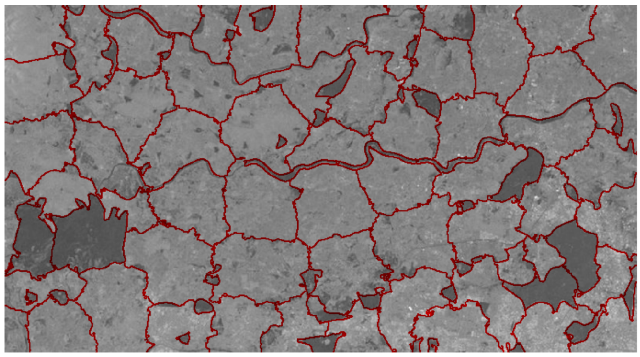
Figure 2: Down-scaled satellite image of Chennai area on 11/24/2015.



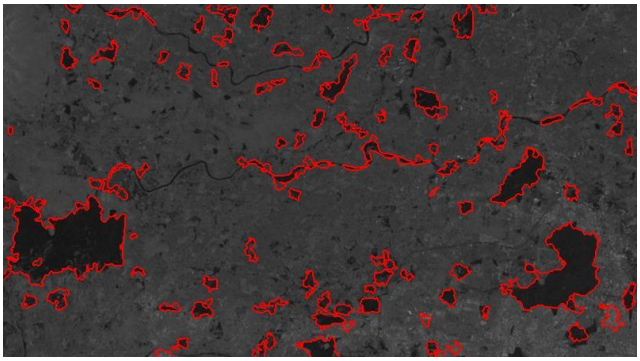
(a) Our Method



(b) Watershed Algorithm

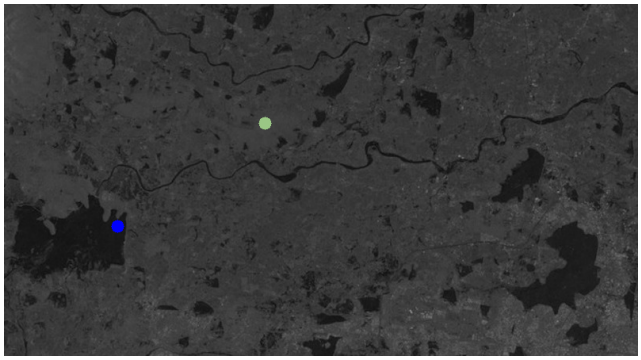


(c) Normalized Cuts Algorithm (100 partitions)

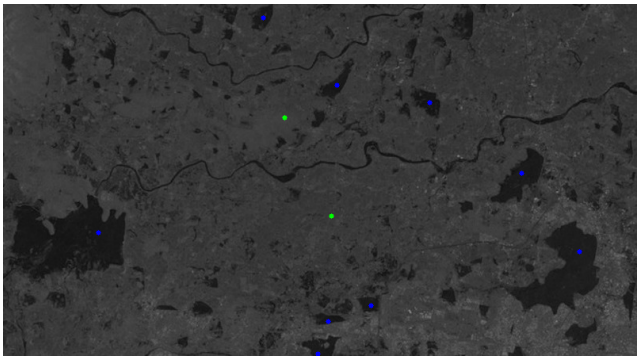


(d) Graph-based Clustering with Post-processing (100 partitions)

Figure 3: Image segmentation results of different approaches on satellite images of 11/24/2015 (down-scaled). (a) is the result of our algorithm, where blue color indicates water areas while green represents land areas. (b) is the result of Watershed algorithm, where blue color indicates water areas and green represents land areas. (c) is the result of Normalized Cut method, where red line marks out the boundaries between land and water areas. (d) is the result of graph-based method with post-processing, where red line also highlights the boundaries between land and water areas.



(a) Two markers provided by the user to our method



(b) Eleven markers provided to Watershed algorithm

Figure 4: Labels that the user provided for the algorithm to learn labeling. Blue points label the areas as water while green points label them as land.

- As shown in Table 2, our method is very fast during interactive labeling and with regard to overall time, it outperforms 2 of the other 3 algorithms. The Normalized Cut algorithm is very slow because it uses spectral clustering and requires computation of the eigenvectors of the Laplacian matrix, which is very computationally expensive. The Graph-based segmentation with post-processing method requires a great deal of computation at the stage of hierarchical merging and is also much slower. Even considering the time of preprocessing for patch generation (30.494 seconds), our method is still much more efficient than the two just mentioned algorithms. Due to the length of patch generation preprocessing, the Watershed algorithm is faster than our method, but our method requires less time during the stage of interactive labeling, which is an important convenience for human users. For example, for one image, we only need to conduct preprocessing once; then, as a result of the efficiency in interactive labeling, the preprocessed image can be interactively labeled by many human users many times easily. This process allows users to find what they consider the 'best' segmentation through trial and error, without long wait times.

4.3 Segmenting out Water Areas on Original Satellite Image

We have shown the advantages of our method compared to other baselines above. Now, we further show the results of our method on all the full-size satellite images in Figure 1. Basic information about the datasets and parameter settings for our algorithm are displayed in Table 1. The segmentation results are shown in Figure 5. From Figure 5, we can observe that our algorithm consistently generates high quality segmentations and is capable of correctly detecting the arbitrary boundaries between land and water. Most long thin rivers, small irregular water bodies, and large wide lakes are correctly extracted.

4.4 Dynamic Analysis for Flood Mapping

While we mainly focus on image segmentation as a method for distinguishing water from land in satellite images above, we now discuss how we adopt the image segmentation method developed to detect flood areas. To this end, we refer to the historical satellite image from before the flood and conduct dynamic analysis.

By simply looking at Figure 5, which shows images of Chennai from 10/19 to 12/18, we can see the water areas greatly increase between 10/31 and 11/12. The water areas start decreasing from 12/06 onward.

To create a flood map, we use the segmentation result from 10/31/2015 as the baseline and compare this segmentation to the segmentations from later dates. Figure 6 presents the dynamic changes of water areas. Red color indicates the areas that change from land into water while yellow color indicates the opposite change. From Figure 6, we can clearly observe that 11/24 and 12/06 have the largest number of water areas; water areas seem to decrease following 12/06. Red areas are likely regions affected by the flood. The flood maps are quite consistent with the fact that the South Indian floods lasted from 11/08/2015 to 12/14/2015.

In addition, we generate two animated gifs and put them on our website [4]. The changes of water areas can be more clearly seen on the animated gifs, revealing the flood surges and recessions.

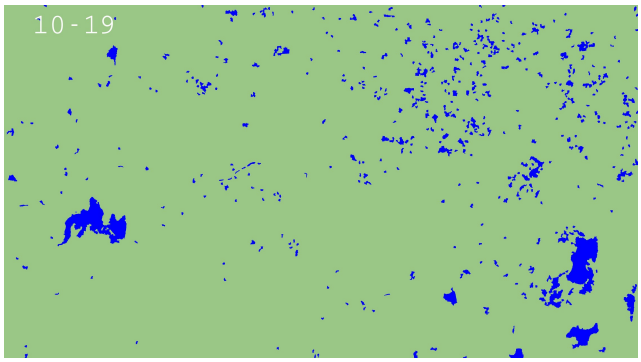
5. ONGOING AND FUTURE WORK

In this section, we discuss some of the directions that we are working on.

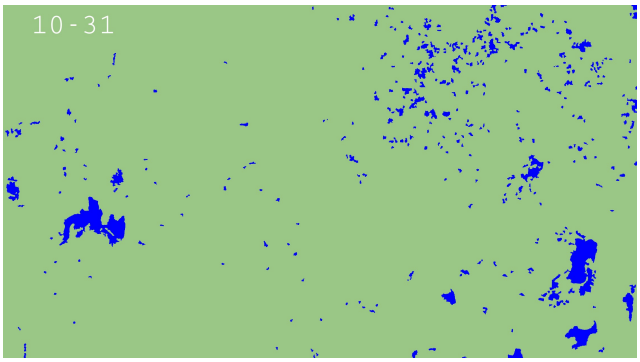
- **Improve the learning models.** While we use very simple k-NN method for classification in this paper, we would like to adopt more advanced classifiers to label patches, such as SVM and neural networks. Meanwhile, more features for each pixel can be leveraged, such as RGB values instead of just intensity. We also want to design an active learning mechanism so that the user will be encouraged to label patches that our algorithm is most uncertain about. This will further reduce the efforts of humans and also improve the flood mapping quality.

- **Crowd Sourcing Experiments.** Some satellite images might be difficult for one person to label and there might be uncertainties and confusions at some parts of images (water or land) due to various reasons, such

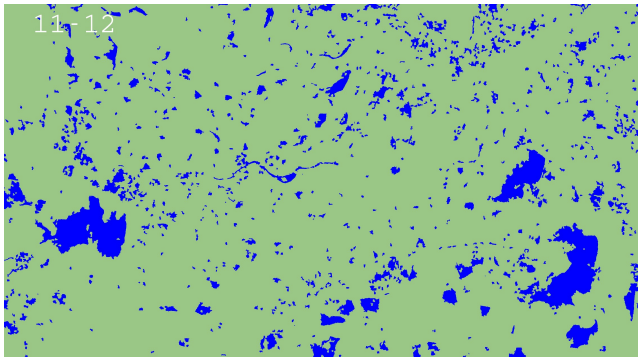
[4]http://web.cse.ohio-state.edu/~liangji/floodmap.html



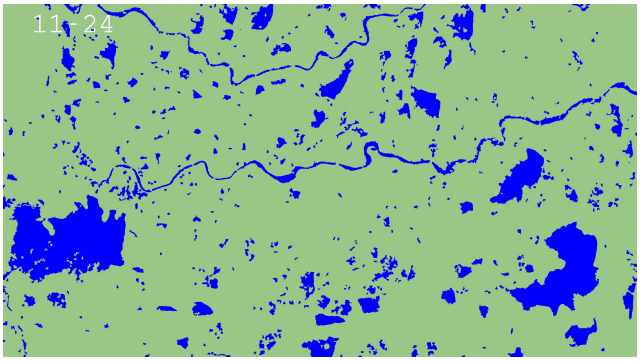
(a) 10/19



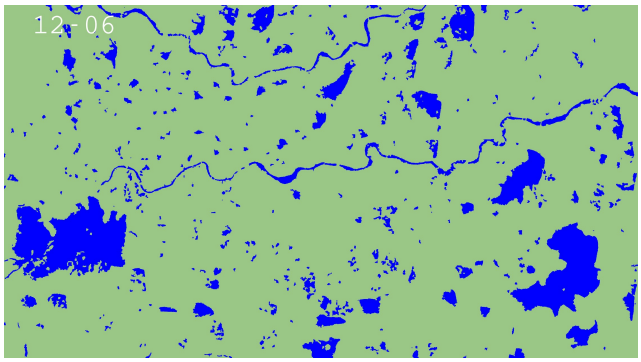
(b) 10/31



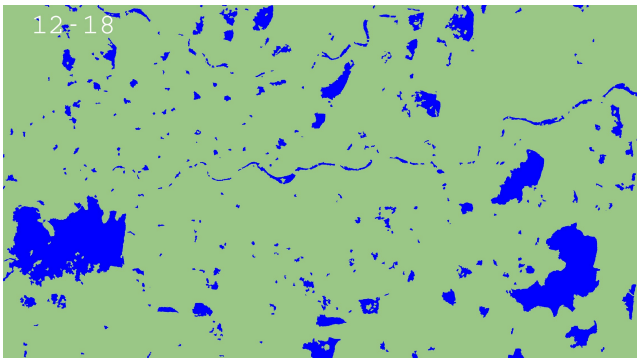
(c) 11/12



(d) 11/24

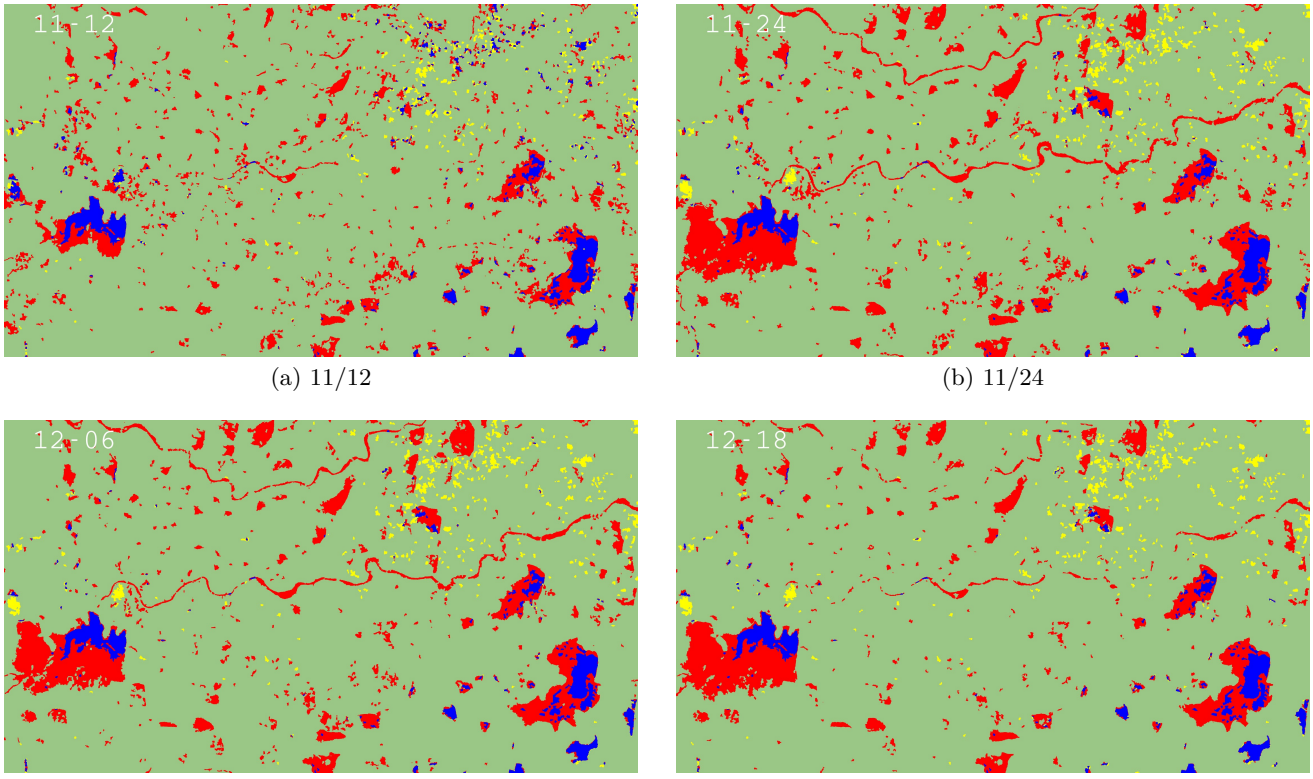


(e) 12/06



(f) 12/18

Figure 5: Results of our segmentation algorithm on satellite images from 10/19/2015 to 12/18/2015. One image for every 12 days.



(a) 11/12 (b) 11/24

(c) 12/06 (d) 12/18

Figure 6: Water area changes from 11/12/2015 to 12/18/2015 using 10/31/2015 as the baseline. One image for every 12 days. Red color indicates areas that were land on 10/30/2015 but were water on the given date, while yellow color indicates areas that were water on 10/30/2015 but were land on the given date. Blue and green represent areas that were originally water or land on 10/30/2015 and remain so on the given date.

as the limitation of resolution. Motivated by this, we plan to lauch a crowd sourcing experiment at platform such as Amazon Mechanical Turk, where we ask different people to help interactively label the satellite images. The segmentation results on the same image are then aggregated. We employ more humans to label those images that involve more conflicts.

- **Incorporating Social Media Information.** During the flood, social media users might publish useful information on social media, which can potentially provide supervision to our method. For example, users might publish tweets on Twitter about the flood in a specific region, and this information can be used as a marker in our method. This means that the information on flood from social media can be used as supervision and labeled markers for the flood mapping approach.

6. CONCLUSION

In this paper, we provide an effective and efficient solution to the flood mapping problem by leveraging human guidance. We generate patches using a graph-based approach and adopt a semi-supervised algorithm involving human guidance to label the patches. Our results show that our algorithm can correctly segment out water and land areas with less noise, compared to other baselines. Further dynamic analysis reveals that it can effectively detect the flooded areas.

Acknowledgements. This work is supported by NSF Award NSF-EAR-1520870 and NSF-DMS-1418265. We also thank Desheng Liu and Jiayong Liang for useful discussions and help with data collection.

7. REFERENCES

[1] A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan 1979.

[2] S. S. Al-Amri, N. V. Kalyankar, et al. Image segmentation by using threshold techniques. *arXiv preprint arXiv:1005.4020*, 2010.

[3] M. Baatz and A. Schäpe. Multiresolution segmentation: an optimization approach for high quality multi-scale image segmentation, 2000.

[4] A. M. Bensaid, L. O. Hall, J. C. Bezdek, and L. P. Clarke. Partially supervised clustering for image segmentation. *Pattern Recognition*, 29(5):859–871, 1996.

[5] S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. *OPTICAL ENGINEERING-NEW YORK-MARCEL DEKKER INCORPORATED-*, 34:433–433, 1992.

[6] Y. Boykov and G. Funka-Lea. Graph cuts and efficient nd image segmentation. *International journal of computer vision*, 70(2):109–131, 2006.

[7] A. Browet, P. A. Absil, and P. Van Dooren. *Combinatorial Image Analysis: 14th International Workshop, IWCIA 2011, Madrid, Spain, May 23-25, 2011. Proceedings*, chapter Community Detection for Hierarchical Image Segmentation, pages 358–371. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[8] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1124–1131. IEEE, 2005.

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[10] N. Grira, M. Crucianu, and N. Boujemaa. Unsupervised and semi-supervised clustering: a brief survey. *A review of machine learning techniques for processing multimedia content, Report of the MUSCLE European Network of Excellence (FP6)*, 2004.

[11] S. Hallegatte, C. Green, R. J. Nicholls, and J. Corfee-Morlot. Future flood losses in major coastal cities. *Nature climate change*, 3(9):802–806, 2013.

[12] J.-B. Henry, P. Chastanet, K. Fellah, and Y.-L. Desnos. Envisat multi-polarized asar data for flood mapping. *International Journal of Remote Sensing*, 27(10):1921–1929, 2006.

[13] J. N. Kapur, P. K. Sahoo, and A. K. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer vision, graphics, and image processing*, 29(3):273–285, 1985.

[14] G. A. Lazarova. Semi-supervised image segmentation. In *Artificial Intelligence: Methodology, Systems, and Applications*, pages 59–68. Springer, 2014.

[15] A. Rosenfeld and P. De La Torre. Histogram concavity analysis as an aid in threshold selection. *Systems, Man and Cybernetics, IEEE Transactions on*, (2):231–235, 1983.

[16] V. Satuluri and S. Parthasarathy. Scalable graph clustering using stochastic flows: applications to community discovery. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 737–746. ACM, 2009.

[17] S. B. Serpico, S. Dellepiane, G. Boni, G. Moser, E. Angiati, and R. Rudari. Information extraction from remote sensing images for flood monitoring and damage evaluation. *Proceedings of the IEEE*, 100(10):2946–2970, 2012.

[18] M. I. Sezan. A peak detection algorithm and its application to histogram-based image data reduction. *Computer vision, graphics, and image processing*, 49(1):36–51, 1990.

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[20] S. P. Simonovic and P. Eng. Role of remote sensing in disaster management. 2002.

[21] D.-M. Tsai. A fast thresholding selection procedure for multimodal and unimodal histograms. *Pattern Recognition Letters*, 16(6):653–666, 1995.

[22] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. *AAAI/IAAI*, 1097, 2000.

[23] Y. Wang, J. Colby, and K. Mulcahy. An efficient method for mapping flood extent in a coastal floodplain using landsat tm and dem data. *International Journal of Remote Sensing*, 23(18):3681–3696, 2002.

SIDE: A Web App for Interactive Visual Data Exploration with Subjective Feedback

Jefrey Lijffijt[1] Bo Kang[1] Kai Puolamäki[2] Tijl De Bie[1]
[1] Data Science Lab, Ghent University, Belgium
[2] Finnish Institute of Occupational Health, Finland
{jefrey.lijffijt;bo.kang;tijl.debie}@ugent.be, kai.puolamaki@ttl.fi

ABSTRACT

Data visualization and iterative/interactive data mining are growing rapidly in attention, both in research as well as in industry. However, integrated methods and tools that combine advanced visualization and/or interaction with data mining techniques are rare, and those that exist are specialized to a single problem or domain. We present SIDE, a generic tool for Subjective Interactive Data Exploration, which lets users explore high dimensional data via subjectively informative two-dimensional data visualizations. In contrast to most visualization tools, it is not based on the traditional dogma of manually zooming and rotating data. Instead, the tool initially presents the user with an 'interesting' projection, and then allows users to flexibly and intuitively express their interests or beliefs using visual interactions that update/constrain a background model of the data. These constraints expressed by the user are then taken into account by a projection-finding algorithm employing data randomization to compute a new 'interesting' projection. This process can be iterated until the user runs out of time or finds that the difference between the randomized data and the real data is no longer interesting. We present the tool by means of two case studies, one controlled study on synthetic data and another on real census data.

Keywords

Exploratory Data Mining; Dimensionality Reduction; Data Randomization; Subjective Interestingness

1. INTRODUCTION

Data visualization and iterative/interactive data mining are both mature, actively researched topics of great practical importance. However, while progress in both fields is abundant (see Section 4), methods that combine iterative data mining with visualization and interaction are rare; only a few tools designed for specific problem domains exist.

Yet, tools that combine state-of-the-art data mining with visualization and interaction are highly desirable as they would maximally exploit the strengths of both human data analysts and computer algorithms. Humans are unmatched in spotting interesting patterns in low-dimensional visual representations, but poor at reading high-dimensional data, while computers excel in manipulating high-dimensional data and are weaker at identifying patterns that are truly relevant to the user. A symbiosis of human analysts and well-designed computer systems thus promises to provide an efficient way of navigating the complex information space hidden within high-dimensional data [17].

IDEA Workshop, SIGKDD '16, August 14, 2016, San Francisco, CA, USA

DOI: --

Contributions.

In this paper we introduce a generically applicable method for finding interesting projections of data, given some prior knowledge about that data, and we introduce a tool that demonstrates the proposed approach for interactive visual exploration of (high-dimensional) data. The underlying idea is that the analysis process is iterative, and during each iteration there are three steps. The hypothesis is that throughout the iterations, the user builds up an increasingly accurate understanding of the data. This understanding is explicated in the *background model*, which is used at the beginning of each iteration in order to find a maximally informative projection. More generally, the background model is a representation for the user's *belief state*. The tool works as indicated in Figure 1. Details of all steps are given below.

Step 1. The tool initially presents the user with an 'interesting' projection of the data, visualized as a scatter plot. Here, interestingness is formalized with respect to the initial belief state.

Step 2. On investigation of this scatter plot, the user may take note of some features of the data that contrast with, or add to, their beliefs about the data. We will refer to such features as *patterns*. The user then interacts with the tool

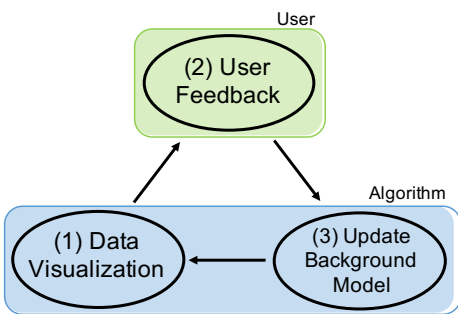


Figure 1: The three steps of SIDE's operation cycle.

to indicate what patterns they have seen and assimilated.
Step 3. The tool updates the background model according to the user feedback, in order to reflect the newly assimilated information.
Next iteration. Then the most interesting projection with respect to this updated background model can be computed, and the cyclic process iterates until the user runs out of time or finds that background model (and thus the user's belief state) explains everything the user is currently interested in.

Formalization of the background model.

A crucial challenge in the realization of such a tool is the formalization of the background model. To allow the process to be iterative, the formalization has to allow for the model to be updated after a user has been provided with new information (i.e., shown a visualization) and given feedback on it. There exist two frameworks for iterative data mining: FORSIED [3, 4] and a framework that has no name yet, but which we will refer to as CORAND [7, 13], for COnstrained RANDomization. In both cases, the background model is a probability distribution over data sets and the user beliefs are modelled as a set of *constraints* on that distribution.

The CORAND approach is to specify a randomization procedure that, when applied to the data, does not affect how plausible the user would deem it to be. That is, the user's beliefs should be satisfied, and otherwise the data should be shuffled as much as possible. Given an appropriate randomization scheme, we can then find interesting remaining structure that is not yet known to the user by contrasting the real data against the randomized data. New beliefs can be incorporated in the background model by adding corresponding constraints to the randomization procedure, ensuring that the patterns observed by the user are present also in the subsequent randomized data.

An illustrative example.

As an example, consider a synthetic data set that consists of 1000 ten-dimensional data vectors of which dimensions 1–4 can be clustered into five clusters, dimensions 5–6 into four clusters *involving different subsets of data points*, and of which dimensions 7–10 are Gaussian noise. All dimensions have equal variance.

We designed this example to illustrate the two types of feedback that a user can give in the current implementation of our tool. Additionally, it shows how the tool succeeds in finding interesting projections given previously identified patterns. Thirdly, it also demonstrates how the user interactions meaningfully affect subsequent visualizations. In this example we aim to provide an overview of how the tool works, technical details are presented in Section 2.

We observe that the first projection computed by SIDE maps the data onto a two-dimensional (2D) subspace of the dimensions 1–4 (Figure 2a), i.e., to a subspace of the space where the data is clustered into 5 clusters. This is indeed sensible, as the structure within this 4D subspace is arguably the most striking.

We then consider two possible user actions (Step 2, Figure 2b). In the first scenario (Figure 2 left path), the user marks all points within each cluster (one cluster at a time), indicating they have taken note of the positions of these groups of points *within this particular projection*. In the second scenario (Figure 2 right path), the user gives the feedback that these points appear to be clustered *in this projection and possibly also in other dimensions*.

Both these 'pattern types' lead to a set of constraints on the randomization procedure. The effect of these constraints is identical with respect to the current 2D projection (Figure 2c): the projections of the randomized points onto this plane are identical to the projections of the original points onto this plane. Not visible though is that in the second scenario the randomization is restricted also in orthogonal dimensions (possibly different ones for different clusters), to account for the user feedback that also orthogonal subspaces that yield the same clusters are not interesting anymore.

The subsequent most interesting projection is different in the two scenarios (Figure 2d). In the first scenario, the remaining cluster structure within dimensions 1–4 is shown. However, in the second scenario this cluster structure is fully explained by the constraints, and as a result, the cluster structure in dimensions 5–6 being is shown instead.

The difference can be observed in the visualization because on the left three clusters are pure and one is mixed (an artefact of how we chose the cluster centers). Yet, on the right all clusters are mixed with respect the previous clustering. This indeed shows the two clusterings in dimensions 1–4 and dimensions 5–6 are unrelated.

Outline of this paper.

As discussed in Section 2, three challenges had to be addressed to use the CORAND approach: (1) defining intuitive pattern types (constraints) that can be observed and specified based on a scatter plot of a two-dimensional projection of the data; (2) defining a suitable randomization scheme, that can be constrained to take account of such patterns; and (3) a way to identify the most interesting projections given the background model. The evaluation with respect to usefulness as well as computational properties of the resulting system is presented in Section 3. Experiments were conducted both on synthetic data and on a census dataset. Finally, related work and conclusions are discussed in Sections 4 and 5, respectively.

NB. This manuscript is an integration of two publications that are to appear in the Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery [10, 16].

2. METHODS

We will use the notational convention that upper case bold face symbols represent matrices, lower case bold face symbols represent column vectors, and lower case standard face symbols represent scalars. We assume that our data set consists of n d-dimensional data vectors $\mathbf{x}_i$. The data set is represented by a real matrix $\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_n^T \end{pmatrix}^T \in \mathbb{R}^{n \times d}$. More generally, we will denote the transpose of the ith row of any matrix $\mathbf{A}$ as $\mathbf{a}_i$ (i.e., $\mathbf{a}_i$ is a column vector). Finally, we will use the shorthand notation $[n] = \{1, \ldots, n\}$.

2.1 Projection tile patterns in two flavours

In the interaction step, the proposed system allows users to declare that they have become aware of (and thus are no longer interested in seeing) the value of the projections of a set of points onto a specific subspace of the data space. We call such information a *projection tile* pattern for reasons that will become clear later. A projection tile parametrizes a set of constraints to the randomization.

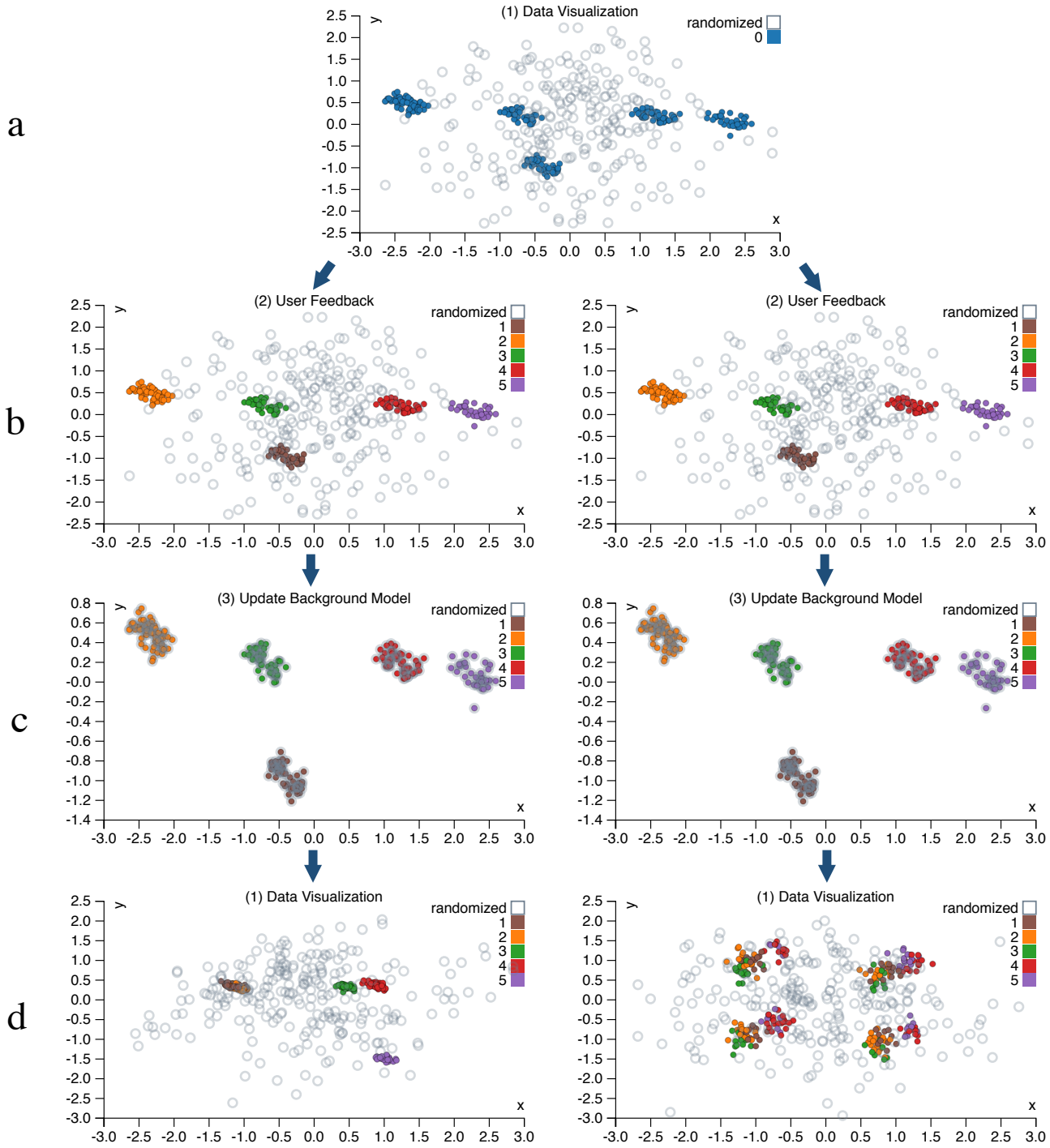


Figure 2: Two user interaction scenarios for the toy data set. Solid dots represent actual data vectors, whereas open circles represent vectors from the randomized data. Row (a) shows the first visualization, which is the starting point for both scenarios. Row (b) shows the sets of data points marked by the user. Although not shown, on the left the user gives feedback to incorporate the selected cluster structure in the currently shown dimensions, while on the right the feedback is that the user expects the cluster structure to generalize to other unshown dimensions. Row (c) shows the newly randomized data and the original data projected still in the same subspace. As expected, the randomized data fully aligns with the real data. Then, row (d) shows the most interesting visualization given the specified patterns (constraints). The left path shows the scenario when the user assumes nothing beyond the values of the data points in the projection in row (a), whereas the right path shows the scenario when the user assumes each of these sets of points may be clustered in other dimensions as well.

Formally, a projection tile pattern, denoted τ, is defined by a k-dimensional (with $k \leq d$ and $k = 2$ in the simplest case) subspace of $\mathbb{R}^d$, and a subset of data points $\mathcal{I}_\tau \subseteq [n]$. We will formalize the k-dimensional subspace as the column space of an orthonormal matrix $\mathbf{W}_\tau \in \mathbb{R}^{d\times k}$ with $\mathbf{W}_\tau^T\mathbf{W}_\tau = \mathbf{I}$, and can thus denote the projection tile as $\tau = (\mathbf{W}_\tau, \mathcal{I}_\tau)$. The proposed tool provides two ways in which the user can define the projection vectors $\mathbf{W}_\tau$ for a projection tile τ.

2D tiles.

The first approach simply chooses $\mathbf{W}_\tau$ as the two weight vectors defining the projection within which the data vectors belonging to $\mathcal{I}_\tau$ were marked. This approach allows the user to simply specify that they have taken note of the positions of that set of data points within this projection. The user makes no further assumptions—they assimilate solely what they see without drawing conclusions not supported by direct evidence, see Figure 2b (left).

Clustering tiles.

It seems plausible, however, that when the marked points are tightly clustered, the user concludes that these points are clustered *not just within the two dimensions shown* in the scatter plot. To allow the user to express such belief, the second approach takes $\mathbf{W}_\tau$ to additionally include a basis for other dimensions along which these data points are strongly clustered, see Figure 2b (right). This is achieved as follows.

Let $\mathbf{X}(\mathcal{I}_\tau, :)$ represent a matrix containing the rows indexed by elements from $\mathcal{I}_\tau$ from $\mathbf{X}$. Let $\mathbf{W} \in \mathbb{R}^{d\times 2}$ contain the two weight vectors onto which the data was projected for the current scatter plot. In addition to $\mathbf{W}$, we want to find any other dimensions along which these data vectors are clustered. These dimensions can be found as those along which the variance of these data points is not much larger than the variance of the projection $\mathbf{X}(\mathcal{I}_\tau, :)\mathbf{W}$.

To find these dimensions, we first project the data onto the subspace orthogonal to $\mathbf{W}$. Let us represent this subspace by a matrix with orthonormal columns, further denoted as $\mathbf{W}^\perp$. Thus, ${\mathbf{W}^\perp}^T\mathbf{W}^\perp = \mathbf{I}$ and $\mathbf{W}^T\mathbf{W}^\perp = \mathbf{0}$. Then, Principal Component Analysis (PCA) is applied to the resulting matrix $\mathbf{X}(\mathcal{I}_\tau, :)\mathbf{W}^\perp$. The principal directions corresponding to a variance smaller than a threshold are then selected and stored as columns in a matrix $\mathbf{V}$. In other words, the variance of each of the columns of $\mathbf{X}(\mathcal{I}_\tau, :)\mathbf{W}^\perp\mathbf{V}$ is below the threshold.

The matrix $\mathbf{W}_\tau$ associated to the projection tile pattern is then taken to be:

$$\mathbf{W}_\tau = \begin{pmatrix}\mathbf{W} & \mathbf{W}^\perp\mathbf{V}\end{pmatrix}.$$

The threshold on the variance used could be a tunable parameter, but was set here to twice the average of the variance of the two dimensions of $\mathbf{X}(\mathcal{I}_\tau, :)\mathbf{W}$.

2.2 The randomization procedure

Here we describe the approach to randomizing the data. The randomized data should represent a sample from an implicitly defined background model that represents the user's belief state about the data. Initially, our approach assumes the user merely has an idea about the overall scale of the data. However, throughout the interactive exploration, the patterns in the data described by the projection tiles will be maintained in the randomization.

Initial randomization.

The proposed randomization procedure is parametrized by n orthogonal rotation matrices $\mathbf{U}_i \in \mathbb{R}^{d\times d}$, where $i \in [n]$, and the matrices satisfy $(\mathbf{U}_i)^T = (\mathbf{U}_i)^{-1}$. We further assume that we have a bijective mapping $f : [n] \times [d] \mapsto [n] \times [d]$ that can be used to permute the indices of the data matrix. The randomization proceeds in three steps:

Random rotation of the rows Each data vector $\mathbf{x}_i$ is rotated by multiplication with its corresponding random rotation matrix $\mathbf{U}_i$, leading to a randomised matrix $\mathbf{Y}$ with rows $\mathbf{y}_i^T$ that are defined by:

$$\forall i : \ \mathbf{y}_i = \mathbf{U}_i\mathbf{x}_i.$$

Global permutation The matrix $\mathbf{Y}$ is further randomized by randomly permuting all its elements, leading to the matrix $\mathbf{Z}$ defined as:

$$\forall i, j : \ \mathbf{Z}_{i,j} = \mathbf{Y}_{f(i,j)}.$$

Inverse rotation of the rows Each randomised data vector in $\mathbf{Z}$ is rotated with the inverse rotation applied in step 1, leading to the fully randomised matrix $\mathbf{X}^*$ with rows $\mathbf{x}_i^*$ defined as follows in terms of the rows $\mathbf{z}_i^T$ of $\mathbf{Z}$:

$$\forall i : \ \mathbf{x}_i^* = {\mathbf{U}_i}^T\mathbf{z}_i.$$

The random rotations $\mathbf{U}_i$ and the permutation f are sampled uniformly at random from all possible rotation matrices and permutations, respectively.

Intuitively, this randomization scheme preserves the scale of the data points. Indeed, the random rotations leave their lengths unchanged, and the global permutation subsequently shuffles the values of the d components of the rotated data points. Note that without the permutation step, the two rotation steps would undo each other such that $\mathbf{X}^* = \mathbf{X}$. Thus, it is the combined effect that results in a randomization of the data set.

The random rotations may seem superfluous: the global permutation randomizes the data so dramatically that the added effect of the rotations is relatively unimportant. However, their role is to make it possible to formalize the growing understanding of the user as simple constraints on this randomization procedure, as discussed next.

Accounting for one projection tile.

Once the user has assimilated the information in a projection tile $\tau = (\mathbf{W}_\tau, \mathcal{I}_\tau)$, the randomization scheme should incorporate this information by ensuring that it is present also in all randomized versions of the data. This ensures that the randomized data is a sample from a distribution representing the user's belief state about the data. This is achieved by imposing the following *constraints* on the parameters defining the randomization:

Rotation matrix constraints For each $i \in \mathcal{I}_\tau$, the component of $\mathbf{x}_i$ that is within the column space of $\mathbf{W}_\tau$ must be mapped onto the first k dimensions of $\mathbf{y}_i = \mathbf{U}_i\mathbf{x}_i$ by the rotation matrix $\mathbf{U}_i$. This can be achieved by ensuring that:

$$\forall i \in \mathcal{I}_\tau : \ \mathbf{W}_\tau^T\mathbf{U}_i = \begin{pmatrix}\mathbf{I} & \mathbf{0}\end{pmatrix}. \tag{1}$$

This explains the name *projection tile*: the information to be preserved in the randomization is concentrated

in a 'tile' (i.e. the intersection of a set of rows and a set of columns) in the intermediate matrix $\mathbf{Y}$ created during the randomization procedure.

Permutation constraints The permutation should not affect any matrix cells with row indices $i \in \mathcal{I}_\tau$ and columns indices $j \in [k]$:

$$\forall i \in \mathcal{I}_\tau, j \in [k] : \; f(i,j) = (i,j). \tag{2}$$

PROPOSITION 1. *Using the above constraints on the rotation matrices $\mathbf{U}_i$ and the permutation f, it holds that:*

$$\forall i \in \mathcal{I}_\tau, \mathbf{x}_i^T \mathbf{W}_\tau = \mathbf{x}_i^{*T} \mathbf{W}_\tau. \tag{3}$$

Thus, the values of the projections of the points in the projection tile remain unaltered by the constrained randomization. Hence, the randomization keeps the user's beliefs intact. We omit the proof as the more general Proposition 2 is provided with proof further below.

Accounting for multiple projection tiles.

Throughout subsequent iterations, additional projection tile patterns will be specified by the user. A set of tiles τ_i for which $\mathcal{I}_{\tau_i} \cap \mathcal{I}_{\tau_j} = \emptyset$ if $i \neq j$ is straightforwardly combined by applying the relevant constraints on the rotation matrices to the respective rows. When the sets of data points affected by the projection tiles overlap though, the constraints on the rotation matrices need to be combined. The aim of such a combined constraint should be to preserve the values of the projections onto the projection directions for *each* of the projection tiles a data vector was part of.

The combined effect of a set of tiles will thus be that the constraint on the rotation matrix $\mathbf{U}_i$ will vary per data vector, and depends on the set of projections $\mathbf{W}_\tau$ for which $i \in \mathcal{I}_\tau$. More specifically, we propose to use the following constraint on the rotation matrices:

Rotation matrix constraints Let $\mathbf{W}_i \in \mathbb{R}^{d \times d_i}$ denote a matrix of which the columns are an orthonormal basis for space spanned by the union of the columns of the matrices $\mathbf{W}_\tau$ for τ with $i \in \mathcal{I}_\tau$. Thus, for any i and $\tau : i \in \mathcal{I}_\tau$, it holds that $\mathbf{W}_\tau = \mathbf{W}_i \mathbf{v}_\tau$ for some $\mathbf{v}_\tau \in \mathbb{R}^{d_i}$. Then, for each data vector i, the rotation matrix $\mathbf{U}_i$ must satisfy:

$$\forall i \in \mathcal{I}_\tau : \; \mathbf{W}_i^T \mathbf{U}_i = (\mathbf{I} \;\; \mathbf{0}). \tag{4}$$

Permutation constraints Then the permutation should not affect any matrix cells in row i and columns $[d_i]$:

$$\forall i \in [n], j \in [d_i] : \; f(i,j) = (i,j).$$

PROPOSITION 2. *Using the above constraints on the rotation matrices $\mathbf{U}_i$ and the permutation f, it holds that:*

$$\forall \tau, \forall i \in \mathcal{I}_\tau, \mathbf{x}_i^T \mathbf{W}_\tau = \mathbf{x}_i^{*T} \mathbf{W}_\tau.$$

PROOF. We first show that $\mathbf{x}_i^{*T} \mathbf{W}_i = \mathbf{x}_i^T \mathbf{W}_i$:

$$\mathbf{x}_i^{*T} \mathbf{W}_i = \mathbf{z}_i^T \mathbf{U}_i^T \mathbf{W}_i = \mathbf{z}_i^T \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \end{pmatrix}$$

$$= \mathbf{z}_i(1:d_i)^T = \mathbf{y}_i(1:d_i)^T = \mathbf{y}_i^T \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \end{pmatrix} = \mathbf{x}_i^T \mathbf{W}_i.$$

The result now follows from the fact that $\mathbf{W}_\tau = \mathbf{W}_i \mathbf{v}_\tau$ for some $\mathbf{v}_\tau \in \mathbb{R}^{d_i}$. □

Technical implementation of the randomization.

To ensure the randomization can be carried out efficiently throughout the process, note that the matrix $\mathbf{W}_i$ for the $i \in \mathcal{I}_\tau$ for a new projection tile τ can be updated by computing an orthonormal basis for $(\mathbf{W}_i \;\; \mathbf{W})$. Such a basis can be found efficiently as the columns of $\mathbf{W}_i$ in addition to the columns of an orthonormal basis of $\mathbf{W} - \mathbf{W}_i^T \mathbf{W}_i \mathbf{W}$ (the components of $\mathbf{W}$ orthogonal to $\mathbf{W}_i$), the latter of which can be computed using the QR-decomposition.

Additionally, note that the tiles define an equivalence relation over the row indices, in which i and j are equivalent if they were included in the same set of projection tiles so far. Within each equivalence class, the matrix $\mathbf{W}_i$ will be constant, such that it suffices to compute it only once, keeping track of which points belong to which equivalence class.

2.3 Visualization: Finding the most interesting two-dimensional projection

Given the data set $\mathbf{X}$ and the randomized data set $\mathbf{X}^*$, it is now possible to quantify the extent to which the empirical distribution of a projection $\mathbf{X}\mathbf{w}$ and $\mathbf{X}^*\mathbf{w}$ onto a weight vector $\mathbf{w}$ differ. There are various ways in which this difference can be quantified. We investigated a number of possibilities and found that the L_1-distance between the cumulative distribution functions works well in practice. Thus, with $F_\mathbf{x}$ the empirical cumulative distribution function for the set of values in $\mathbf{x}$, the optimal projection is found by solving:

$$\max_{\mathbf{w}} \left\| F_{\mathbf{X}\mathbf{w}} - F_{\mathbf{X}^*\mathbf{w}} \right\|_1 .$$

The second dimension of the scatter plot can be sought by optimizing the same objective while requiring it to be orthogonal to the first dimension.

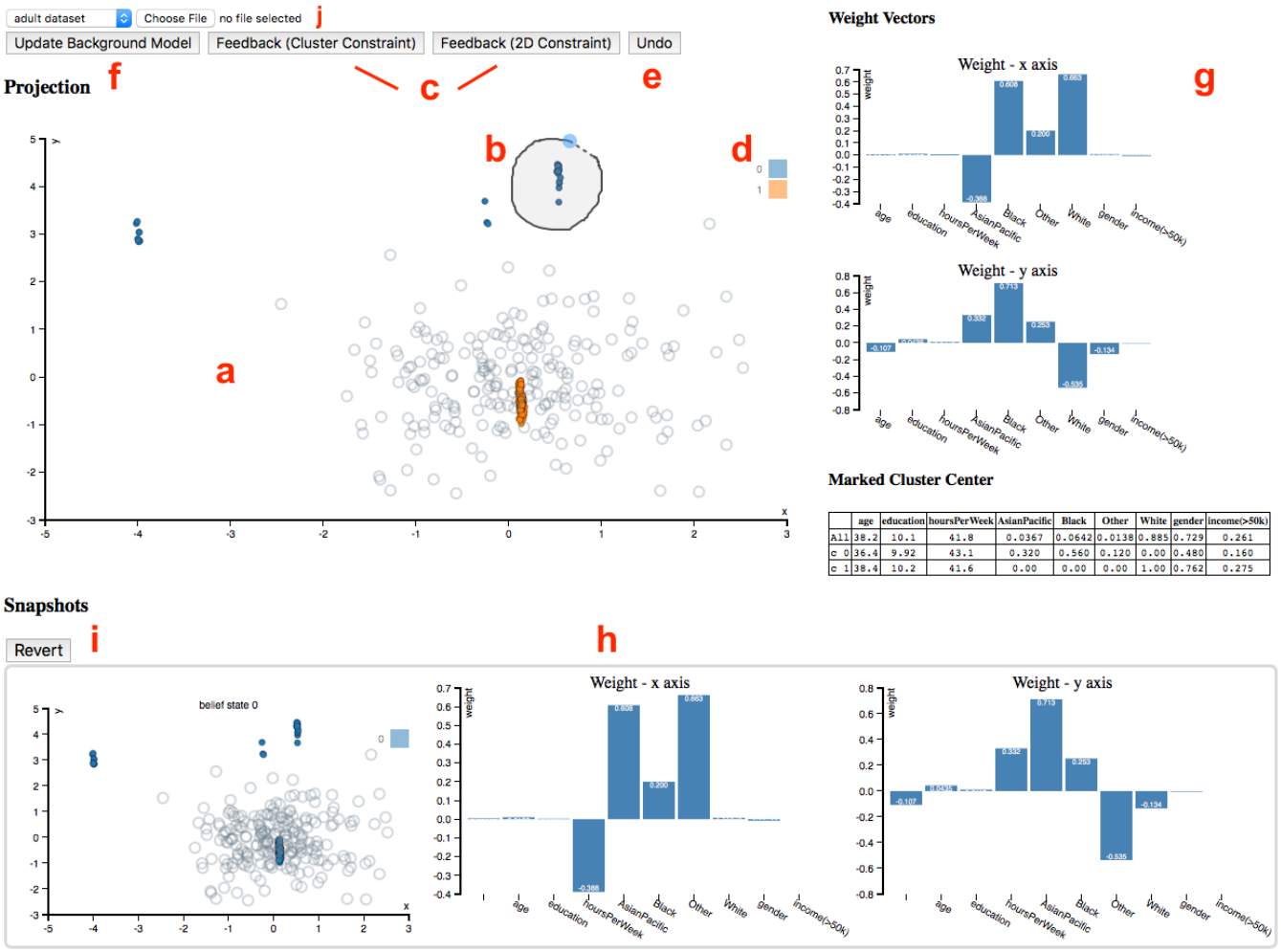
We are unaware of any special structure of this optimization problem that makes solving it particularly efficient. Yet, using the standard quasi-Newton solver in R [18] with random initialization and default settings (the general-purpose `optim` function with `method="BFGS"`) already yields satisfactory results, as shown in the experiments below.

2.4 Interface

The full interface of SIDE is shown in Figure 3. SIDE was designed according to three principles for visually controllable data mining [17], which essentially says that both the model and the interactions should be transparent to users, and that the analysis method should be fast enough such that the user does not lose its trail of thought.

The main component is the interactive scatter plot (3a). The scatter plot visualizes the projected data (solid dots) and the randomized data (open gray circles) in the current 2D projection. By drawing circles (3b), the user can highlight data points to define a *projection tile pattern*. Once a set of points is marked, the user can press either of the two feedback buttons (3c), to indicate these points form a cluster. If the user thinks the points are clustered only in the shown projection, they click '2D Constraint', while 'Cluster Constraint' indicates they expect that these points will be clustered in other dimensions as well.

To identify the defined clusters, data points associated with the same feedback (i.e., user's belief) are filled by the same color (3d), and their statistics are shown in a table. The user can define multiple clusters in a single projection, and they can also *undo* (3e) the feedback. Once a user finishes exploring the current projection, they can press 'Up-



	age	education	hoursPerWeek	AsianPacific	Black	Other	White	gender	income(>50k)
All	38.2	10.1	41.8	0.0367	0.0642	0.0138	0.885	0.729	0.261
c 0	36.4	9.92	43.1	0.320	0.560	0.120	0.00	0.480	0.160
c 1	38.4	10.2	41.6	0.00	0.00	0.00	1.00	0.762	0.275

Figure 3: Layout of our web app SIDE, which contains the data visualization and interaction area (a–f), projection meta information (g), and timeline (h).

date Background Model' (3f). Then, the background model is updated with the provided feedback and a new scatter plot is computed and presented to the user, etc.

A few extra features are provided to assist the data exploration process: to gain an understanding of a projection, the weight vectors associated with the projection axes are plotted as bar charts (3g). At the bottom of 3g, a table lists the mean vectors of each colored point set (i.e., cluster). The exploration history is maintained by taking snapshots of the background model when updated, together with the associated data projection (scatter plot) and bar charts (weight vectors). This history in reverse chronological order is illustrated in Figure 3h.

The tool also allows a user to click and revert back to a certain snapshot (3i), to restart from that time point. This allows the user to discover different aspects of a dataset more consistently. Finally, custom datasets can be selected for analysis from the drop-down menu (3j). Currently our tool only works with CSV files and it automatically sub-samples the custom data set so that the interactive experience is not compromised. By default, two datasets are preloaded so that users can get familiar with the tool.

3. EXPERIMENTS

We present two case studies to illustrate the framework and its utility. The case studies are completed with the a JavaScript version of our tool, which is available freely online, along with the used data for reproducibility.[1]

3.1 Synthetic data case study

This section gives an extended discussion of the illustrative example from the introduction, namely the synthetic data case study. The data is described in Section 1. The first projection shows that the projected data (solid blue dots in Figure 2a) differs strongly from the randomized data (open gray circles). The weight vectors defining the projection, shown in the 1st row of Table 1, contain large weights in dimensions 1–4. Therefore, the cluster structure seen here mainly corresponds to dimensions 1–4 of the data.

A user can indicate this insight by means of a *clustering tile* for each of the clustered sets of data points (2b, right). Encoding this into the background model, results in a randomization, where the randomized points perfectly

[1]http://www.interesting-patterns.net/forsied/a-tool-for-subjective-and-interactive-visual-data-exploration/)

Table 1: Projection weight vectors for the synthetic data (Sections 1 and 3.1).

Figure	axis	1	2	3	4	5	6	7	8	9	10
2a	X	**0.194**	**0.545**	**-0.630**	**0.499**	-0.119	-0.041	0.057	0.001	-0.029	0.003
	Y	**-0.269**	**-0.754**	**-0.481**	**0.340**	0.091	-0.004	0.016	-0.057	0.003	0.005
2d (left)	X	**0.143**	**-0.118**	0.005	**0.981**	0.001	-0.013	-0.031	-0.022	0.044	-0.031
	Y	**-0.245**	**0.448**	**0.854**	0.088	0.004	-0.001	0.005	0.008	-0.043	0.023
2d (right)	X	0.121	0.019	-0.232	0.017	**-0.963**	-0.008	0.022	0.023	0.037	0.004
	Y	-0.139	-0.067	-0.369	-0.082	0.111	**-0.898**	-0.083	0.086	0.005	-0.017

Table 2: Projection weight vectors for the UCI Adult data (Section 3.2).

Figure	axis	Age	Edu.	h/w	EG_AsPl	EG_Bl.	EG_Oth.	EG_Whi.	Gender	Income
4a	X	-0.039	-0.001	0.001	**0.312**	**-0.530**	**-0.193**	**0.763**	0.017	0.008
	Y	0.004	-0.004	-0.002	**0.816**	**-0.141**	**0.465**	**-0.313**	-0.011	0.002
4c	X	0.081	-0.028	-0.022	-0.259	-0.233	-0.104	-0.380	**-0.846**	-0.001
	Y	-0.590	0.541	0.143	-0.233	-0.380	-0.026	-0.293	0.232	0.000
4d	X	0.119	-0.149	0.047	0.102	0.191	0.104	**-0.556**	0.0581	**-0.769**
	Y	**-0.382**	**-0.626**	**-0.406**	0.346	0.317	-0.0287	0.111	-0.248	0.059

Table 3: Mean vectors of user marked clusters for the UCI Adult data (Section 3.2).

Figure	Cluster	Age	Edu.	h/w	EG_AsPl	EG_Bl.	EG_Oth.	EG_Whi.	Gender	Income
4b	top left	35.0	8.67	34.7	0.00	0.00	**1.00**	0.00	0.667	0.333
	bott. left	37.2	9.43	40.3	0.00	**1.00**	0.00	0.00	0.286	0.071
	top right	35.6	1.3	51.1	**1.00**	0.00	0.00	0.00	0.750	0.250
	bott. right	38.4	10.2	41.6	0.00	0.00	0.00	**1.00**	0.762	0.275
4c	left	39.0	10.2	43.3	0.0377	0.0252	0.0126	0.925	**1.00**	0.321
	right	36.0	9.95	37.9	0.0339	0.169	0.0169	0.780	**0.00**	0.102
4d	left	**42.5**	**11.6**	**46.3**	0.00	0.00	0.00	**1.00**	1.00	**1.00**

align with data points (2c, right). The new projection that differs most from this updated background model reveals the four clusters in dimensions 5–6 that the user was not aware of before (2d, right).

If the user does not want to draw conclusions about the points being clustered in dimensions other than those shown, she can use *2D tiles* instead of *clustering tiles* (Figure 2b, left). The updated background model then results in a randomization that is indistinguishable in the given projection from the one with a clustering tile (2c, left), but it results in a different subsequent projection (2d, left). Indeed, this leads to just another view of the five clusters in dimensions 1–4, as confirmed by the large weights for dimensions 1–4 (2nd row of Table 1). Thus, by these simple interactions the user can choose whether she will allow additional exploration of the cluster structure in dimensions 1–4 or if she is now already aware of the cluster structure, in which case the system directs her to the structure occurring in dimensions 5–6. This behavior aligns perfectly with our expectations.

3.2 UCI Adult dataset case study

In this case study, we demonstrate the utility of our method by exploring a real world dataset. The data is compiled from UCI Adult dataset[2]. To ensure the real time interactivity, we sub-sampled 218 data points and selected six features: "Age" $(17-90)$, "Education" $(1-16)$, "HoursPerWeek" $(1-99)$, "Ethnic Group" (White, AsianPacIslander, Black, Other), "Gender" (Female, Male), "Income" $(\geq 50k)$. Among the selected features, "Ethnic Group" is a categorical feature with five categories, "Gender" and "Income" are binary features, the rest are all numeric. To make our method applicable to this dataset, we further binarized the "Ethnic Group" feature (yielding four binary features), and the final dataset consists of 218 points and 9 features.

We assume the user uses clustering tiles throughout the exploration. Each of the patterns discovered during the exploration process thus corresponds to a certain demographic clustering pattern. To illustrate how our tool helps the user rapidly gain an understanding of the data, we discuss the first three iterations of the exploration process. The first projection (Figure 4a) visually consists of four clusters. The user notes that the weight vectors corresponding to the axes of the plot assign large weights to the "Ethnic Group" attributes (Table 2, 1st row). As mentioned, we assume the user marks these points as part of the same clustering tile. When marking the clusters (Figure 4b), the tool informs the user of the mean vectors of the points within each clustering tile. The 1st row of Table 3 shows that each cluster completely represents one out of four ethnic groups, which may corroborate with the user's understanding.

Taking the user's feedback into consideration, a new projection is generated by the tool. The new scatter plot (Figure 4c) shows two large clusters, each consisting of some points from the previous four-cluster structure (points from these four clusters are colored differently). Thus, the new scatter plot elucidates structure not shown in the previous one. Indeed, the weight vectors (2nd row of Table 2) show that the clusters are separated mainly according to the "Gender" attribute. After marking the two clusters separately, the mean vector of each cluster (2nd row of Table 3) again confirms this: the cluster on the left represents male group, and the female group is on the right.

[2]https://archive.ics.uci.edu/ml/datasets/Adult

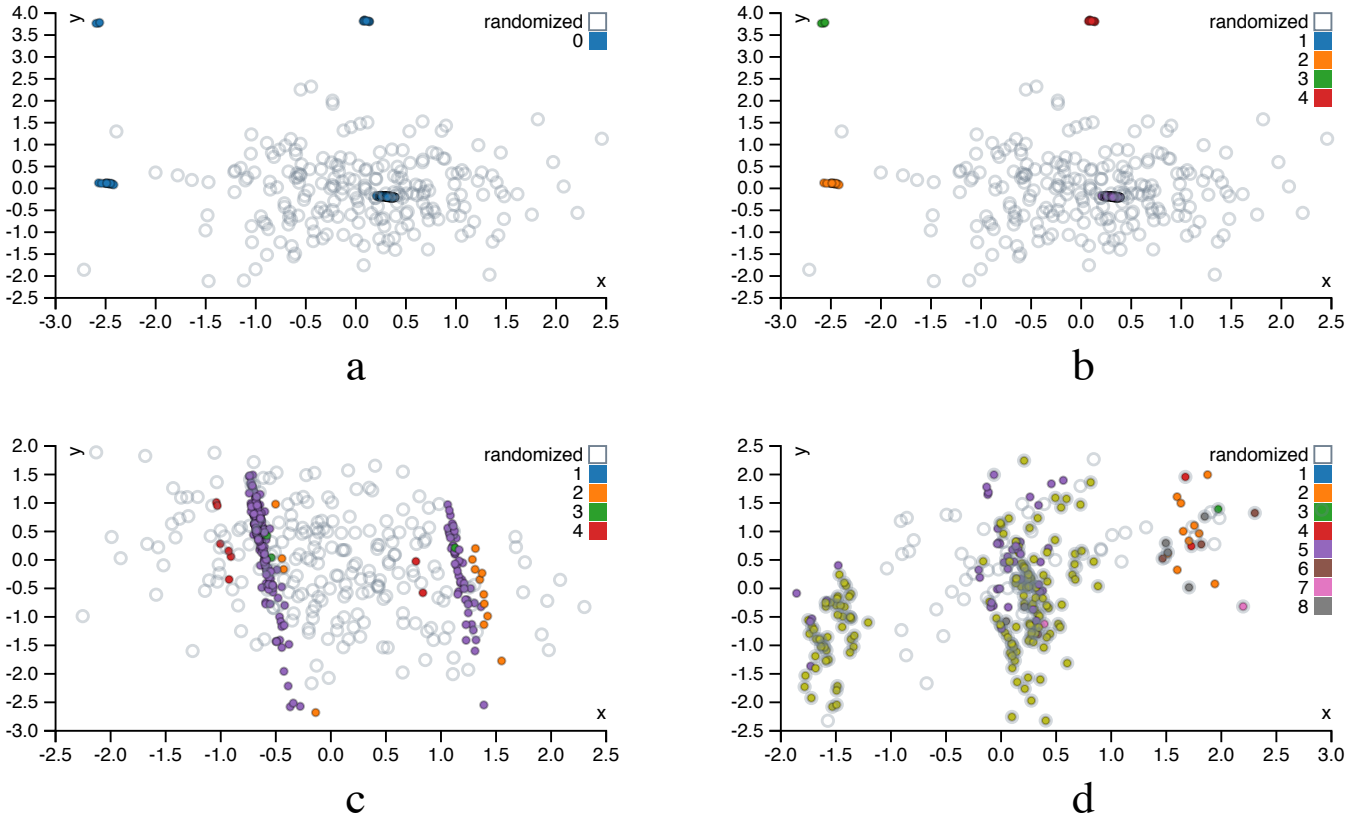


Figure 4: Projections of UCI Adult dataset: (a) projection in the 1*st* iteration, (b) clusters marked by user in the 1*st* iteration, (c) projection in the 2*nd* iteration, and (d) projection in the 3*rd* iteration

The projection in the third iteration (Figure 4d) consists of three clusters, separated only along the x-axis. Interestingly, the corresponding weight vector (3rd row of Table 2) has strongly negative weights for the attributes "Income" and "Ethnic Group - White". This indicates the left cluster mainly represents the people with high income and whose ethnic group is also "White". This cluster has relatively low y-value; according to the weight vector, they are also generally older and more highly educated. These observations are corroborated by the cluster mean (Table 3, 3rd row).

This case study illustrates how the proposed tool facilitates human data exploration by iteratively presenting an informative projection, considering what the user has already learned about the data.

3.3 Performance on synthetic data

Ideally interactive data exploration tools should work in close to real time. This section contains an empirical analysis of an (unoptimized) R implementation of our tool, as a function of the size, dimensionality, and complexity of the data. Note that limits on screen resolution as well as on human visual perception render it useless to display more than of the order of a few hundred data vectors, such that larger data sets can be down-sampled without noticeably affecting the content of the visualizations.

We evaluated the scalability on synthetic data with $d \in \{16, 32, 64, 128\}$ dimensions and $n \in \{64, 128, 256, 512\}$ data points scattered around $k \in \{2, 4, 8, 16\}$ randomly drawn cluster centroids (Table 4). The randomization is done here with the initial background model. The most costly part in randomization is usually the multiplication of orthogonal matrices, indeed, the running time of the randomization scales roughly as nd^{2-3}. The results suggests that the running time of the optimization is roughly proportional to the size of the data matrix nd and that the complexity of data k has here only a minimal effect in the running time of the optimization.

Furthermore, in 90% of the tests, the L_1 loss on the first axis is within 1% of the best L_1 norm out of ten restarts. The optimization algorithm is therefore quite stable, and in practical applications it may well be be sufficient to run the optimization algorithm only once. These results have been obtained with unoptimized and single-threaded R implementation on a laptop having 1.7 GHz Intel Core i7 processor.[3] The performance could probably be significantly boosted by, e.g., carefully optimizing the code and the implementation. Yet, even with this unoptimized code, response times are already of the order of 1 second to 1 minute.

4. RELATED WORK

Dimensionality reduction.

Dimensionality reduction for exploratory data analysis has been studied for decades. Early research into visual exploration of data led to approaches such as multidimensional

[3]The R implementation used to produce Table 4 is available also via the demo page (footnote 1).

Table 4: Median wall clock running times, for randomization and optimization over ten iterations of finding 2D-projections using L_1 loss. Also shown is the number of iterations in which the L_1 norm first component ended up within 1% of the result with the largest L_1 norm (out of 10 tries). A high number indicates the solution quality is stable, even though the actual projections may vary.

		rand.	$k \in \{2, 4, 8, 16\}$	
n	d	(s)	optim. (s)	#tries $\Delta < 1\%$
64	16	0.1	$\{1.0, 1.2, 0.9, 1.2\}$	$\{10, 10, 9, 8\}$
64	32	0.5	$\{1.8, 2.1, 2.4, 2.5\}$	$\{10, 8, 10, 10\}$
64	64	2.5	$\{5.6, 3.5, 4.6, 4.5\}$	$\{10, 9, 10, 8\}$
64	128	11.5	$\{8.9, 10.1, 11.4, 10.2\}$	$\{10, 10, 8, 9\}$
128	16	0.2	$\{2.0, 1.7, 2.4, 2.0\}$	$\{10, 1, 6, 8\}$
128	32	0.8	$\{2.6, 3.5, 4.0, 4.8\}$	$\{9, 10, 10, 10\}$
128	64	5.1	$\{6.7, 5.3, 8.3, 9.6\}$	$\{8, 10, 10, 9\}$
128	128	24.5	$\{13.8, 17.4, 15.2, 20.4\}$	$\{10, 9, 10, 7\}$
256	16	0.4	$\{4.3, 2.6, 3.3, 4.7\}$	$\{10, 8, 10, 9\}$
256	32	1.8	$\{6.3, 8.2, 7.9, 8.8\}$	$\{8, 9, 10, 10\}$
256	64	9.2	$\{12.4, 10.1, 19.2, 16.3\}$	$\{10, 10, 10, 9\}$
256	128	39.9	$\{33.5, 36.3, 30.6, 35.6\}$	$\{10, 9, 8, 9\}$
512	16	0.5	$\{6.7, 6.3, 6.1, 7.5\}$	$\{10, 9, 10, 10\}$
512	32	2.4	$\{16.6, 19.6, 20.2, 17.5\}$	$\{9, 9, 10, 10\}$
512	64	13.6	$\{34.9, 23.5, 22.3, 41.0\}$	$\{10, 10, 8, 7\}$
512	128	68.0	$\{74.5, 68.1, 72.3, 62.8\}$	$\{10, 1, 9, 9\}$

scaling [12, 21] and projection pursuit [6, 9]. Most recent research on this topic (also referred to as manifold learning) is still inspired by the aim of multi-dimensional scaling; find a low-dimensional embedding of points such that their distances in the high-dimensional space are well represented. In contrast to Principal Component Analysis [15], one usually does not treat all distances equal. Rather, the idea is to preserve small distances well, while large distances are irrelevant, as long as they remain large; examples are Local Linear and (t-)Stochastic Neighbor Embedding [8, 19, 22]. Even that is typically not possible to achieve perfectly, and a trade-off between precision and recall arises [24]. Recent works are mostly spectral methods along this line.

Iterative data mining and machine learning.

There are two general frameworks for iterative data mining: FORSIED [3, 4] is based on modeling the belief state of the user as an evolving probability distribution in order to formalize subjective interestingness of patterns. This distribution is chosen as the Maximum Entropy distribution subject to the user beliefs as constraints, at that moment in time. Given a pattern syntax, one then aims to find the pattern that provides the most information, quantified as the 'subjective information content' of the pattern.

The other framework, which we here named CORAND [7, 13], is similar, but the evolving distribution does not necessarily have an explicit form. Instead, it relies on sampling, or put differently, on randomization of the data, given the user beliefs as constraints. Both these frameworks are *general* in the sense that it has been shown they can be applied in various data mining settings; local pattern mining, clustering, dimensionality reduction, etc.

The main difference is that in FORSIED, the background model is expressed analytically, while in CORAND it is defined implicitly. This leads to differences in how they are deployed and when they are effective. From a research and development perspective, randomization schemes are easier to propose, or at least they require little mathematical skills. Explicit models have the advantage that they often enable faster search of the best pattern, and the models may be more transparent. Also, randomization schemes are computationally demanding when many randomizations are required. Yet, in cases like the current paper, a single randomization suffices, and the approach scales very well. For both frameworks, it is ultimately the pattern syntax that determines their relative tractability.

Besides FORSIED and CORAND, many special-purpose methods have been developed for active learning, a form of iterative mining or learning, in diverse settings: classification, ranking, and more, as well as explicit models for user preferences. However, since these approaches are not targeted at data exploration, we do not review them here. Finally, several special-purpose methods have been developed for visual iterative data exploration in specific contexts, for example for itemset mining and subgroup discovery [1, 5, 23, 14], information retrieval [20], and network analysis [2].

Visually controllable data mining.

This work was motivated by and can be considered an instance of *visually controllable data mining* [17], where the objective is to implement advanced data analysis method so that they are understandable and efficiently controllable by the user. Our proposed method satisfies the properties of a visually controllable data mining method (see [17], Section II B): (VC1) the data and model space are presented visually, (VC2) there are intuitive visual interactions that allow the user to modify the model space, and (VC3) the method is fast enough to allow for visual interaction.

Information visualization and visual analytics.

Many new interactive visualization methods are presented yearly at the IEEE Conference on Visual Analytics Science and Technology (VAST). The focus in these communities is not on the use or development of advanced data mining or machine learning techniques, and more on human cognition and efficient use of displays, as well as efficient exploration via selection of data objects and features. Yet, the need to interact with the data mining community was already recognized long ago [11].

5. CONCLUSIONS

In order to improve the efficiency and efficacy of data exploration, there is a growing need for generic tools that integrate advanced visualization with data mining techniques to facilitate effective visual data analysis by human users. Our aim with this paper was to present a proof of concept for how this need can be addressed: a tool that initially presents the user with an 'interesting' projection of the data and then employs data randomization with constraints to allow users to flexibly express their interests or beliefs. These constraints expressed by the user are then taken into account by a projection-finding algorithm to compute a new 'interesting' projection, a process that can be iterated until the user runs out of time or finds that constraints explain everything the user needs to know about the data.

In our example, the user can associate two types of constraints on a chosen subset of data points: the appearance of the points in the particular projection or the fact that

the points can be nearby also in other projections. We also tested the tool on two data sets, one controlled experiment on synthetic data and another on real census data. We found that the tool performs according to our expectations; it manages to find interesting projections. Yet, interestingness can be case specific and relies on the definition of an appropriate interestingness measure, here the L_1 norm was employed. More research into this choice is warranted. Nonetheless, we think this approach is useful in constructing new tools and methods for interactive visually controllable data mining in variety of settings.

In further work we intend to investigate the use of the FORSIED framework to also formalize an analytical background model [3, 4], as well as its use for computing the most informative data projections. Additionally, alternative pattern syntaxes (constraints) will be investigated.

Acknowledgements.

This work was supported by the European Union through the ERC Consolidator Grant FORSIED (project reference 615517), Academy of Finland (decision 288814), and Tekes (Revolution of Knowledge Work project).

6. REFERENCES

[1] M. Boley, M. Mampaey, B. Kang, P. Tokmakov, and S. Wrobel. One click mining—interactive local pattern discovery through implicit preference and performance learning. In *Proc. of KDD IDEA*, pages 27–35, 2013.

[2] D. H. Chau, A. Kittur, J. I. Hong, and C. Faloutsos. Apolo: making sense of large network data by combining rich user interaction and machine learning. In *Proc. of CHI*, pages 167–176, 2011.

[3] T. De Bie. An information-theoretic framework for data mining. In *Proc. of KDD*, pages 564–572, 2011.

[4] T. De Bie. Subjective interestingness in exploratory data mining. In *Proc. of IDA*, pages 19–31, 2013.

[5] V. Dzyuba and M. van Leeuwen. Interactive discovery of interesting subgroup sets. In *Proc. of IDA*, pages 150–161, 2013.

[6] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Tr. Comp.*, 100(23):881–890, 1974.

[7] S. Hanhijärvi, M. Ojala, N. Vuokko, K. Puolamäki, N. Tatti, and H. Mannila. Tell me something I don't know: Randomization strategies for iterative data mining. In *Proc. of KDD*, pages 379–388, 2009.

[8] G. E. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *Proc. of NIPS*, pages 857–864, 2003.

[9] P. J. Huber. Projection pursuit. *Ann. Stat.*, 13(2):435–475, 1985.

[10] B. Kang, K. Puolamäki, J. Lijffijt, and T. De Bie. A tool for subjective and interactive visual data exploration. Under review.

[11] D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, editors. *Mastering the Information Age: Solving Problems with Visual Analytics*. Eurographics Association, 2010.

[12] J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29(2):115–129, 1964.

[13] J. Lijffijt, P. Papapetrou, and K. Puolamäki. A statistical significance testing approach to mining the most informative set of patterns. *DMKD*, 28(1):238–263, 2014.

[14] D. Paurat, R. Garnett, and T. Gärtner. Interactive exploration of larger pattern collections: A case study on a cocktail dataset. In *Proc. of KDD IDEA*, pages 98–106, 2014.

[15] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901.

[16] K. Puolamäki, B. Kang, J. Lijffijt, and T. De Bie. Interactive visual data exploration with subjective feedback. Under review.

[17] K. Puolamäki, P. Papapetrou, and J. Lijffijt. Visually controllable data mining methods. In *Proc. of ICDMW*, pages 409–417, 2010.

[18] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[19] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[20] T. Ruotsalo, G. Jacucci, P. Myllymäki, , and S. Kaski. Interactive intent modeling: Information discovery beyond search. *CACM*, 58(1):86–92, 2015.

[21] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[22] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 9(Nov):2579–2605, 2008.

[23] M. van Leeuwen and L. Cardinaels. Viper — visual pattern explorer. In *Proc. of ECML–PKDD*, pages 333–336, 2015.

[24] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *JMLR*, 11(Feb):451–490, 2010.

Interactive Constrained Boolean Matrix Factorization

Nelson Mukuze
Max-Planck-Institut für Informatik
Saarbrücken, Germany
nelson.mukuze@mpi-inf.mpg.de

Pauli Miettinen*
Max-Planck-Institut für Informatik
Saarbrücken, Germany
pauli.miettinen@mpi-inf.mpg.de

ABSTRACT

Boolean matrix factorization (BMF) has become one of the standard methods in data mining with applications to fields such as lifted inference, bioinformatics, and role mining, to name a few. But the standard formalization of BMF assumes all errors are equal, at most giving the user a chance to weigh different types of errors differently. In many cases, however—and here role mining is a good example—making errors at one element of the matrix can be inacceptable, while the value in another element might be rather inconsequential. It is therefore preferable that the user can express her constraints to the mining algorithm. Unfortunately, deciding on the constraints for every element of the matrix easily becomes infeasible. To solve that problem, we propose to query the constraints from the user only when they are needed. In this paper we demonstrate our system for interactive constrained BMF. We will present the problem and the algorithm, and in addition to the demonstration, we will also present a short experimental evaluation showing that our approach can find good factorizations in the presence of constraints.

CSS Concepts

•Human-centered computing → Interactive systems and tools; **•Information systems → Data mining**

Keywords

Boolean matrix factorization; interactive data mining; role mining

1. INTRODUCTION

In *role mining*, we are given a relation between a set of users and a set of rights telling us which user has which right. Such relation is naturally expressed as a binary matrix, and an example of such a matrix is presented in Figure 1. In Figure 1, we have three users, Alice, Bob, and Charles (A, B, and C, respectively) and three rights, print, delete, and execute (respectively p, d, and e), and Alice, for instance, has the right to print and execute, but not delete. The goal of role mining is to find a small collection of *roles*, that is, sets of rights, and corresponding collection of sets of users so that each user of set i has all the rights in role i.

$$\begin{array}{c} \\ p \\ e \\ d \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \end{array}$$

Figure 1: Example role mining dataset with three users, (A)lice, (B)ob, and (C)harles, and three rights, (p)rint, (e)xecute, and (d)elete

There are two common variants of the general role mining problem: either every user must get exactly the rights they held via the roles and the goal is to achieve this with the minimum number of roles, or we are given the maximum allowable number of roles, and the goal is minimize the errors we make (giving users new rights or taking existing ones away). If the data admits a description using a small sets of roles, the former variant is clearly more desirable (and indeed, many real-world data do admit it [2]). But other data could potentially take hundreds of roles to describe exactly (e.g. if the users are smartphone applications and the rights the permissions they ask [3]). In such cases we might be willing to give some users some new rights, and take some existing rights away, in an attempt to find a concise set of roles. Consider the example in Figure 1: to exactly express every user's right, we would need three roles, that is, one role for each user. But if we give Bob the 'execute' right, we can do with just two roles, namely $r_1 = \{p, e\}$ and $r_2 = \{e, d\}$, and Alice would have only role r_1, Charles would have only role r_2, and Bob would have both roles r_1 and r_2.

But a short moment of thought will immediately tell us that not every right is of equal importance to every user; indeed, there might be a very good reason why Bob is not granted the right to execute. If, instead, we want to describe the data with just two roles, we have to take away at least two rights from the users: we can, for instance, remove Bob's right to delete and Alice's right to execute, or we can remove Bob's right to print and Charles' right to execute. Both of these two cases will cause the same amount of error, but they might not be equal in other ways. Printing, for example, can be vital for Bob's job and he should not loose that right.

*Corresponding author

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16) August 14th, 2016, San Francisco, CA, USA.

The user can enter these constraints before she starts finding the roles and tell the algorithm which user–right pairs cannot be changed by giving the user the right or taking it away. But often there are too many constraints to consider: if the data has n users and m rights, there are nm user-right combinations, and each of them is a potential constraint. For example, in Section 5, we report results of our experiments using an Android application permission data that has 117 036 applications and 173 permissions. Considering all of the over twenty million application–permission pairs a priori is practically impossible.

Not only impossible, considering each of these pairs a priori is also often useless: frequently, the role mining algorithm will honor the vast majority of the constraints even without knowing about them. In order to allow the user to concentrate on those right assignments whose changing would help the algorithm to find a smaller set of roles, we propose an *interactive* approach, where the algorithm will inform the user whenever it is about to add a new or remove an existing right. The user can then decide whether the change can be done or whether this user–right pair should be left untouched. In the latter case, the algorithm will never try to change that pair again.

To formalize our problem, we cast it as an equivalent Boolean matrix factorization (BMF) problem, and indeed, our algorithm is designed for interactive constrained BMF. As BMF has many other applications areas besides role mining, so has our algorithm. Although we will use role mining as the motivating application throughout this paper, in Section 5, we will show that our algorithm can be used with many other datasets as well.

The main purpose of this paper is to demonstrate our algorithm, `iConFaRe` (Interactive Constrained Factor Reducer), and we will explain the proposed demonstration in Section 6. Let us first, however, formally define BMF in Section 2 and cover the related work in Section 3. After presenting our algorithm in Section 4, we present our experimental evaluation in Section 5 before concluding in Section 7.

2. NOTATION AND DEFINITIONS

As we explained, in order to have a general framework, we will describe our method in the terms of Boolean matrix factorization. We denote a matrix by upper-case boldface letters ($\boldsymbol{A}$), and vectors by lower-case boldface letters ($\boldsymbol{a}$). For a matrix $\boldsymbol{A}$ we denote its ith row by $\boldsymbol{A}_i$ and its jth column by $\boldsymbol{A}^j$.

We use the shorthand $[n]$ to denote the set of integers up to n, $[n] = \{1, 2, \ldots, n\}$.

Let $\boldsymbol{A} \in \{0,1\}^{n\times m}$, $\boldsymbol{B} \in \{0,1\}^{n\times k}$ and $\boldsymbol{C} \in \{0,1\}^{k\times m}$. We denote by $\boldsymbol{B} \circ \boldsymbol{C}$ the n-by-m *Boolean product* of matrices $\boldsymbol{B}$ and $\boldsymbol{C}$. The Boolean matrix product is defined like the normal product, but over the Boolean semiring, that is, $(\boldsymbol{B} \circ \boldsymbol{C})_{ij} = \bigvee_{\ell=1}^{k} \boldsymbol{B}_{i\ell}\boldsymbol{C}_{\ell j}$.

Let $\langle \boldsymbol{B}, \boldsymbol{C}\rangle$ be an (approximate) Boolean decomposition of $\boldsymbol{A}$, $\boldsymbol{A} \approx \boldsymbol{B} \circ \boldsymbol{C}$. We call $\boldsymbol{B}$ and $\boldsymbol{C}$ *factors* of this decomposition, and for any $1 \le l \le k$, we refer to the rank-1 matrix formed by the vector pair $\langle \boldsymbol{B}^l, \boldsymbol{C}_l\rangle$ as a *block*. If $\boldsymbol{X}$ and $\boldsymbol{Y}$ are n-by-m binary matrices, we use $\boldsymbol{X} \oplus \boldsymbol{Y}$ to denote their *element-wise exclusive or*. Finally, we denote by $|\boldsymbol{A}|$ the number of non-zeros in Boolean matrix $\boldsymbol{A}$, that is, $|\boldsymbol{A}| = \sum_{i,j} a_{ij}$.

The standard *Boolean matrix factorization* (BMF) problem is now:

Problem 1. Given $\boldsymbol{A} \in \{0,1\}^{n\times m}$ and $k \in \mathbb{N}$, find $\boldsymbol{B} \in \{0,1\}^{n\times k}$ and $\boldsymbol{C} \in \{0,1\}^{k\times m}$ that minimize

$$|\boldsymbol{A} - \boldsymbol{B} \circ \boldsymbol{C}| \ . \tag{1}$$

As we explained, the standard BMF considers all errors equal, but for example in the role mining application, this is not what is wanted. To address that problem, we can formulate the *constrained BMF* (cBMF) problem, where we are additionally given a set of index pairs denoting the locations of $\boldsymbol{A}$ where the factorization is not allowed to make mistakes:

Problem 2. Given $\boldsymbol{A} \in \{0,1\}^{n\times m}$, $k \in \mathbb{N}$, and a set of constraints $C = \{(i,j) : i \in [n], j \in [m]\}$, find $\boldsymbol{B} \in \{0,1\}^{n\times k}$ and $\boldsymbol{C} \in \{0,1\}^{k\times m}$ that minimize (1) while admitting the constraints, that is,

$$a_{ij} = (\boldsymbol{B} \circ \boldsymbol{C})_{ij} \quad \text{for all } (i,j) \in C. \tag{2}$$

Notice that our definition of cBMF has a significant problem: it is possible that there exists no valid solution. A simple example is if we let $\boldsymbol{A}$ to be the n-by-n identity matrix, set $C = [n] \times [n]$ (i.e. require exact decomposition), and set $k < n$. We can avoid this problem by requiring that the rank k is always high-enough, for example, by requiring that $k \ge \max\{|\{i \in [n] : (i,j) \in C\}|, |\{j \in [m] : (i,j) \in C\}|\}$; with this inequality, we know we can always represent the rows (or columns) with constraints exactly.

Another, and arguably more severe, problem of the above formulation is that it requires the user to pre-specify all constraints. Our approach is to make the algorithm query for the constraints when it needs to, and let the rank k be implicitly set: the algorithm tries to reduce the rank as much as possible without violating any constraints. To formalize the process of obtaining the constraints, consider a function $\mathcal{Q}_{\boldsymbol{A}} : [n]\times[m] \to \{0,1\}$. For every element (i,j) of $\boldsymbol{A}$, function $\mathcal{Q}_{\boldsymbol{A}}(i,j)$ returns 0 if the element is not constrained, and 1 if the element is constrained. We assume that our algorithm does not know the definition of $\mathcal{Q}_{\boldsymbol{A}}$, but it does have a way to evaluate it for any element (by asking the user). With these, we can define the problem we study in this paper, the *interactive constrained BMF* (icBMF):

Problem 3. Given $\boldsymbol{A} \in \{0,1\}^{n\times m}$ and a way to evaluate the function $\mathcal{Q}_{\boldsymbol{A}}$, find $\boldsymbol{B} \in \{0,1\}^{n\times k}$ and $\boldsymbol{C} \in \{0,1\}^{k\times m}$ such that k is minimized and the factorization $\boldsymbol{B} \circ \boldsymbol{C}$ does not violate any constraints, that is

$$\sum_{i=1}^{n}\sum_{j=1}^{m}(a_{ij} - (\boldsymbol{B} \circ \boldsymbol{C})_{ij})\mathcal{Q}_{\boldsymbol{A}}(i,j) = 0 \ . \tag{3}$$

Problem 3 does not consider the error at all. Obviously, it should be considered, but in the definition of Problem 3, it is implicit in the constraint query $\mathcal{Q}_{\boldsymbol{A}}$: if the user feels that there is going to be too much error, she can limit it by imposing more constraints. There are definitely other possible approaches, and we point the reader to Section 7 for more discussion on this topic.

3. RELATED WORK

Boolean matrix factorizations have received considerable research interest in data mining. The problem was introduced to the field in [11] together with the `Asso` algorithm. Subsequent papers have proposed new algorithms [1, 8], new

Algorithm 1 `iConFaRe`

Input: matrix $\boldsymbol{A} \in \{0,1\}^{n \times m}$, a way to evaluate function $\mathcal{Q}_{\boldsymbol{A}}$
Output: Factors $\boldsymbol{B} \in \{0,1\}^{n \times k}$ and $\boldsymbol{C} \in \{0,1\}^{k \times m}$

```
1: function iConFaRe(A, Q_A)
2:     ⟨B, C⟩ ← exact Boolean factorization of A
3:     repeat
4:         d ← the block ⟨B^d, C_d⟩ that causes the least error if deleted
5:         (m_1, m_2) ← the pair of blocks ⟨B^{m_1}, C_{m_1}⟩, ⟨B^{m_2}, C_{m_2}⟩ that cause the least error if merged
6:         op ← the delete or merge operation that causes the least error
7:         query Q_A if op violates any constraints
8:         if op does not violate any constraints then
9:             perform op to obtain new B and C
10:            mark all elements with errors unconstrained
11:        else
12:            mark op as inadmissible operation
13:            update the constraints
14:        end if
15:    until there are no admissible operations
16:    return B and C
17: end function
```

optimization goals [12], and new algorithms aiming to optimize these goals [5].

The special cases of exact Boolean matrix factorization (a.k.a. Boolean rank) and dominated Boolean matrix factorization were studied even earlier [4], although under the different name of *tiling*.

The role mining problem and its connections to BMF, were popularized in [15]. This approach was later extended in [6, 16], leading to constraint-aware role mining in [7]. Notice, however, that in [7], the constraints are something the algorithm is supposed to mine and express via *negative permissions*, while in our work, the user has to explicitly state the constraints. On the other hand, [2] provided an algorithm for computing the optimal Boolean matrix factorization (in exponential time) and applied it to many real-world user–right datasets. All of these approaches are based on combinatorial approaches; recently, [3] proposed a probabilistic approach.

4. OUR ALGORITHM

In this section we will present our approach for the icBMF problem. We divide our treatise into two parts, the back-end that is responsible for doing the computation, and the front-end that provides the user interface and in essence implements the evaluation of $\mathcal{Q}_{\boldsymbol{A}}$, although our front-end also allows for the user to steer the computation in other ways, as well.

Our algorithm, including the user interface, is implemented in Python and it is freely available from `http://people.mpi-inf.mpg.de/~pmiettin/bmf/interactive/`.

4.1 The Back-End

The general process of our algorithm, `iConFaRe`, is straightforward: it starts with an exact factorization and then reduces the factors, either by removing them, or by merging them, while making sure that it never violates any constraints. When it cannot anymore do any changes, it terminates. The pseudo-code of `iConFaRe` is presented in Algorithm 1.

Finding the exact decomposition.

We start `iConFaRe` by finding an exact Boolean matrix factorization of $\boldsymbol{A}$ in line 2. As this is an NP-hard problem, we use the heuristic minimum tiling algorithm of Geerts et al. [4]. That algorithm works essentially by first finding all closed itemsets of the data, and then solving the minimum set cover problem on an instance where each 1 of the data is an element, and each closed itemset is a set. For efficiency we use a slight modification of that idea, and instead of considering all closed itemsets, we allow the user to set a minimum-frequency threshold. To ensure that we can find an exact decomposition, we add all columns as closed itemsets, even if their frequency is below the user-set minimum-frequency threshold.

Instead of using the tiling algorithm, we could of course use any other algorithm returning an exact BMF; for smaller matrices, for instance, the method proposed by Ene et al. [2] can provide the optimal exact decomposition. We could also use non-exact decompositions, but then we would have to ensure that we do not violate any constraints as our algorithm is not guaranteed to fix errors that were committed during the exact factorization process.

Reducing the rank.

The exact decomposition usually has a rank that is too high for the application, and the main part of `iConFaRe` is to reduce that rank while ensuring it does not violate any constraints. It considers two ways of reducing the rank: delete a block, or merge two blocks (see below for how the blocks are selected for these operations). Both of these operations result in a new factorization that has one block less. To choose which of these operations it should perform, `iConFaRe` compares the errors the operations cause and selects the one that causes the least error. While we do not directly aim at minimizing the error, an operation that causes only little error is less likely to violate any constraint than an operation that causes a large error. Therefore, selecting the operation that causes the least error (lines 4–6) is a sensible heuristic.

The error is defined to be the number of elements where the operation would cause an error and that are not known to be constrained. If the operation would cause an error on any constrained element, it cannot be performed, and consequently, `iConFaRe` does not consider it. On the other hand, `iConFaRe` does not penalize operations for causing errors on known-unconstrained elements. The logic behind this is that if an element is known to be unconstrained, we can set it to whatever value we want. Further, `iConFaRe` will only consider elements to be unconstrained if it has already committed errors on those elements, and ignoring those elements avoids double-counting the errors.

Before `iConFaRe` commits the operation, it evaluates $\mathcal{Q}_{\boldsymbol{A}}$ (i.e. queries the user) to guarantee that it is not violating any constraints. This querying needs to be done for all new errors committed by the operation (the user has already allowed the old errors, so those elements cannot be constrained). If the operation does not violate any constraints, `iConFaRe` commits it in line 9. It also marks all elements where the new error is committed as unconstrained.

If the operation violates some constraints, `iConFaRe` marks it as inadmissible (to avoid considering it again) and updates the information regarding the constrained elements (in practise, though, the latter information might not be available; see Section 4.2).

The main loop, and the algorithm, end when `iConFaRe` cannot anymore find any operations that would not violate

some constraints. At that point it simply returns the current factorization.

The delete and merge operations.

The operation of deleting a block is straightforward, and so is finding the block to delete. For every block in the factorization, `iConFaRe` keeps a list of elements that are 1 only in this block. If that element is 1 also in the data, we know that removing this block would also commit an error of not covering the 1 (e.g. removing a right from a user). Deleting a block, obviously, cannot ever add 1s in the factorization. An inverted index, matching every 1 in the data to the blocks in the factorization, can be used to efficiently update the information whether a block is the only one covering a 1.

The merge operation is more complicated, and in fact we consider two different merge operations: *and-merge* and *or-merge*. Consider two blocks

$$\boldsymbol{B}^1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \qquad \boldsymbol{C}^1 = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{B}^2 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad \boldsymbol{C}^2 = \begin{pmatrix} 1 & 0 & 1 & 0 \end{pmatrix}$$

The *and-merge* of the two blocks would result to factors $\boldsymbol{B}^{\text{and}} = (1000)^T$ and $\boldsymbol{C}^{\text{and}} = (1000)^T$, while the *or-merge* would result to factors $\boldsymbol{B}^{\text{or}} = (1110)^T$ and $\boldsymbol{C}^{\text{or}} = (1110)^T$. Notice that the *and-merge* can only remove 1s from the factorization, while the *or-merge* can only add them.

Finding good blocks to merge is harder than with deletions as it is hard to know how much error each operation would cause without first computing all pair-wise merges (or both types). That, naturally, is infeasible. Rather, we try to find two rows of $\boldsymbol{C}$ that have a high Jaccard similarity, that is, the value $J(\boldsymbol{x}, \boldsymbol{y}) = |\boldsymbol{x} \wedge \boldsymbol{y}| \,/\, |\boldsymbol{x} \vee \boldsymbol{y}|$ is high. For such vectors, both types of merges should yield small errors as the vectors are already rather similar and hence we will only consider the pair of vectors with the highest similarity for merging.

Finding pairs of vectors with high Jaccard similarity will still require us to consider all $\binom{k}{2}$ pairs of rows of $\boldsymbol{C}$ for every run of the loop. As `iConFaRe` is supposed to be an interactive algorithm, we might not be able to wait so long. To speed up the processing, we use the minhash signatures [14, Ch. 3]: we use the min-wise hashing (or *minhashing*) to compute a short signature for each row of $\boldsymbol{C}$, and the similarity between these signatures gives us a good approximation of the Jaccard similarity of the rows.

After `iConFaRe` has selected the pair of blocks for merging, it simply checks which of *and-merge* or *or-merge* yields the smallest error and recommends it to the user.

4.2 The Front-End

The back-end of `iConFaRe` is essentially non-interactive, and all of the interaction happens during the evaluation of the function $\mathcal{Q}_{\boldsymbol{A}}$: this evaluation is done by the front-end that queries the user whether the recommended operation is admissible. The front-end will then inform the back-end whether it can commit the operation or not. Computing the exact decomposition is not done in an interactive way, and as it can be time-consuming, we have implemented it as a separate pre-processing step.

The front-end allows for other types of interaction than just verifying the recommended operations. In particular, it lets the user mark full rows or columns restricted and remove and merge blocks herself, even if `iConFaRe` did not recommend those operations. Our goal here is to allow the user to use her domain knowledge and semantic understanding of the data to perform operations that might seem sub-optimal for the back-end, but that the user knows are admissible in the domain. For example, the user might know that certain users have a right to use machinery that is decommissioned. Removing that right, then, will not harm the user's capability to carry on their duties, but it can help `iConFaRe` to find more concise set of roles. On the other hand, being able to mark the whole column (or row) restricted will reduce the number of unnecessarily proposed deletes and merges. This can be useful if some rights are considered very sensitive.

The main user interface.

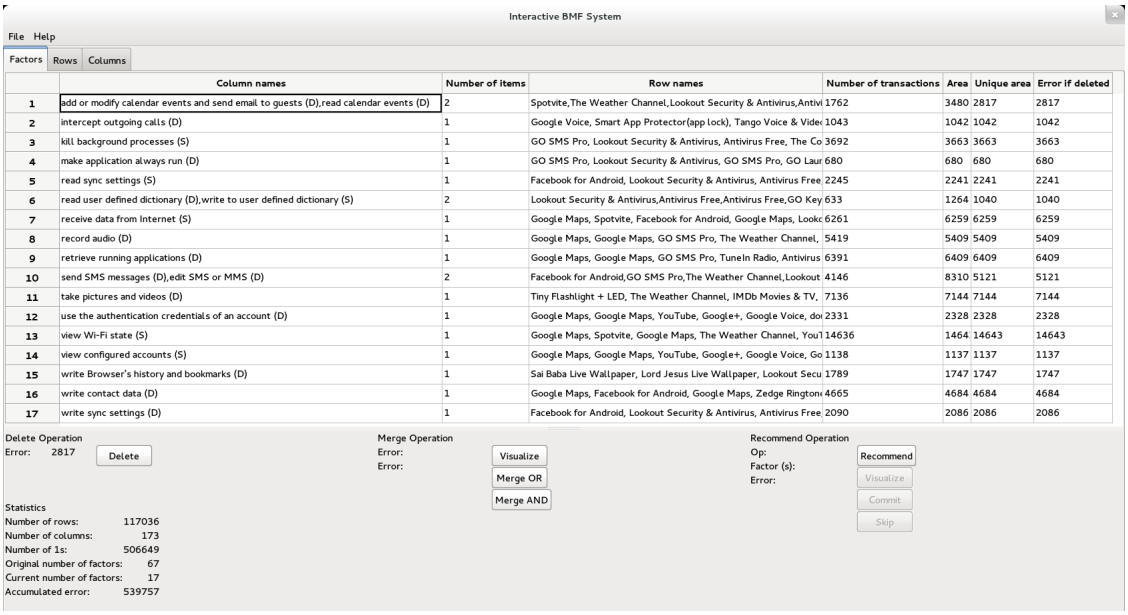
The main user interface of `iConFaRe`'s front-end consists of a list view showing the current blocks (see Figure 2). This view allows the user to see all of the factors, sort them using different criteria, and manually perform merge and delete operations. It also allows the user to interact with the back-end, asking it to produce the next recommended operation (i.e. run one iteration of the main loop until the next evaluation of the constraints). The next tabs of the main view allow the user to see more information on the rows and columns in the data, and specify entire rows and columns as constraints in a fashion similar to constrained clustering.

A simple but powerful application of the main list view is to quickly delete a number of blocks. In particular, the exact initial decomposition can yield an excessive number of very small blocks (covering only few, or just one, rows and columns). In many applications, they can be deleted with only minimal consideration. In the main view, the user can easily sort the blocks based on their size and quickly delete all small blocks from the list.

In addition to the list view, `iConFaRe` has also a persistent global view of the data (Figure 3). This view shows the effects of operations on the full data. The main information this visualization conveys is the effects of the operations on the current factorization. For example, if we want to delete a block, we can see which 1s in the current factorization would turn into 0s (e.g. which rights would be removed from which users), which 1s would be covered by only one block (removal of which would remove the 1s from the factorization), and which other 1s this block covers (together with at least two other blocks). Further information is shown for merge operations. This allows the user to do long-term planning beyond that of `iConFaRe`: if an operation is going to make many 1s dependable on one other block only, the user might cancel the operation to allow for deleting other factors in the future.

Especially with big datasets, the global view might not be detailed enough and the effects scattered around the data can be hard to interpret. For that purpose, `iConFaRe` also includes a local view (Figure 4) that shows only the elements that the operation is going to affect.

Figure 4 shows a case of *or-merge* where few elements would turn into 1 in the factorization (denoted using blue



	Column names	Number of items	Row names	Number of transactions	Area	Unique area	Error if deleted
1	add or modify calendar events and send email to guests (D),read calendar events (D)	2	Spotvite,The Weather Channel,Lookout Security & Antivirus,Antiv	1762	3480	2817	2817
2	intercept outgoing calls (D)	1	Google Voice, Smart App Protector(app lock), Tango Voice & Vide	1043	1042	1042	1042
3	kill background processes (S)	1	GO SMS Pro, Lookout Security & Antivirus, Antivirus Free, The Co	3692	3663	3663	3663
4	make application always run (D)	1	GO SMS Pro, Lookout Security & Antivirus, GO SMS Pro, GO Laur	680	680	680	680
5	read sync settings (S)	1	Facebook for Android, Lookout Security & Antivirus, Antivirus Free	2245	2241	2241	2241
6	read user defined dictionary (D),write to user defined dictionary (S)	2	Lookout Security & Antivirus,Antivirus Free,Antivirus Free,GO Key	633	1264	1040	1040
7	receive data from Internet (S)	1	Google Maps, Spotvite, Facebook for Android, Google Maps, Lookc	6261	6259	6259	6259
8	record audio (D)	1	Google Maps, Google Maps, GO SMS Pro, The Weather Channel,	5419	5409	5409	5409
9	retrieve running applications (D)	1	Google Maps, Google Maps, GO SMS Pro, TuneIn Radio, Antivirus	6391	6409	6409	6409
10	send SMS messages (D),edit SMS or MMS (D)	2	Facebook for Android,GO SMS Pro,The Weather Channel,Lookout	4146	8310	5121	5121
11	take pictures and videos (D)	1	Tiny Flashlight + LED, The Weather Channel, IMDb Movies & TV,	7136	7144	7144	7144
12	use the authentication credentials of an account (D)	1	Google Maps, Google Maps, YouTube, Google+, Google Voice, do	2331	2328	2328	2328
13	view Wi-Fi state (S)	1	Google Maps, Spotvite, Google Maps, The Weather Channel, YouT	14636	1464	14643	14643
14	view configured accounts (S)	1	Google Maps, Google Maps, YouTube, Google+, Google Voice, Go	1138	1137	1137	1137
15	write Browser's history and bookmarks (D)	1	Sai Baba Live Wallpaper, Lord Jesus Live Wallpaper, Lookout Secu	1789	1747	1747	1747
16	write contact data (D)	1	Google Maps, Facebook for Android, Google Maps, Zedge Ringtone	4665	4684	4684	4684
17	write sync settings (D)	1	Facebook for Android, Lookout Security & Antivirus, Antivirus Free	2090	2086	2086	2086

Figure 2: The main user interface of `iConFaRe`. The list shows the current factors with associated statistics, and the space at the bottom-left shows detailed statistics for the selected factor. At the middle, the buttons allow the user to commit a delete or merge operation, with relevant statistics shown next to the button, or to ask the `iConFaRe` to propose the next operation. This view shows factors from the AndroidApps data.

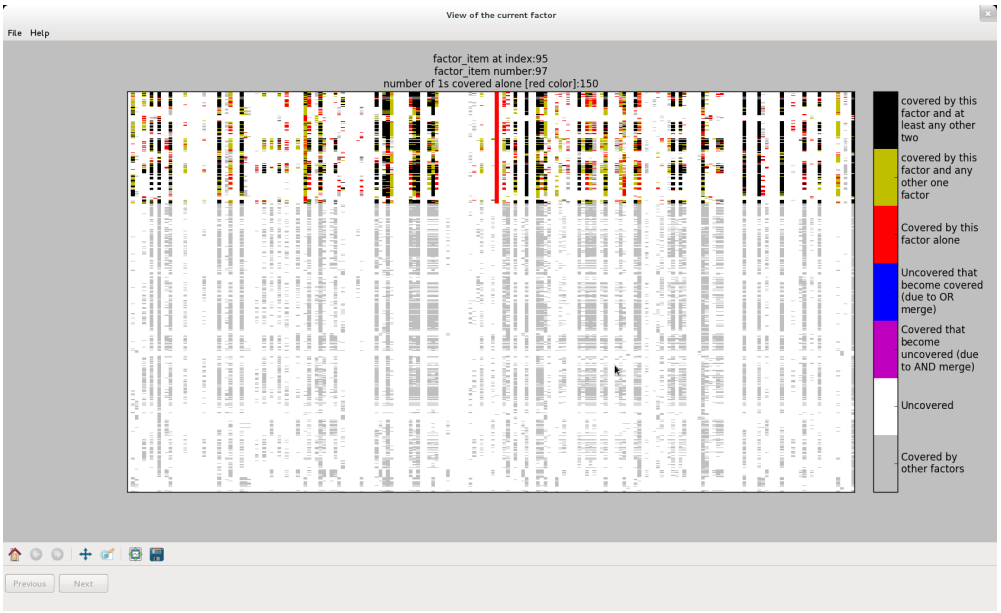


Figure 3: Visualization of the full data. One factor is selected, and the visualization shows how it effects the current factorization. For example, deleting this factor would mean that all the red dots in the data would turn to 0 in the factorization. The dataset shown is the Mammals data.

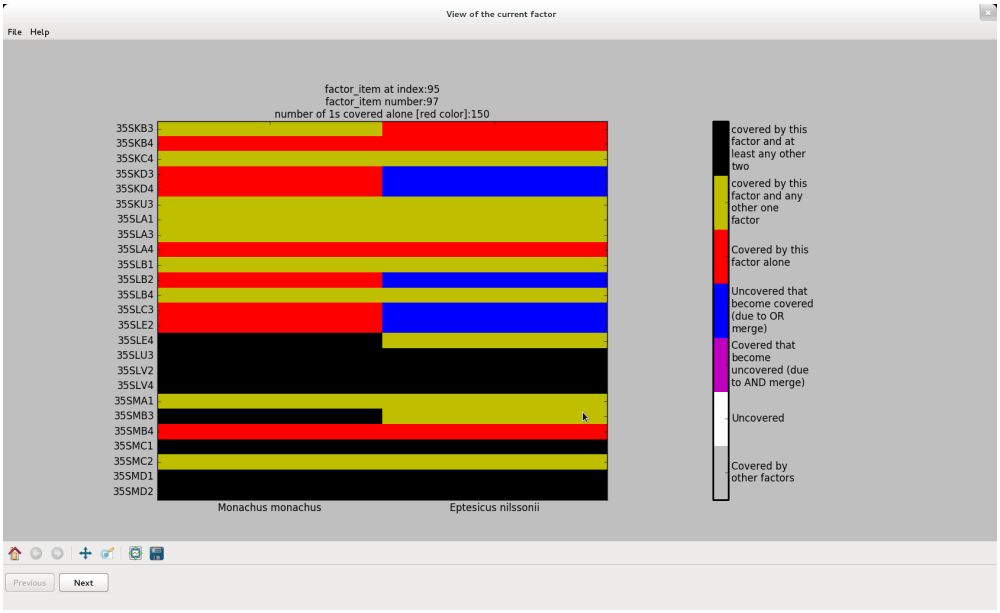


Figure 4: Visualization of the local view. Two factors are to be merged with *or-merge*, and the visualization shows how it effects the current factorization. The colors are the same as in Figure 3. The dataset shown is the Mammals data.

color). As the overall area of the blue color is rather small, the user might as well decide to merge these two blocks unless she knows that *Eptesicus nilssonii* (a type of a bat) should never appear in the areas corresponding to the rows, for example, in square 35SKD3 (latitude 39.26, longitude 24.00, close to the East coast of Greece); indeed, *E. nilssonii* is not supposed to live so south, and the user might well reject this block for being unintuitive.

Communicating with the back-end.

When the user clicks the "Recommend" button at the main view, `iConFaRe` computes the next recommended operation that is then shown to the user. If the user commits the operation, `iConFaRe` knows that all errors committed are allowed, and consequently marks all those elements to be unconstrained. If, on the other hand, the user skips the operation, `iConFaRe` does not know what was the reason: not all errors might have been in the constrained elements, or the user might have decided to skip the recommended operation for other reasons. Hence, by default, `iConFaRe` will not mark any elements constrained even if the user does not allow an operation. However, `iConFaRe` does not recommend the same operation again as the user skipped it before, but recommends the user the next operation resulting in the smallest error.

To mark elements as constrained, the user must take specific action. In particular, she has to select the row on which the element is, and left-click the element on the local view to constrain it. This extra step is needed as otherwise `iConFaRe` cannot infer the constraints. Also, in many situations the user might elect to not give the constraints explicitly the first time they are violated. It might be simplest to just skip the recommended operation, and only mark it as a constraint if `iConFaRe` repeatedly recommends violating it.

To mark full rows or columns constrained, the user can use the two other tabs of `iConFaRe`'s main interface. These tabs show information regarding all the rows and columns of the data, and also allow the user to mark them constrained.

5. EXPERIMENTAL EVALUATIONS

While the main purpose of this manuscript is to demonstrate the interactive `iConFaRe` system, we have also run some off-line experiments to validate our approach.

5.1 Real-World Datasets

We have tested `iConFaRe` with three real-world datasets. We summarize the characteristics of the datasets in Table 1.

The first is the DBLP dataset. It contains the names of 6 980 authors and which of the 19 conferences they have published to. The dataset was collected from the DBLP database[1], and is pre-processed as in [9]. It is the first dataset on which `iConFaRe` was tested to see its performance as a system for constrained Boolean matrix factorization given user's constraints, because it is sparse and has less features (only 19 columns, corresponding to the 19 conferences).

The second real-world dataset that `iConFaRe` was tested on is the Mammals dataset. It consists of presence–absence data[2] of European mammals with geographical areas of 50-by-50 kilometers [13]. This dataset is denser than the DBLP dataset and has far more features (194 columns).

The third real-world dataset that we used is the AndroidApps dataset[3] [3]. For each application, the dataset provides the permissions requested by the application, the price, the number of downloads, the average user rating, and a short prosaic description. The original data was pre-

[1] `http://www.informatik.uni-trier.de/~ley/db/`
[2] Available for research purposes from the Societas Europaea Mammalogica at `http://www.european-mammals.org`
[3] `http://www.mariofrank.net/andrApps/`

Dataset	Rows	Columns	Density (%)
DBLP	6 980	19	13
Mammals	2 618	194	16
AndroidApps	117 036	173	2

Table 1: Real world datasets overview.

Dataset	Rank
DBLP	11
Mammals	129
AndroidApps	67

Table 2: Factorization ranks returned by `iConFaRe` with real-world data.

processed by removing all applications with ratings and number of downloads less than the average rating and number of downloads. We also removed all applications which request no permission at all during the pre-processing step. Additionally, all columns which are not permissions (the price, the number of downloads, the average user rating, and a short prosaic description) were removed from the data. After the pre-processing step, the dataset contained 117 036 android applications and the presence (1) or absence (0) of 173 permission. This dataset is a good use-case for the role mining application of `iConFaRe`.

5.2 Algorithms Used

To test `iConFaRe` in a controlled manner, we used it in a non-interactive way. Namely, we sampled random constraints for the data and replaced the evaluation of $\mathcal{Q}_{\boldsymbol{A}}$ with a function that checked whether the proposed option would violate the constraints or not.

To the best of our knowledge, `iConFaRe` is the first algorithm for icBMF (or cBMF). To compare it against other algorithms, we took `Asso` [11], a popular algorithm for standard BMF, and edited it to accept pre-defined constraints. This edit was done essentially by adjusting the evaluation function of `Asso` to make any factorization that would violate the constraints infinitely bad. Notice however, that as `Asso` builds the factorization from bottom up (i.e. it starts with an empty factorization), it cannot guarantee that the final factorization does cover all constrained 1s in the data.

5.3 Results

To test these two algorithms, we computed the cBMF factorizations with both algorithms using the same sets of random constraints. As `Asso` requires the rank as an input, we could not compare the methods based on the rank they returned. Instead, we first ran `iConFaRe` with the given datasets to obtain the error it gave and the rank it returned (`iConFaRe` was reducing the rank until it could not find any admissible operations). The ranks proposed by `iConFaRe` are listed in Table 2. We want to emphasize that these ranks do not denote any kind of "latent rank" of the data (see, e.g. [5,12]) as they depend heavily on the constraints we have randomly set.

After we got the ranks, we ran the modified `Asso` with the same ranks and constraints, and recorded the error. Errors for both `iConFaRe` and `Asso` are reported in Table 3. In DBLP, the modified `Asso` algorithm obtains smaller reconstruction

Dataset	`iConFaRe`	`Asso`
DBLP	78.3148	**68.3515**
Mammals	**44.7505**	48.5783
AndroidApps	**1.3058**	3.7815

Table 3: Errors in percentages of 1s in the data for `iConFaRe` and a constrained version of `Asso` on real-world datasets.

error, but in the other datasets, Mammals and AndroidApps, `iConFaRe` is actually better.

The error especially on DBLP, and arguably also on Mammals, is high with both systems. This is probably due to the constraints that we imposed as the standard version of `Asso` is known to perform relatively well on both of these datasets. Hence, in this experiment the actual error is much less important than the error relative to modified `Asso`. In this respect `iConFaRe` performs very well especially as, unlike `Asso`, it actually guarantees to admit all constraints.

In our final experiment we tested the effects of the different constraint sets. For this experiment, we generated five different random sets of constraints and ran both `iConFaRe` and the modified `Asso` on all of them. The average error of `iConFaRe` was 1.31% (with standard deviation of ± 0.01), while for `Asso` the average error was 6.62 (± 3.28), showing that not only is `iConFaRe` significantly better than `Asso`, but also more consistent.

6. THE DEMONSTRATION

`iConFaRe` is inherently interactive, and hence best demonstrated in a way that allows the audience to have hands-on time with it. On the other hand, the initial pre-processing step can be time-consuming, and consequently unsuitable for demonstrations. Hence we have to limit the demonstration to the pre-processed datasets, for which we are planning to use the three datasets used in the experiments.

At the beginning of the demonstration, the audience is explained the goal of constrained BMF and the general ideas behind `iConFaRe`. They can then choose one of the pre-processed datasets based on their interests. For the purpose of the demonstration, we plan to use the AndroidApps dataset, as most audience members are expected to have at least a passing familiarity with smartphone application permissions. Hence, barring special request from the audience, we will use it. The demonstrator will then walk them through the basic functionality of `iConFaRe` by means of example merges and deletions. This step also explains the visualizations and their interpretations. The audience could then come up with constraints as they see fit, and test if `iConFaRe` indeed recommends operations which do not violate any of those constraints.

After the user has become used to the system—and provided that they are interested—we can load a new dataset (or re-load the AndroidApps data) and set the user a task: come up with the least-error decomposition of given rank (to be defined later) and admitting some constraints (also to be defined). The goal of this experiment is two-fold: for one, it should give the user better understanding of `iConFaRe`, but it should also give us important data on how well `iConFaRe` performs in these situations. Furthermore, it is interesting to see whether the humans with their semantic understanding

and pattern recognition skills can perform better than the automated iConFaRe setup.

7. CONCLUSIONS

In this paper we have presented our system called iConFaRe for interactive constrained Boolean matrix factorization. In some sense iConFaRe presents a first-order system: after starting with an exact decomposition, it always considers only one step ahead. Hence, it never tries to add any new blocks but only to merge or delete the existing ones—adding a new block will never be an optimal move alone. On the other hand, a higher-order system would consider multiple steps ahead, and it could consider adding new blocks if they would allow it to remove many existing blocks in the subsequent steps. Such higher-order systems are, however, significantly more complicated, and it is also unclear how to present their operations to the user.

As of now, iConFaRe attempts only to minimize the rank. This was partially done to avoid the problematic bi-optimization criterium that tries to balance the rank and the error. This criterium is problematic as it requires us to decide the relative weights between reducing the rank and increasing the error. One way to do that, though, would be to use the Minimum Description Length (MDL) principle. Using the MDL principle for BMF was pioneered in [12], and algorithms optimizing MDL directly have been proposed in recent years [5, 8].

Systems similar to iConFaRe could use the MDL principle to choose which operation to perform and when to stop. We argue, however, that in the iConFaRe system this would be unlikely to provide much, or any, benefits. For one, being able to infer the rank is a less important problem in iConFaRe as the user is able to stop the algorithm when she feels it has obtained small-enough decomposition. The selection of the next operation, on the other hand, would probably not see much changes: in a first-order system like iConFaRe, the MDL-optimal way to reduce the rank by 1 is often to do that in a way that minimizes the increase in the error. Here a higher-order system could behave differently.

Another aspect omitted from iConFaRe's formal problem statement is the amount of user-involvement required. In practical terms, an interactive system like iConFaRe should aim to minimize the cases the user needs to consider. This is not explicitly stated in the definition of Problem 3, but we have designed iConFaRe to follow these guidelines as much as possible. In particular, the goal to minimize the error the operation causes implies that the user needs to consider the least number of elements for constraints.

The goal of minimizing the user involvement could be formalized in a *budgeted problem*, where the system has a fixed budged B and each query of $\mathcal{Q}_{\boldsymbol{A}}$ reduces it. This budged-based approach could allow for more principled approaches on selecting which operation to perform. In particular, when in iConFaRe we select the operation that tries to minimize the amount of error, a potential approach to the budgeted approach could be to first query the status of few elements that, if unrestricted, would let the algorithm reduce the number of factors the most. We leave the study of the budgeted version and its algorithms as a future work.

Another very interesting direction of future work would involve the user as a helper for the algorithm. Particularly, we could approach the unconstrained Boolean matrix factorization as a human-assisted data mining problem where the algorithm could ask from the user which operation she would think would be the best. The greedy algorithms often involved in BMF-style problems try to make locally optimal decisions, but these can lead into globally very sub-optimal outcomes. It could be that by involving the human in the decision-making process (in a limited manner), the overall quality of the results would improve.

We believe that iConFaRe is capable of providing practical benefits to real-world applications of BMF beyond our poster child application of role mining. Being able to tell the algorithm to avoid non-intuitive factors before they are being created, but without having to pre-specify what "non-intuitive" means, can greatly improve the usability of the results to the end-user. On the other hand, such great powers come with great responsibly, and there is a real risk that when iConFaRe (and similar tools) are applied to general data mining, the user inadvertently guides the system to find only results she knew a priori [10]. Designing checks to prevent such outcomes is an interesting direction of future work.

8. REFERENCES

[1] R. Bělohlávek and V. Vychodil. Discovery of optimal factors in binary data via a novel method of matrix decomposition. *J. Comput. Syst. Sci.*, 76(1):3–20, 2010.

[2] A. Ene, W. Horne, N. Milosavljevic, P. Rao, R. Schreiber, and R. E. Tarjan. Fast exact and heuristic methods for role minimization problems. In *SACMAT '08*, pages 1–10. ACM Press, 2008.

[3] M. Frank, B. Dong, A. Porter Felt, and D. Song. Mining Permission Request Patterns from Android and Facebook Applications. In *ICDM '12*, pages 870–875, 2012.

[4] F. Geerts, B. Goethals, and T. Mielikäinen. Tiling databases. In *DS '04*, pages 278–289, 2004.

[5] S. Karaev, P. Miettinen, and J. Vreeken. Getting to Know the Unknown Unknowns: Destructive-Noise Resistant Boolean Matrix Factorization. In *SDM '15*, pages 325–333, 2015.

[6] H. Lu, J. Vaidya, and V. Atluri. Optimal Boolean Matrix Decomposition: Application to Role Engineering. In *ICDE '08*, pages 297–306, 2008.

[7] H. Lu, J. Vaidya, V. Atluri, and Y. Hong. Constraint-Aware Role Mining Via Extended Boolean Matrix Decomposition. *IEEE Trans. Depend. Secure*, 9(5):655–669, 2012.

[8] C. Lucchese, S. Orlando, and R. Perego. A Unifying Framework for Mining Approximate Top-k Binary Patterns. *IEEE Trans. Knowl. Data Eng.*, 26(12):2900–2913, Dec. 2013.

[9] P. Miettinen. *Matrix Decomposition Methods for Data Mining: Computational Complexity and Algorithms.* PhD thesis, Department of Computer Science, University of Helsinki, 2009.

[10] P. Miettinen. Interactive Data Mining Considered Harmful (If Done Wrong). In *IDEA '14*, pages 85–87, July 2014.

[11] P. Miettinen, T. Mielikäinen, A. Gionis, G. Das, and H. Mannila. The Discrete Basis Problem. *IEEE Trans. Knowl. Data Eng.*, 20(10):1348–1362, Oct. 2008.

[12] P. Miettinen and J. Vreeken. MDL4BMF: Minimum Description Length for Boolean Matrix Factorization. *ACM Trans. Knowl. Discov. Data*, 8(4), Oct. 2014.

[13] A. J. Mitchell-Jones, G. Amori, W. Bogdanowicz, B. Krystufek, P. Reijnders, F. Spitzenberger, M. Stubbe, J. Thissen, V. Vohralik, and J. Zima. *The atlas of European mammals*. Poyser, 1999.

[14] A. Rajaraman, J. Leskovec, and J. D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, 1.3 edition, July 2013.

[15] J. Vaidya, V. Atluri, and Q. Guo. The role mining problem: finding a minimal descriptive set of roles. In *SACMAT '07*, pages 175–184, 2007.

[16] J. Vaidya, V. Atluri, Q. Guo, and H. Lu. Edge-RMP: Minimizing administrative assignments for role-based access control. *J. Comp. Secur.*, 17(2):211–235, 2009.

"Why Should I Trust You?" Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

ABSTRACT

Despite widespread adoption, machine learning models remain mostly black boxes. Understanding the reasons behind predictions is, however, quite important in assessing *trust*, which is fundamental if one plans to take action based on a prediction, or when choosing whether to deploy a new model. Such understanding also provides insights into the model, which can be used to transform an untrustworthy model or prediction into a trustworthy one.

In this work, we propose LIME, a novel explanation technique that explains the predictions of *any* classifier in an interpretable and faithful manner, by learning an interpretable model locally around the prediction. We also propose a method to explain models by presenting representative individual predictions and their explanations in a non-redundant way, framing the task as a submodular optimization problem. We demonstrate the flexibility of these methods by explaining different models for text (e.g. random forests) and image classification (e.g. neural networks). We show the utility of explanations via novel experiments, both simulated and with human subjects, on various scenarios that require trust: deciding if one should trust a prediction, choosing between models, improving an untrustworthy classifier, and identifying why a classifier should not be trusted.

This work will be presented at the main conference of KDD.

1. INTRODUCTION

Machine learning is at the core of many recent advances in science and technology. Unfortunately, the important role of humans is an oft-overlooked aspect in the field. Whether humans are directly using machine learning classifiers as tools, or are deploying models within other products, a vital concern remains: *if the users do not trust a model or a prediction, they will not use it.* It is important to differentiate between two different (but related) definitions of trust: (1) *trusting a prediction*, i.e. whether a user trusts an individual prediction sufficiently to take some action based on it, and (2) *trusting a model*, i.e. whether the user trusts a model to behave in reasonable ways if deployed. Both are directly impacted by how much the human understands a model's behaviour, as opposed to seeing it as a black box.

Determining trust in individual predictions is an important problem when the model is used for decision making. When using machine learning for medical diagnosis [6] or terrorism detection, for example, predictions cannot be acted upon on blind faith, as the consequences may be catastrophic.

Apart from trusting individual predictions, there is also a need to evaluate the model as a whole before deploying it "in the wild". To make this decision, users need to be confident that the model will perform well on real-world data, according to the metrics of interest. Currently, models are evaluated using accuracy metrics on an available validation dataset. However, real-world data is often significantly different, and further, the evaluation metric may not be indicative of the product's goal. Inspecting individual predictions and their explanations is a worthwhile solution, in addition to such metrics. In this case, it is important to aid users by suggesting which instances to inspect, especially for large datasets.

In this paper, we propose providing explanations for individual predictions as a solution to the "trusting a prediction" problem, and selecting multiple such predictions (and explanations) as a solution to the "trusting the model" problem. Our main contributions are summarized as follows.

- LIME, an algorithm that can explain the predictions of *any* classifier or regressor in a faithful way, by approximating it locally with an interpretable model.
- SP-LIME, a method that selects a set of representative instances with explanations to address the "trusting the model" problem, via submodular optimization.
- Comprehensive evaluation with simulated and human subjects, where we measure the impact of explanations on trust and associated tasks. In our experiments, non-experts using LIME are able to pick which classifier from a pair generalizes better in the real world. Further, they are able to greatly improve an untrustworthy classifier trained on 20 newsgroups, by doing feature engineering using LIME. We also show how understanding the predictions of a neural network on images helps practitioners know when and why they should not trust a model.

2. THE CASE FOR EXPLANATIONS

By "explaining a prediction", we mean presenting textual or visual artifacts that provide qualitative understanding of the

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16) August 14th, 2016, San Francisco, CA, USA.

ACM ISBN .

DOI:

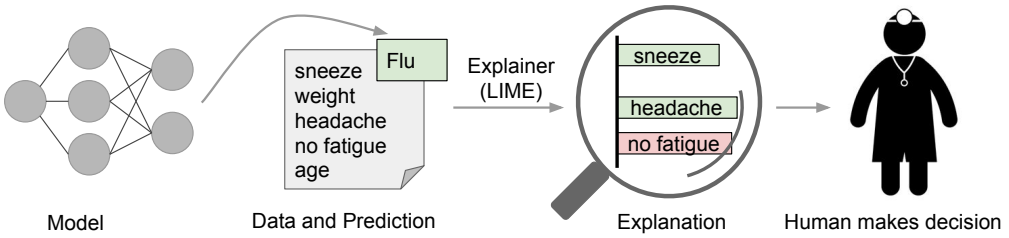


Figure 1: Explaining individual predictions. A model predicts that a patient has the flu, and LIME highlights the symptoms in the patient's history that led to the prediction. Sneeze and headache are portrayed as contributing to the "flu" prediction, while "no fatigue" is evidence against it. With these, a doctor can make an informed decision about whether to trust the model's prediction.

relationship between an instance's components (e.g. words in text, patches in an image) and the model's prediction. We argue that explaining predictions is an important aspect in getting humans to trust and use machine learning effectively, if the explanations are faithful and intelligible.

The process of explaining individual predictions is illustrated in Figure 1. It is clear that a doctor is much better positioned to make a decision with the help of a model if intelligible explanations are provided. In this case, an explanation is a small list of symptoms with relative weights – symptoms that either contribute to the prediction (in green) or are evidence against it (in red). Humans usually have prior knowledge about the application domain, which they can use to accept (trust) or reject a prediction if they understand the reasoning behind it. It has been observed, for example, that providing explanations can increase the acceptance of movie recommendations [12] and other automated systems [8].

Every machine learning application also requires a certain measure of overall trust in the model. Development and evaluation of a classification model often consists of collecting annotated data, of which a held-out subset is used for automated evaluation. Although this is a useful pipeline for many applications, evaluation on validation data may not correspond to performance "in the wild", as practitioners often overestimate the accuracy of their models [21], and thus trust cannot rely solely on it. Looking at examples offers an alternative method to assess truth in the model, especially if the examples are explained. We thus propose explaining several representative individual predictions of a model as a way to provide a global understanding.

There are several ways a model or its evaluation can go wrong. Data leakage, for example, defined as the unintentional leakage of signal into the training (and validation) data that would not appear when deployed [14], potentially increases accuracy. A challenging example cited by **(author?)** [14] is one where the patient ID was found to be heavily correlated with the target class in the training and validation data. This issue would be incredibly challenging to identify just by observing the predictions and the raw data, but much easier if explanations such as the one in Figure 1 are provided, as patient ID would be listed as an explanation for predictions. Another particularly hard to detect problem is dataset shift [5], where training data is different than test data (we give an example in the famous 20 newsgroups dataset later on). The insights given by explanations are particularly helpful in identifying what must be done to convert an untrustworthy model into a trustworthy one – for example, removing leaked data or changing the training data to avoid dataset shift.

Machine learning practitioners often have to select a model

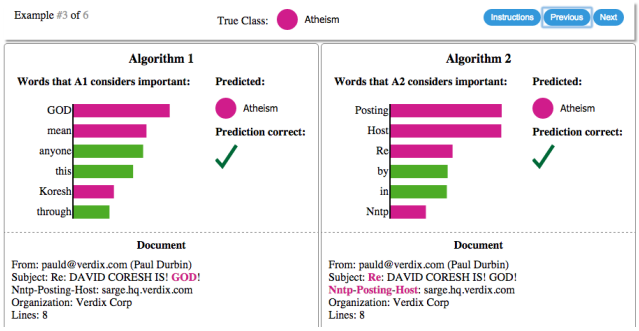


Figure 2: Explaining individual predictions of competing classifiers trying to determine if a document is about "Christianity" or "Atheism". The bar chart represents the importance given to the most relevant words, also highlighted in the text. Color indicates which class the word contributes to (green for "Christianity", magenta for "Atheism").

from a number of alternatives, requiring them to assess the relative trust between two or more models. In Figure 2, we show how individual prediction explanations can be used to select between models, in conjunction with accuracy. In this case, the algorithm with higher accuracy on the validation set is actually much worse, a fact that is easy to see when explanations are provided (again, due to human prior knowledge), but hard otherwise. Further, there is frequently a mismatch between the metrics that we can compute and optimize (e.g. accuracy) and the actual metrics of interest such as user engagement and retention. While we may not be able to measure such metrics, we have knowledge about how certain model behaviors can influence them. Therefore, a practitioner may wish to choose a less accurate model for content recommendation that does not place high importance in features related to "clickbait" articles (which may hurt user retention), even if exploiting such features increases the accuracy of the model in cross validation. We note that explanations are particularly useful in these (and other) scenarios if a method can produce them for *any* model, so that a variety of models can be compared.

Desired Characteristics for Explainers

We now outline a number of desired characteristics from explanation methods.

An essential criterion for explanations is that they must be **interpretable**, i.e., provide qualitative understanding between the input variables and the response. We note that interpretability must take into account the user's limitations.

Thus, a linear model [24], a gradient vector [2] or an additive model [6] may or may not be interpretable. For example, if hundreds or thousands of features significantly contribute to a prediction, it is not reasonable to expect any user to comprehend why the prediction was made, even if individual weights can be inspected. This requirement further implies that explanations should be easy to understand, which is not necessarily true of the features used by the model, and thus the "input variables" in the explanations may need to be different than the features. Finally, we note that the notion of interpretability also depends on the target audience. Machine learning practitioners may be able to interpret small Bayesian networks, but laymen may be more comfortable with a small number of weighted features as an explanation.

Another essential criterion is **local fidelity**. Although it is often impossible for an explanation to be completely faithful unless it is the complete description of the model itself, for an explanation to be meaningful it must at least be *locally faithful*, i.e. it must correspond to how the model behaves in the vicinity of the instance being predicted. We note that local fidelity does not imply global fidelity: features that are globally important may not be important in the local context, and vice versa. While global fidelity would imply local fidelity, identifying globally faithful explanations that are interpretable remains a challenge for complex models.

While there are models that are inherently interpretable [6, 17, 26, 27], an explainer should be able to explain *any* model, and thus be **model-agnostic** (i.e. treat the original model as a black box). Apart from the fact that many state-of-the-art classifiers are not currently interpretable, this also provides flexibility to explain future classifiers.

In addition to explaining predictions, providing a **global perspective** is important to ascertain trust in the model. As mentioned before, accuracy may often not be a suitable metric to evaluate the model, and thus we want to *explain the model*. Building upon the explanations for individual predictions, we select a few explanations to present to the user, such that they are representative of the model.

3. LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS

We now present Local Interpretable Model-agnostic Explanations (**LIME**). The overall goal of LIME is to identify an **interpretable** model over the *interpretable representation* that is **locally faithful** to the classifier.

3.1 Interpretable Data Representations

Before we present the explanation system, it is important to distinguish between features and interpretable data representations. As mentioned before, **interpretable** explanations need to use a representation that is understandable to humans, regardless of the actual features used by the model. For example, a possible *interpretable representation* for text classification is a binary vector indicating the presence or absence of a word, even though the classifier may use more complex (and incomprehensible) features such as word embeddings. Likewise for image classification, an *interpretable representation* may be a binary vector indicating the "presence" or "absence" of a contiguous patch of similar pixels (a super-pixel), while the classifier may represent the image as a tensor with three color channels per pixel. We denote $x \in \mathbb{R}^d$ be the original representation of an instance being explained, and we use $x' \in \{0,1\}^{d'}$ to denote a binary vector for its interpretable representation.

3.2 Fidelity-Interpretability Trade-off

Formally, we define an explanation as a model $g \in G$, where G is a class of potentially *interpretable* models, such as linear models, decision trees, or falling rule lists [27], i.e. a model $g \in G$ can be readily presented to the user with visual or textual artifacts. The domain of g is $\{0,1\}^{d'}$, i.e. g acts over absence/presence of the *interpretable components*. As not every $g \in G$ may be simple enough to be interpretable - thus we let $\Omega(g)$ be a measure of *complexity* (as opposed to *interpretability*) of the explanation $g \in G$. For example, for decision trees $\Omega(g)$ may be the depth of the tree, while for linear models, $\Omega(g)$ may be the number of non-zero weights.

Let the model being explained be denoted $f : \mathbb{R}^d \rightarrow \mathbb{R}$. In classification, $f(x)$ is the probability (or a binary indicator) that x belongs to a certain class[1]. We further use $\pi_x(z)$ as a proximity measure between an instance z to x, so as to define locality around x. Finally, let $\mathcal{L}(f, g, \pi_x)$ be a measure of how unfaithful g is in approximating f in the locality defined by π_x. In order to ensure both **interpretability** and **local fidelity**, we must minimize $\mathcal{L}(f, g, \pi_x)$ while having $\Omega(g)$ be low enough to be interpretable by humans. The explanation produced by **LIME** is obtained by the following:

$$\xi(x) = \operatorname*{argmin}_{g \in G} \ \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{1}$$

This formulation can be used with different explanation families G, fidelity functions $\mathcal{L}$, and complexity measures Ω. Here we focus on sparse linear models as explanations, and on performing the search using perturbations.

3.3 Sampling for Local Exploration

We want to minimize the locality-aware loss $\mathcal{L}(f, g, \pi_x)$ without making any assumptions about f, since we want the explainer to be **model-agnostic**. Thus, in order to learn the local behavior of f as the interpretable inputs vary, we approximate $\mathcal{L}(f, g, \pi_x)$ by drawing samples, weighted by π_x. We sample instances around x' by drawing nonzero elements of x' uniformly at random (where the number of such draws is also uniformly sampled). Given a perturbed sample $z' \in \{0,1\}^{d'}$ (which contains a fraction of the nonzero elements of x'), we recover the sample in the original representation $z \in R^d$ and obtain $f(z)$, which is used as a *label* for the explanation model. Given this dataset $\mathcal{Z}$ of perturbed samples with the associated labels, we optimize Eq. (1) to get an explanation $\xi(x)$. The primary intuition behind LIME is presented in Figure 3, where we sample instances both in the vicinity of x (which have a high weight due to π_x) and far away from x (low weight from π_x). Even though the original model may be too complex to explain globally, LIME presents an explanation that is locally faithful (linear in this case), where the locality is captured by π_x. It is worth noting that our method is fairly robust to sampling noise since the samples are weighted by π_x in Eq. (1). We now present a concrete instance of this general framework.

3.4 Sparse Linear Explanations

For the rest of this paper, we let G be the class of linear models, such that $g(z') = w_g \cdot z'$. We use the locally weighted

[1]For multiple classes, we explain each class separately, thus $f(x)$ is the prediction of the relevant class.

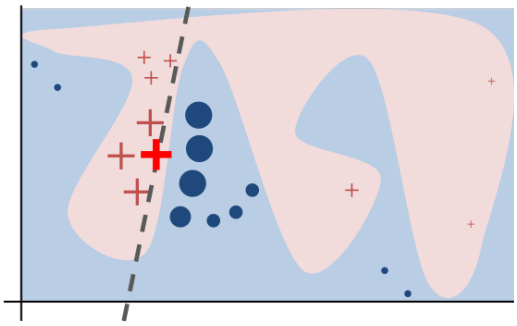


Figure 3: Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

square loss as $\mathcal{L}$, as defined in Eq. (2), where we let $\pi_x(z) = exp(-D(x,z)^2/\sigma^2)$ be an exponential kernel defined on some distance function D (e.g. cosine distance for text, $L2$ distance for images) with width σ.

$$\mathcal{L}(f, g, \pi_x) = \sum_{z,z' \in \mathcal{Z}} \pi_x(z) \left(f(z) - g(z')\right)^2 \qquad (2)$$

For text classification, we ensure that the explanation is **interpretable** by letting the *interpretable representation* be a bag of words, and by setting a limit K on the number of words, i.e. $\Omega(g) = \infty \mathbb{1}[\|w_g\|_0 > K]$. Potentially, K can be adapted to be as big as the user can handle, or we could have different values of K for different instances. In this paper we use a constant value for K, leaving the exploration of different values to future work. We use the same Ω for image classification, using "super-pixels" (computed using any standard algorithm) instead of words, such that the interpretable representation of an image is a binary vector where 1 indicates the original super-pixel and 0 indicates a grayed out super-pixel. This particular choice of Ω makes directly solving Eq. (1) intractable, but we approximate it by first selecting K features with Lasso (using the regularization path [9]) and then learning the weights via least squares (a procedure we call K-LASSO in Algorithm 1). Since Algorithm 1 produces an explanation for an individual prediction, its complexity does not depend on the size of the dataset, but instead on time to compute $f(x)$ and on the number of samples N. In practice, explaining random forests with 1000 trees using scikit-learn (http://scikit-learn.org) on a laptop with $N = 5000$ takes under 3 seconds without any optimizations such as using gpus or parallelization. Explaining each prediction of the Inception network [25] for image classification takes around 10 minutes.

Any choice of interpretable representations and G will have some inherent drawbacks. First, while the underlying model can be treated as a black-box, certain interpretable representations will not be powerful enough to explain certain behaviors. For example, a model that predicts sepia-toned images to be *retro* cannot be explained by presence of absence of super pixels. Second, our choice of G (sparse linear models) means that if the underlying model is highly non-linear even in the locality of the prediction, there may not be a faithful explanation. However, we can estimate the faithfulness of the explanation on $\mathcal{Z}$, and present this information to the user. This estimate of faithfulness can also be used for selecting an appropriate family of explanations from a set of multiple interpretable model classes, thus adapting to the given dataset and the classifier. We leave such exploration for future work, as linear explanations work quite well for multiple black-box models in our experiments.

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K
  $\mathcal{Z} \leftarrow \{\}$
  for $i \in \{1, 2, 3, ..., N\}$ **do**
    $z'_i \leftarrow sample\_around(x')$
    $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i)\rangle$
  end for
  $w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
  return w

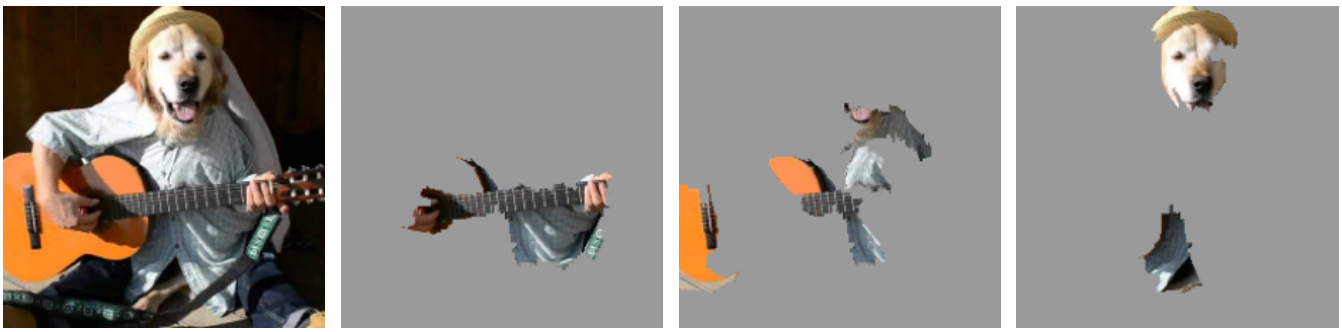
3.5 Example 1: Text classification with SVMs

In Figure 2 (right side), we explain the predictions of a support vector machine with RBF kernel trained on unigrams to differentiate "Christianity" from "Atheism" (on a subset of the 20 newsgroup dataset). Although this classifier achieves 94% held-out accuracy, and one would be tempted to trust it based on this, the explanation for an instance shows that predictions are made for quite arbitrary reasons (words "Posting", "Host", and "Re" have no connection to either Christianity or Atheism). The word "Posting" appears in 22% of examples in the training set, 99% of them in the class "Atheism". Even if headers are removed, proper names of prolific posters in the original newsgroups are selected by the classifier, which would also not generalize.

After getting such insights from explanations, it is clear that this dataset has serious issues (which are not evident just by studying the raw data or predictions), and that this classifier, or held-out evaluation, cannot be trusted. It is also clear what the problems are, and the steps that can be taken to fix these issues and train a more trustworthy classifier.

3.6 Example 2: Deep networks for images

When using sparse linear explanations for image classifiers, one may wish to just highlight the super-pixels with positive weight towards a specific class, as they give intuition as to why the model would think that class may be present. We explain the prediction of Google's pre-trained Inception neural network [25] in this fashion on an arbitrary image (Figure 4a). Figures 4b, 4c, 4d show the superpixels explanations for the top 3 predicted classes (with the rest of the image grayed out), having set $K = 10$. What the neural network picks up on for each of the classes is quite natural to humans - Figure 4b in particular provides insight as to why acoustic guitar was predicted to be electric: due to the fretboard. This kind of explanation enhances trust in the classifier (even if the top predicted class is wrong), as it shows that it is not acting in an unreasonable manner.



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

4. SUBMODULAR PICK FOR EXPLAINING MODELS

Although an explanation of a single prediction provides some understanding into the reliability of the classifier to the user, it is not sufficient to evaluate and assess trust in the model as a whole. We propose to give a global understanding of the model by explaining a set of individual instances. This approach is still model agnostic, and is complementary to computing summary statistics such as held-out accuracy.

Even though explanations of multiple instances can be insightful, these instances need to be selected judiciously, since users may not have the time to examine a large number of explanations. We represent the time/patience that humans have by a budget B that denotes the number of explanations they are willing to look at in order to understand a model. Given a set of instances X, we define the **pick step** as the task of selecting B instances for the user to inspect.

The pick step is not dependent on the existence of explanations - one of the main purpose of tools like Modeltracker [1] and others [11] is to assist users in selecting instances themselves, and examining the raw data and predictions. However, since looking at raw data is not enough to understand predictions and get insights, the pick step should take into account the explanations that accompany each prediction. Moreover, this method should pick a diverse, representative set of explanations to show the user – i.e. non-redundant explanations that represent how the model behaves globally.

Given the explanations for a set of instances X ($|X| = n$), we construct an $n \times d'$ *explanation matrix* $\mathcal{W}$ that represents the local importance of the interpretable components for each instance. When using linear models as explanations, for an instance x_i and explanation $g_i = \xi(x_i)$, we set $\mathcal{W}_{ij} = |w_{g_{ij}}|$. Further, for each component (column) j in $\mathcal{W}$, we let I_j denote the *global* importance of that component in the explanation space. Intuitively, we want I such that features that explain many different instances have higher importance scores. In Figure 5, we show a toy example $\mathcal{W}$, with $n = d' = 5$, where $\mathcal{W}$ is binary (for simplicity). The importance function I should score feature f2 higher than feature f1, i.e. $I_2 > I_1$, since feature f2 is used to explain more instances. Concretely for the text applications, we set $I_j = \sqrt{\sum_{i=1}^{n} \mathcal{W}_{ij}}$. For images, I must measure something that is comparable across the super-pixels in different images,

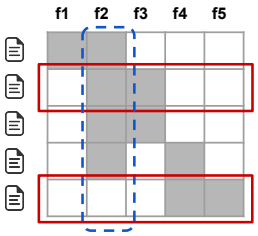


Figure 5: Toy example $\mathcal{W}$. Rows represent instances (documents) and columns represent features (words). Feature f2 (dotted blue) has the highest importance. Rows 2 and 5 (in red) would be selected by the pick procedure, covering all but feature f1.

Algorithm 2 Submodular pick (SP) algorithm

Require: Instances X, Budget B
  for all $x_i \in X$ **do**
    $\mathcal{W}_i \leftarrow$ **explain**(x_i, x_i') ▷ Using Algorithm 1
  end for
  for $j \in \{1 \dots d'\}$ **do**
    $I_j \leftarrow \sqrt{\sum_{i=1}^{n} |\mathcal{W}_{ij}|}$ ▷ Compute feature importances
  end for
  $V \leftarrow \{\}$
  while $|V| < B$ **do** ▷ Greedy optimization of Eq (4)
    $V \leftarrow V \cup \text{argmax}_i\, c(V \cup \{i\}, \mathcal{W}, I)$
  end while
  return V

such as color histograms or other features of super-pixels; we leave further exploration of these ideas for future work.

While we want to pick instances that cover the important components, the set of explanations must not be redundant in the components they show the users, i.e. avoid selecting instances with similar explanations. In Figure 5, after the second row is picked, the third row adds no value, as the user has already seen features f2 and f3 - while the last row exposes the user to completely new features. Selecting the second and last row results in the coverage of almost all the features. We formalize this non-redundant coverage intuition in Eq. (3), where we define coverage as the set function c that, given $\mathcal{W}$ and I, computes the total importance of the features that appear in at least one instance in a set V.

$$c(V, \mathcal{W}, I) = \sum_{j=1}^{d'} \mathbb{1}_{[\exists i \in V : \mathcal{W}_{ij} > 0]} I_j \quad (3)$$

The pick problem, defined in Eq. (4), consists of finding the set $V, |V| \leq B$ that achieves highest coverage.

$$Pick(\mathcal{W}, I) = \underset{V, |V| \leq B}{\operatorname{argmax}} c(V, \mathcal{W}, I) \quad (4)$$

The problem in Eq. (4) is maximizing a weighted coverage function, and is NP-hard [10]. Let $c(V \cup \{i\}, \mathcal{W}, I) - c(V, \mathcal{W}, I)$ be the marginal coverage gain of adding an instance i to a set V. Due to submodularity, a greedy algorithm that iteratively adds the instance with the highest marginal coverage gain to the solution offers a constant-factor approximation guarantee of $1-1/e$ to the optimum [15]. We outline this approximation in Algorithm 2, and call it **submodular pick**.

5. SIMULATED USER EXPERIMENTS

In this section, we present simulated user experiments to evaluate the utility of explanations in trust-related tasks. In particular, we address the following questions: (1) Are the explanations faithful to the model, (2) Can the explanations aid users in ascertaining trust in predictions, and (3) Are the explanations useful for evaluating the model as a whole. Code and data for replicating our experiments are available at https://github.com/marcotcr/lime-experiments.

5.1 Experiment Setup

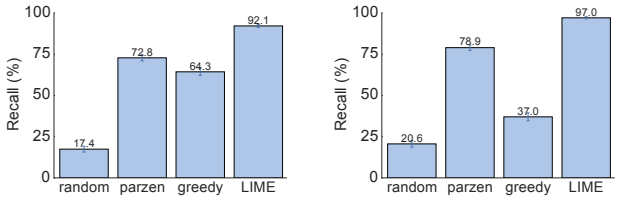
We use two sentiment analysis datasets (*books* and *DVDs*, 2000 instances each) where the task is to classify product reviews as positive or negative [4]. We train decision trees (**DT**), logistic regression with L2 regularization (**LR**), nearest neighbors (**NN**), and support vector machines with RBF kernel (**SVM**), all using bag of words as features. We also include random forests (with 1000 trees) trained with the average word2vec embedding [19] (**RF**), a model that is impossible to interpret without a technique like LIME. We use the implementations and default parameters of scikit-learn, unless noted otherwise. We divide each dataset into train (1600 instances) and test (400 instances).

To explain individual predictions, we compare our proposed approach (**LIME**), with **parzen** [2], a method that approximates the black box classifier globally with Parzen windows, and explains individual predictions by taking the gradient of the prediction probability function. For parzen, we take the K features with the highest absolute gradients as explanations. We set the hyper-parameters for parzen and LIME using cross validation, and set $N = 15,000$. We also compare against a **greedy** procedure (similar to **(author?)** [18]) in which we greedily remove features that contribute the most to the predicted class until the prediction changes (or we reach the maximum of K features), and a **random** procedure that randomly picks K features as an explanation. We set K to 10 for our experiments.

For experiments where the pick procedure applies, we either do random selection (random pick, **RP**) or the procedure described in §4 (submodular pick, **SP**). We refer to pick-explainer combinations by adding RP or SP as a prefix.

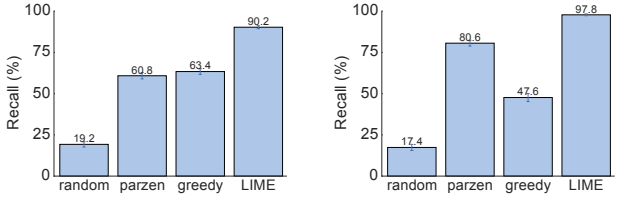
5.2 Are explanations faithful to the model?

We measure faithfulness of explanations on classifiers that are by themselves interpretable (sparse logistic regression and decision trees). In particular, we train both classifiers such that the maximum number of features they use for any instance is 10, and thus we know the *gold* set of features that the are considered important by these models. For each prediction on the test set, we generate explanations and compute the fraction of these *gold* features that are recovered by the explanations. We report this recall averaged over all the test instances in Figures 6 and 7. We observe that the greedy approach is comparable to parzen on logistic regression, but is substantially worse on decision trees since changing a single feature at a time often does not have an effect on the prediction. The overall recall by parzen is low, likely due to the difficulty in approximating the original high-dimensional classifier. LIME consistently provides $> 90\%$ recall for both classifiers on both datasets, demonstrating that LIME explanations are faithful to the models.



(a) Sparse LR (b) Decision Tree

Figure 6: Recall on truly important features for two interpretable classifiers on the books dataset.



(a) Sparse LR (b) Decision Tree

Figure 7: Recall on truly important features for two interpretable classifiers on the DVDs dataset.

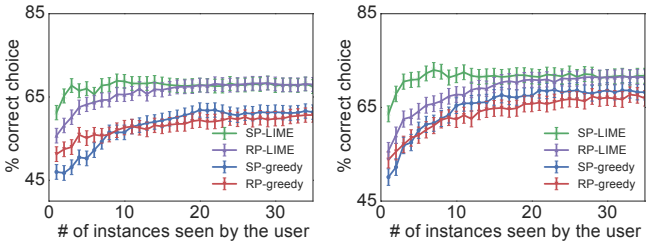
5.3 Should I trust this prediction?

In order to simulate trust in individual predictions, we first randomly select 25% of the features to be "untrustworthy", and assume that the users can identify and would not want to trust these features (such as the headers in 20 newsgroups, leaked data, etc). We thus develop *oracle* "trustworthiness" by labeling test set predictions from a black box classifier as "untrustworthy" if the prediction changes when untrustworthy features are removed from the instance, and "trustworthy" otherwise. In order to simulate users, we assume that users deem predictions untrustworthy from LIME and parzen explanations if the prediction from the linear approximation changes when all untrustworthy features that appear in the explanations are removed (the simulated human "discounts" the effect of untrustworthy features). For greedy and random, the prediction is mistrusted if any untrustworthy features are present in the explanation, since these methods do not provide a notion of the contribution of each feature to the prediction. Thus for each test set prediction, we can evaluate whether the simulated user trusts it using each explanation method, and compare it to the trustworthiness oracle.

Using this setup, we report the F1 on the trustworthy

Table 1: Average F1 of *trustworthiness* for different explainers on a collection of classifiers and datasets.

	Books				DVDs			
	LR	NN	RF	SVM	LR	NN	RF	SVM
Random	14.6	14.8	14.7	14.7	14.2	14.3	14.5	14.4
Parzen	84.0	87.6	94.3	92.3	87.0	81.7	94.2	87.3
Greedy	53.7	47.4	45.0	53.3	52.4	58.1	46.6	55.1
LIME	**96.6**	**94.5**	**96.2**	**96.7**	**96.6**	**91.8**	**96.1**	**95.6**



(a) Books dataset (b) DVDs dataset

Figure 8: Choosing between two classifiers, as the number of instances shown to a simulated user is varied. Averages and standard errors from 800 runs.

predictions for each explanation method, averaged over 100 runs, in Table 1. The results indicate that LIME dominates others (all results are significant at $p = 0.01$) on both datasets, and for all of the black box models. The other methods either achieve a lower recall (i.e. they mistrust predictions more than they should) or lower precision (i.e. they trust too many predictions), while LIME maintains both high precision and high recall. Even though we artificially select which features are untrustworthy, these results indicate that LIME is helpful in assessing trust in individual predictions.

5.4 Can I trust this model?

In the final simulated user experiment, we evaluate whether the explanations can be used for model selection, simulating the case where a human has to decide between two competing models with similar accuracy on validation data. For this purpose, we add 10 artificially "noisy" features. Specifically, on training and validation sets (80/20 split of the original training data), each artificial feature appears in 10% of the examples in one class, and 20% of the other, while on the test instances, each artificial feature appears in 10% of the examples in each class. This recreates the situation where the models use not only features that are informative in the real world, but also ones that introduce spurious correlations. We create pairs of competing classifiers by repeatedly training pairs of random forests with 30 trees until their validation accuracy is within 0.1% of each other, but their test accuracy differs by at least 5%. Thus, it is not possible to identify the *better* classifier (the one with higher test accuracy) from the accuracy on the validation data.

The goal of this experiment is to evaluate whether a user can identify the better classifier based on the explanations of B instances from the validation set. The simulated human marks the set of artificial features that appear in the B explanations as untrustworthy, following which we evaluate how many total predictions in the validation set should be trusted (as in the previous section, treating only marked features as untrustworthy). Then, we select the classifier with fewer untrustworthy predictions, and compare this choice to the classifier with higher held-out test set accuracy.

We present the accuracy of picking the correct classifier as B varies, averaged over 800 runs, in Figure 8. We omit SP-parzen and RP-parzen from the figure since they did not produce useful explanations, performing only slightly better than random. LIME is consistently better than greedy, irrespective of the pick method. Further, combining submodular pick with LIME outperforms all other methods, in particular it is much better than RP-LIME when only a few examples are shown to the users. These results demonstrate that the trust assessments provided by SP-selected LIME explanations are good indicators of generalization, which we validate with human experiments in the next section.

6. EVALUATION WITH HUMAN SUBJECTS

In this section, we recreate three scenarios in machine learning that require trust and understanding of predictions and models. In particular, we evaluate LIME and SP-LIME in the following settings: (1) Can users choose which of two classifiers generalizes better (§ 6.2), (2) based on the explanations, can users perform feature engineering to improve the model (§ 6.3), and (3) are users able to identify and describe classifier irregularities by looking at explanations (§ 6.4).

6.1 Experiment setup

For experiments in §6.2 and §6.3, we use the "Christianity" and "Atheism" documents from the 20 newsgroups dataset mentioned beforehand. This dataset is problematic since it contains features that do not generalize (e.g. very informative header information and author names), and thus validation accuracy considerably overestimates real-world performance.

In order to estimate the real world performance, we create a new *religion dataset* for evaluation. We download Atheism and Christianity websites from the DMOZ directory and human curated lists, yielding 819 webpages in each class. High accuracy on this dataset by a classifier trained on 20 newsgroups indicates that the classifier is generalizing using semantic content, instead of placing importance on the data specific issues outlined above. Unless noted otherwise, we use SVM with RBF kernel, trained on the 20 newsgroups data with hyper-parameters tuned via the cross-validation.

6.2 Can users select the best classifier?

In this section, we want to evaluate whether explanations can help users decide which classifier generalizes better, i.e., which classifier would the user deploy "in the wild". Specifically, users have to decide between two classifiers: SVM trained on the original 20 newsgroups dataset, and a version of the same classifier trained on a "cleaned" dataset where many of the features that do not generalize have been manually removed. The original classifier achieves an accuracy score of 57.3% on the *religion dataset*, while the "cleaned" classifier achieves a score of 69.0%. In contrast, the test accuracy on the original 20 newsgroups split is 94.0% and 88.6%, respectively – suggesting that the worse classifier would be selected if accuracy alone is used as a measure of trust.

We recruit human subjects on Amazon Mechanical Turk – by no means machine learning experts, but instead people with basic knowledge about religion. We measure their ability to choose the better algorithm by seeing side-by-side explanations with the associated raw data (as shown in Figure 2). We restrict both the number of words in each explanation (K) and the number of documents that each

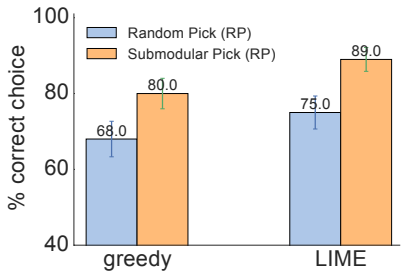


Figure 9: Average accuracy of human subject (with standard errors) in choosing between two classifiers.

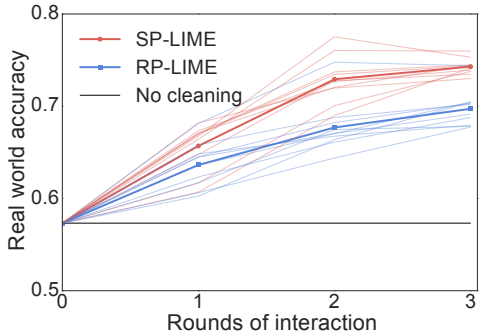


Figure 10: Feature engineering experiment. Each shaded line represents the average accuracy of subjects in a path starting from one of the initial 10 subjects. Each solid line represents the average across all paths per round of interaction.

person inspects (B) to 6. The position of each algorithm and the order of the instances seen are randomized between subjects. After examining the explanations, users are asked to select which algorithm will perform best in the real world. The explanations are produced by either greedy (chosen as a baseline due to its performance in the simulated user experiment) or LIME, and the instances are selected either by random (RP) or submodular pick (SP). We modify the greedy step in Algorithm 2 slightly so it alternates between explanations of the two classifiers. For each setting, we repeat the experiment with 100 users.

The results are presented in Figure 9. Note that all of the methods are good at identifying the better classifier, demonstrating that the explanations are useful in determining which classifier to trust, while using test set accuracy would result in the selection of the wrong classifier. Further, we see that the submodular pick (SP) greatly improves the user's ability to select the best classifier when compared to random pick (RP), with LIME outperforming greedy in both cases.

6.3 Can non-experts improve a classifier?

If one notes that a classifier is untrustworthy, a common task in machine learning is feature engineering, i.e. modifying the set of features and retraining in order to improve generalization. Explanations can aid in this process by presenting the important features, particularly for removing features that the users feel do not generalize.

We use the 20 newsgroups data here as well, and ask Amazon Mechanical Turk users to identify which words from the explanations should be removed from subsequent training, for the worse classifier from the previous section (§6.2). In each round, the subject marks words for deletion after observing $B = 10$ instances with $K = 10$ words in each explanation (an interface similar to Figure 2, but with a single algorithm). As a reminder, the users here are not experts in machine learning and are unfamiliar with feature engineering, thus are only identifying words based on their semantic content. Further, users do not have any access to the *religion* dataset – they do not even know of its existence. We start the experiment with 10 subjects. After they mark words for deletion, we train 10 different classifiers, one for each subject (with the corresponding words removed). The explanations for each classifier are then presented to a set of 5 users in a new round of interaction, which results in 50 new classifiers. We do a final round, after which we have 250 classifiers, each with a path of interaction tracing back to the first 10 subjects.

The explanations and instances shown to each user are produced by **SP-LIME** or **RP-LIME**. We show the average accuracy on the *religion* dataset at each interaction round for the paths originating from each of the original 10 subjects (shaded lines), and the average across all paths (solid lines) in Figure 10. It is clear from the figure that the crowd workers are able to improve the model by removing features they deem unimportant for the task. Further, **SP-LIME** outperforms **RP-LIME**, indicating selection of the instances to show the users is crucial for efficient feature engineering.

Each subject took an average of 3.6 minutes per round of cleaning, resulting in just under 11 minutes to produce a classifier that generalizes much better to real world data. Each path had on average 200 words removed with **SP**, and 157 with **RP**, indicating that incorporating coverage of important features is useful for feature engineering. Further, out of an average of 200 words selected with SP, 174 were selected by at least half of the users, while 68 by *all* the users. Along with the fact that the variance in the accuracy decreases across rounds, this high agreement demonstrates that the users are converging to similar *correct* models. This evaluation is an example of how explanations make it easy to improve an untrustworthy classifier – in this case easy enough that machine learning knowledge is not required.

6.4 Do explanations lead to insights?

Often artifacts of data collection can induce undesirable correlations that the classifiers pick up during training. These issues can be very difficult to identify just by looking at the raw data and predictions. In an effort to reproduce such a setting, we take the task of distinguishing between photos of Wolves and Eskimo Dogs (huskies). We train a logistic regression classifier on a training set of 20 images, hand selected such that all pictures of wolves had snow in the background, while pictures of huskies did not. As the features for the images, we use the first max-pooling layer of Google's pre-trained Inception neural network [25]. On a collection of additional 60 images, the classifier predicts "Wolf" if there is snow (or light background at the bottom), and "Husky" otherwise, regardless of animal color, position, pose, etc. We trained this *bad* classifier intentionally, to evaluate whether subjects are able to detect it.

The experiment proceeds as follows: we first present a balanced set of 10 test predictions (without explanations), where one wolf is not in a snowy background (and thus the prediction is "Husky") and one husky is (and is thus predicted as "Wolf"). We show the "Husky" mistake in Figure 11a. The other 8 examples are classified correctly. We then ask the subject three questions: (1) Do they trust this algorithm

(a) Husky classified as wolf (b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

	Before	After
Trusted the bad model	10 out of 27	3 out of 27
Snow as a potential feature	12 out of 27	25 out of 27

Table 2: "Husky vs Wolf" experiment results.

to work well in the real world, (2) why, and (3) how do they think the algorithm is able to distinguish between these photos of wolves and huskies. After getting these responses, we show the same images with the associated explanations, such as in Figure 11b, and ask the same questions.

Since this task requires some familiarity with the notion of spurious correlations and generalization, the set of subjects for this experiment were graduate students who have taken at least one graduate machine learning course. After gathering the responses, we had 3 independent evaluators read their reasoning and determine if each subject mentioned snow, background, or equivalent as a feature the model may be using. We pick the majority to decide whether the subject was correct about the insight, and report these numbers before and after showing the explanations in Table 2.

Before observing the explanations, more than a third trusted the classifier, and a little less than half mentioned the snow pattern as something the neural network was using – although all speculated on other patterns. After examining the explanations, however, almost all of the subjects identified the correct insight, with much more certainty that it was a determining factor. Further, the trust in the classifier also dropped substantially. Although our sample size is small, this experiment demonstrates the utility of explaining individual predictions for getting insights into classifiers knowing when not to trust them and why.

7. RELATED WORK

The problems with relying on validation set accuracy as the primary measure of trust have been well studied. Practitioners consistently overestimate their model's accuracy [21], propagate feedback loops [23], or fail to notice data leaks [14]. In order to address these issues, researchers have proposed tools like Gestalt [20] and Modeltracker [1], which help users navigate individual instances. These tools are complementary to LIME in terms of explaining models, since they do not address the problem of explaining individual predictions. Further, our submodular pick procedure can be incorporated in such tools to aid users in navigating larger datasets.

Some recent work aims to anticipate failures in machine learning, specifically for vision tasks [3, 29]. Letting users know when the systems are likely to fail can lead to an increase in trust, by avoiding "silly mistakes" [8]. These solutions either require additional annotations and feature engineering that is specific to vision tasks or do not provide insight into why a decision should not be trusted. Furthermore, they assume that the current evaluation metrics are reliable, which may not be the case if problems such as data leakage are present. Other recent work [11] focuses on exposing users to different kinds of mistakes (our pick step). Interestingly, the subjects in their study did not notice the serious problems in the 20 newsgroups data even after looking at many mistakes, suggesting that examining raw data is not sufficient. Note that **(author?)** [11] are not alone in this regard, many researchers in the field have unwittingly published classifiers that would not generalize for this task. Using LIME, we show that even non-experts are able to identify these irregularities when explanations are present. Further, LIME can complement these existing systems, and allow users to assess trust even when a prediction seems "correct" but is made for the wrong reasons.

Recognizing the utility of explanations in assessing trust, many have proposed using interpretable models [27], especially for the medical domain [6, 17, 26]. While such models may be appropriate for some domains, they may not apply equally well to others (e.g. a supersparse linear model [26] with $5 - 10$ features is unsuitable for text applications). Interpretability, in these cases, comes at the cost of flexibility, accuracy, or efficiency. For text, EluciDebug [16] is a full human-in-the-loop system that shares many of our goals (interpretability, faithfulness, etc). However, they focus on an already interpretable model (Naive Bayes). In computer vision, systems that rely on object detection to produce candidate alignments [13] or attention [28] are able to produce explanations for their predictions. These are, however, constrained to specific neural network architectures or incapable of detecting "non object" parts of the images. Here we focus on general, model-agnostic explanations that can be applied to any classifier or regressor that is appropriate for the domain - even ones that are yet to be proposed.

A common approach to model-agnostic explanation is learning a potentially interpretable model on the predictions of the original model [2, 7, 22]. Having the explanation be a gradient vector [2] captures a similar locality intuition to that of LIME. However, interpreting the coefficients on the gradient is difficult, particularly for confident predictions (where gradient is near zero). Further, these explanations approximate the original model *globally*, thus maintaining local fidelity becomes a significant challenge, as our experiments demonstrate. In contrast, LIME solves the much more feasible task of finding a model that approximates the original model *locally*. The idea of perturbing inputs for explanations has been explored before [24], where the authors focus on learning a specific *contribution* model, as opposed to our general framework. None of these approaches explicitly take cognitive limitations into account, and thus may produce non-interpretable explanations, such as a gradients or linear models with thousands of non-zero weights. The problem becomes worse if the original features are nonsensical to humans (e.g. word embeddings). In contrast, LIME incorporates interpretability both in the optimization and in our notion of *interpretable representation*, such that domain and task specific interpretability criteria can be accommodated.

8. CONCLUSION AND FUTURE WORK

In this paper, we argued that trust is crucial for effective human interaction with machine learning systems, and that explaining individual predictions is important in assessing trust. We proposed LIME, a modular and extensible approach to faithfully explain the predictions of *any* model in an interpretable manner. We also introduced SP-LIME, a method to select representative and non-redundant predictions, providing a global view of the model to users. Our experiments demonstrated that explanations are useful for a variety of models in trust-related tasks in the text and image domains, with both expert and non-expert users: deciding between models, assessing trust, improving untrustworthy models, and getting insights into predictions.

There are a number of avenues of future work that we would like to explore. Although we describe only sparse linear models as explanations, our framework supports the exploration of a variety of explanation families, such as decision trees; it would be interesting to see a comparative study on these with real users. One issue that we do not mention in this work was how to perform the pick step for images, and we would like to address this limitation in the future. The domain and model agnosticism enables us to explore a variety of applications, and we would like to investigate potential uses in speech, video, and medical domains, as well as recommendation systems. Finally, we would like to explore theoretical properties (such as the appropriate number of samples) and computational optimizations (such as using parallelization and GPU processing), in order to provide the accurate, real-time explanations that are critical for any human-in-the-loop machine learning system.

Acknowledgements

We would like to thank Scott Lundberg, Tianqi Chen, and Tyler Johnson for helpful discussions and feedback. This work was supported in part by ONR awards #W911NF-13-1-0246 and #N00014-13-1-0023, and in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

9. REFERENCES

[1] S. Amershi, M. Chickering, S. M. Drucker, B. Lee, P. Simard, and J. Suh. Modeltracker: Redesigning performance analysis tools for machine learning. In *Human Factors in Computing Systems (CHI)*, 2015.

[2] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11, 2010.

[3] A. Bansal, A. Farhadi, and D. Parikh. Towards transparent systems: Semantic characterization of failure modes. In *European Conference on Computer Vision (ECCV)*, 2014.

[4] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Association for Computational Linguistics (ACL)*, 2007.

[5] J. Q. Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. MIT, 2009.

[6] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Knowledge Discovery and Data Mining (KDD)*, 2015.

[7] M. W. Craven and J. W. Shavlik. Extracting tree-structured representations of trained networks. *Neural information processing systems (NIPS)*, pages 24–30, 1996.

[8] M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck. The role of trust in automation reliance. *Int. J. Hum.-Comput. Stud.*, 58(6), 2003.

[9] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[10] U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4), July 1998.

[11] A. Groce, T. Kulesza, C. Zhang, S. Shamasunder, M. Burnett, W.-K. Wong, S. Stumpf, S. Das, A. Shinsel, F. Bice, and K. McIntosh. You are the only possible oracle: Effective test selection for end users of interactive machine learning systems. *IEEE Trans. Softw. Eng.*, 40(3), 2014.

[12] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Conference on Computer Supported Cooperative Work (CSCW)*, 2000.

[13] A. Karpathy and F. Li. Deep visual-semantic alignments for generating image descriptions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[14] S. Kaufman, S. Rosset, and C. Perlich. Leakage in data mining: Formulation, detection, and avoidance. In *Knowledge Discovery and Data Mining (KDD)*, 2011.

[15] A. Krause and D. Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, February 2014.

[16] T. Kulesza, M. Burnett, W.-K. Wong, and S. Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Intelligent User Interfaces (IUI)*, 2015.

[17] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics*, 2015.

[18] D. Martens and F. Provost. Explaining data-driven document classifications. *MIS Q.*, 38(1), 2014.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems (NIPS)*. 2013.

[20] K. Patel, N. Bancroft, S. M. Drucker, J. Fogarty, A. J. Ko, and J. Landay. Gestalt: Integrated support for implementation and analysis in machine learning. In *User Interface Software and Technology (UIST)*, 2010.

[21] K. Patel, J. Fogarty, J. A. Landay, and B. Harrison. Investigating statistical machine learning as a tool for software development. In *Human Factors in Computing Systems (CHI)*, 2008.

[22] I. Sanchez, T. Rocktaschel, S. Riedel, and S. Singh. Towards extracting faithful and descriptive representations of latent variable models. In *AAAI Spring Syposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches*, 2015.

[23] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, and J.-F. Crespo. Hidden technical debt in machine learning systems. In *Neural Information Processing Systems (NIPS)*. 2015.

[24] E. Strumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11, 2010.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[26] B. Ustun and C. Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 2015.

[27] F. Wang and C. Rudin. Falling rule lists. In *Artificial Intelligence and Statistics (AISTATS)*, 2015.

[28] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*, 2015.

[29] P. Zhang, J. Wang, A. Farhadi, M. Hebert, and D. Parikh. Predicting failures of vision systems. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

Direct-Manipulation Visualization of Deep Networks

Daniel Smilkov
Google, Inc.
5 Cambridge Center,
Cambridge MA, 02142
smilkov@google.com

Shan Carter
Google, Inc.
1600 Amphitheater Parkway,
Mountain View CA, 94043
shancarter@google.com

D. Sculley
Google, Inc.
5 Cambridge Center,
Cambridge MA, 02142
dsculley@google.com

Fernanda B. Viégas
Google, Inc.
5 Cambridge Center,
Cambridge MA, 02142
viegas@google.com

Martin Wattenberg
Google, Inc.
5 Cambridge Center,
Cambridge MA, 02142
wattenberg@google.com

ABSTRACT

Disclaimer: This work has been previously submitted to the ICML 2016 Workshop on Visualization for Deep Learning. We are submitting it here with permission from organizers of both workshops.

The recent successes of deep learning have led to a wave of interest from non-experts. Gaining an understanding of this technology, however, is difficult. While the theory is important, it is also helpful for novices to develop an intuitive feel for the effect of different hyperparameters and structural variations. We describe TensorFlow Playground[1], an interactive, open sourced[2] visualization that allows users to experiment via direct manipulation rather than coding, enabling them to quickly build an intuition about neural nets.

1. INTRODUCTION

Deep learning systems are currently attracting a huge amount of interest, as they see continued success in practical applications. Students who want to understand this new technology encounter two primary challenges.

First, the theoretical foundations of the field are not always easy for a typical software engineer or computer science student, since they require a solid mathematical intuition. It's not trivial to translate the equations defining a deep network into a mental model of the underlying geometric transformations.

Even more challenging are aspects of deep learning where theory does not provide crisp, clean explanations. Critical choices experts make in building a real-world system–the number of units and layers, the activation function, regularization techniques, etc.–are currently guided by intuition and experience as much as theory. Acquiring this intuition is a lengthy process, since it typically requires coding and training many different working systems.

One possible shortcut is to use interactive visualization to help novices with mathematical and practical intuition. Recently, several impressive systems have appeared that do exactly this. Olah's elegant interactive online essays [5] let a viewer watch the training of a simple classifier, providing a multiple perspectives on how a network learns a transformation of space. Karpathy created a Javascript library [4] and provided a series of dynamic views of networks training, again in a browser. Others have found beautiful ways to visualize the features learned by image classification nets [10], [9].

Taking inspiration from the success of these examples, we created the TensorFlow Playground. As with the work of Olah and Karpathy, the Playground is an in-browser visualization of a running neural network. However, it is specifically designed for experimentation by direct manipulation, and also visualizes the derived "features" found by every unit in the network simultaneously. The system provides a variety of affordances for rapidly and incrementally changing hyperparameters and immediately seeing the effects of those changes, as well as for sharing experiments with others.

2. TENSORFLOW PLAYGROUND: VISUALIZATION

The structure of the Playground visualization is a standard network diagram. The visualization shows a network that is designed to solve either classification or regression problems based on two abstract real-valued features, x_1 and x_2, which vary between -1 and 1. Input units, representing these features and various mathematical combinations, are at the left. Units in hidden layers are shown as small boxes, with connections between units drawn as curves whose color and width indicate weight values. Finally, on the right, a visualization of the output of the network is shown: a square with a heatmap showing the output value of the single unit that makes up the final layer of the network. When the user presses the "play" button, the network begins to train.

There is a new twist in this visualization, however. Inside the box that represents each unit is a heatmap that maps the unit's response to all values of (x_1, x_2) in a square centered at the origin. As seen in Figure 1, this provides a quick geo-

[1]http://playground.tensorflow.org
[2]https://github.com/tensorflow/playground

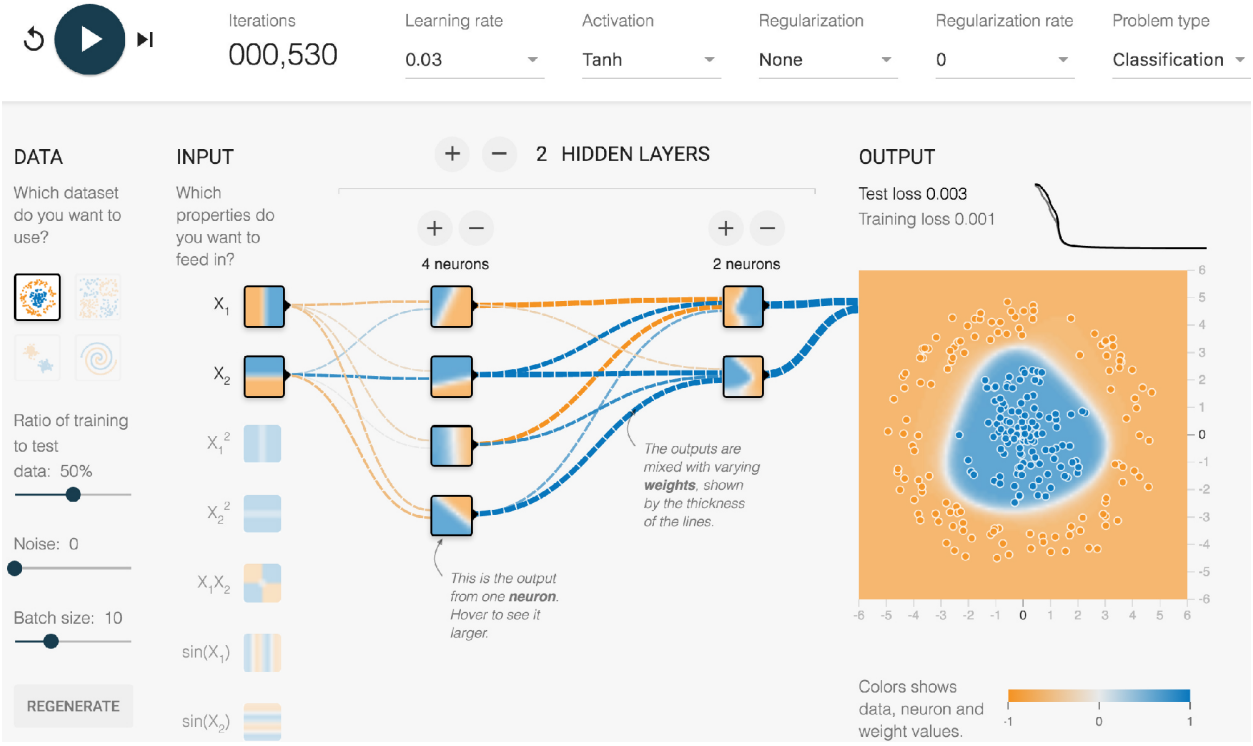


Figure 1: TensorFlow Playground. This network is, roughly speaking, classifying data based on distance to the origin. Curves show weight parameters, with thickness denoting absolute magnitude and color indicating sign. The feature heatmaps for each unit show how the classification function (large heatmap at right) is built from input features, then near-linear combinations of these features, and finally more complex features. At upper right is a graph showing loss over time. At left are possible features; x_1 and x_2 are highlighted, while other mathematical combinations are faded to indicate they should not be used by the network.

metric view of how the network builds complex features from simpler ones. For example, in the figure the input features are simply x_1 and x_2, which themselves are represented by the same type of heatmap. In the next layer, we see units that correspond to various linear combinations, leading to a final layer with more complicated non-linear classifiers. Moving the mouse over any of these units projects a larger version of the heatmap, on the final unit, where it can be overlaid with input and test data.

The activation heatmaps help users build a mental model of the mathematics underlying deep networks. For many configurations of the network, after training there is an obvious visual progression in complexity across the network. In these configurations, viewers can see how the first layer of units (modulo activation function, acting as linear classifiers) combine to recognize clearly nonlinear regions. The heatmaps also help viewers understand the different effects of various activation functions. For example, there is a clear visual difference in the effect of ReLU and tanh functions. Just as instructive, however, are suboptimal combinations of architecture and hyperparameters. Often when there are redundant units (Figure 3), it is easy to see that units in intermediate layers have actually learned the classifier perfectly well and that many other units have little effect on the final outcome. In cases where learning is simply unsuccessful, the viewer will often see weights going to zero, and that there is no natural progression of complexity in the activation heatmaps (Figure 4).

The visualization is implemented in JavaScript using d3.js[3]. It is worth noting that for the neural network computation, we are not using the TensorFlow library[1] since we needed the whole visualization to run in the browser. Instead, we wrote a small library[3] that meets the demands of this educational visualization.

3. AFFORDANCES FOR EDUCATION AND EXPERIMENTATION

The real strength of this visualization is its interactivity, which is especially helpful for gaining an intuition for the practical aspects of training a deep network. The Playground lets users make the following choices of network structure and hyperparameters:

- Problem type: regression or classification
- Training data: a choice of four synthetic data sets, from well-separated clusters to interleaved "swiss roll" spirals.
- Number of layers

[3]https://github.com/tensorflow/playground/blob/master/nn.ts

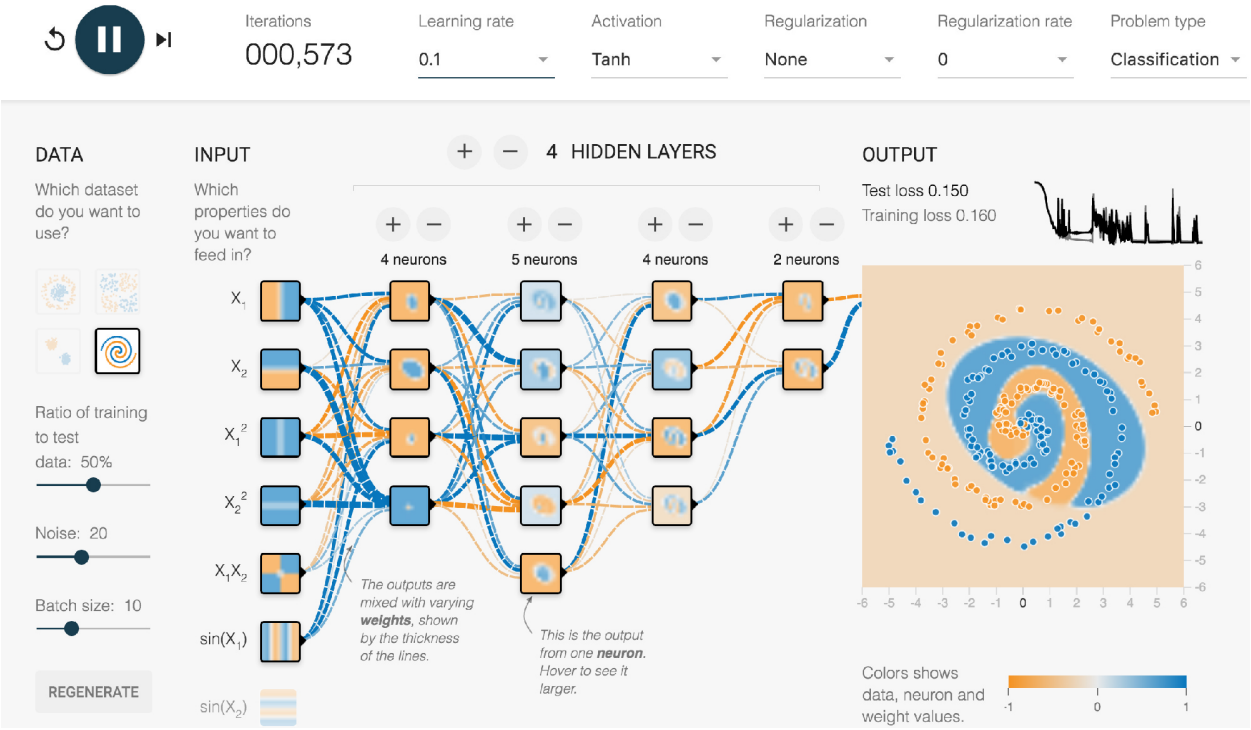


Figure 2: A complex configuration of TensorFlow Playground, in which a user is attempting to find hyperparameters that will allow the classification of spiral data. Many possible feature combinations have been activated.

- Number of units in each layer
- Activation function
- Learning rate
- Batch size
- Regularization: L^1, L^2, or none
- Input features: in addition to the two real-valued features x_1 and x_2, the Playground allows users to add some simple algebraic combinations, such as $x_1 x_2$ and x_1^2.
- Noise level for input data

These particular variations were chosen based on experience teaching software engineers how to use neural networks in their applications, and are meant to highlight key decisions that are made in real life. They are also meant to be easily combined to support particular lessons. For instance, allowing users to add algebraic combinations of the two primary features makes it easy to show how a linear classifier can do "non-linear" tasks when given non-linear feature combinations.

The user interface is designed to make these choices as easy to modify as possible. The standard definition of direct manipulation is that changes should be "rapid, incremental and reversible" [7]. Allowing fast, smooth changes to variables helps build intuition for their effects. Reversibility encourages experimentation: indeed, we chose as our tagline for the visualization, "You can't break it. We promise."

Additional aspects of the visualization make it well-suited to education. We have found that the smooth animation engages users. It also lends itself to a good "spectator experience" [6], drawing students in during presentations. We have seen onlookers laugh and even gasp as they watch a network try and fail to classify the spiral data set, for example. Although animation has not always been found to be helpful in educational contexts, simulations are one case where there is good evidence that it is beneficial [2].

One particularly important feature is the ability to seamlessly bookmark [8] a particular configuration of hyperparameters and structure. As the user plays with the tool, the URL in the browser dynamically updates to reflect its current state. If the user (or a teacher preparing a lesson plan) finds a configuration they would like to share with others, they need only copy the URL. Additionally, using the checkboxes below the visualization, each UI component can be hidden, making it easy to repurpose the interface.

We have found this bookmarking capability invaluable in the teaching process. For example, it has allowed us to put together tutorials in which students can move, step by step, through a series of lessons that focus on particular aspects of neural networks. Using the visualization in these "living lessons" makes it straightforward to create a dynamic, interactive educational experience.

4. CONCLUSION AND FUTURE WORK

The TensorFlow Playground illustrates a direct-manipulation approach to understanding neural nets. Given the importance of intuition and experimentation to the field of deep learning, the visualization is designed to make it easy to get a hands-on feel for how these systems work without any coding. Not only does this extend the reach of the tool to people who aren't programmers, it provides a much faster route, even for coders, to try many variations quickly. By playing with the visualization, users have a chance to build a mental model of the mathematics behind deep learning, as well as develop a natural feeling for how these networks respond to tweaks in architecture and hyperparameters.

In addition to internal success with the tool, we have seen a strong positive reaction since it has been open-sourced. Besides general positive comments, we have seen interesting, playful interactions. On one Reddit thread, for example, people competed to find a way to classify the spiral data, posting screenshots of their successful configurations. This suggests that the tool is instigating a vibrant social reaction to the visualization.

Since the launch of TensorFlow Playground, we have seen many suggestions for extensions. Affordances for many other structural variations and hyperparameters could be added; for instance, a common request is for an option to see the effect of dropout. Architectures such as convolutional nets and LSTMs could also be illuminated through direct manipulation techniques. Our hope is that, as an open-source project, the Playground will be extended to accommodate many such ideas. More broadly, the ideas of visualization, direct manipulation, and shareability that we have used may prove useful in explaining other aspects of deep learning besides network structure and hyperparameters.

A further question is whether this same direct-manipulation environment can be extended to help researchers as well as students. While there are obvious technical obstacles–breaking new ground often requires large data sets and computational resources beyond what a browser offers–it may be possible to create minimal "research playgrounds" that yield insights and allow rapid experimentation.

5. REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] M. Betrancourt. The animation and interactivity principles in multimedia learning. *The Cambridge handbook of multimedia learning*, pages 287–296, 2005.

[3] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.

[4] A. Karpathy. Convnetjs: Deep learning in your browser.

[5] C. Olah. colah's blog.

[6] S. Reeves, S. Benford, C. O'Malley, and M. Fraser. Designing the spectator experience. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 741–750. ACM, 2005.

[7] B. Shneiderman. 1.1 direct manipulation: a step beyond programming languages. *Sparks of innovation in human-computer interaction*, 17:1993, 1993.

[8] F. B. Viegas, M. Wattenberg, F. Van Ham, J. Kriss, and M. McKeon. Manyeyes: a site for visualization at internet scale. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1121–1128, 2007.

[9] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer vision–ECCV 2014*, pages 818–833. Springer, 2014.

[10] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *CoRR*, abs/1412.6856, 2014.

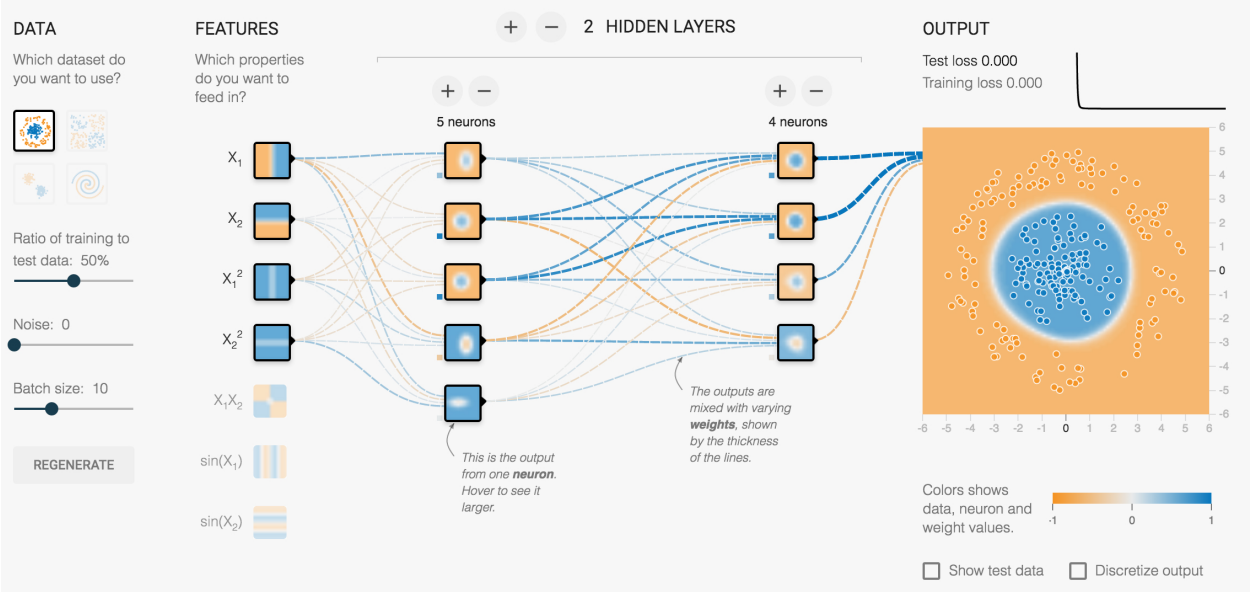


Figure 3: A network architecture with redundant layers and units. Several units in the first hidden layer have already essentially learned to classify the data, as seen by inspecting the in-network activation visualizations.

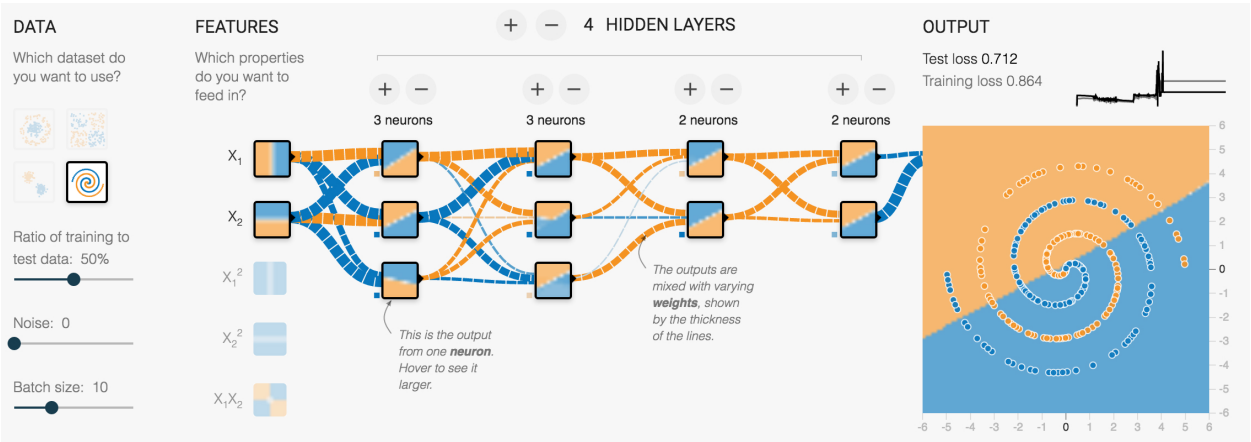


Figure 4: This network has completely failed to classify the data, even after many epochs. The high-contrast activation visualizations and thick weight connections hint at a systemic problem. This diagram was the result of setting the learning rate to the maximum speed.

Clustering with a Reject Option: Interactive Clustering as Bayesian Prior Elicitation

Akash Srivastava
Informatics Forum, University of Edinburgh
10, Crichton St
EH8 9AB, Edinburgh, UK
akash.srivastava@ed.ac.uk

James Zou
Microsoft Research and Stanford University
One Memorial Drive,
Cambridge, MA 02142, USA
jamesyzou@gmail.com

Charles Sutton
Informatics Forum, University of Edinburgh
10, Crichton St
EH8 9AB, Edinburgh, UK
csutton@inf.ed.ac.uk

ABSTRACT

A good clustering can help a data analyst to explore and understand a data set, but what constitutes a good clustering may depend on domain-specific and application-specific criteria. These criteria can be difficult to formalize, even when it is easy for an analyst to know a good clustering when she sees one. We present a new approach to interactive clustering for data exploration, called TINDER, based on a particularly simple feedback mechanism, in which an analyst can choose to reject individual clusters and request new ones. The new clusters should be different from previously rejected clusters while still fitting the data well. We formalize this interaction in a novel Bayesian prior elicitation framework. In each iteration, the prior is adapted to account for all the previous feedback, and a new clustering is then produced from the posterior distribution. To achieve the computational efficiency necessary for an interactive setting, we propose an incremental optimization method over data minibatches using Lagrangian relaxation. Experiments demonstrate that TINDER can produce accurate and diverse clusterings.

1. INTRODUCTION

Clustering is a popular tool for exploratory data analysis. A good clustering can help to guide the analyst to better understanding of the data set at hand. An informative clustering captures not only the properties of the data, but also the goals of the analyst. What makes it challenging to identify a good clustering is that it is often difficult to encode the analyst's goals explicitly as machine learning objectives. Moreover, in many settings, the analyst does not have a well-specified objective in mind prior to encountering the data, but rather continuously updates her goals as she learns more through exploratory analysis. Because the clustering problem is ill-posed, many good clusterings of similar quantitative value exist for a given data set. Even if a clustering algorithm succeeds in finding a quantitatively good clustering, it still may not be what the user qualitatively wanted. Nevertheless, the data analyst may not be able to formalize precisely as a quantitative criterion what differentiates a "good" clustering from a "bad" one. Still, it seems reasonable to expect that the analyst will know a good clustering when she sees one.

This gap between formal clustering criteria and the user's exploratory intuition is the motivation for interactive clustering [9, 3, 24, 6] and alternative clustering approaches [7, 8, 18, 10]. Interactive clustering methods focus on allowing the user to specify precisely how the clustering should be improved, such as by splitting or merging clusters [9, 3]. Although this can be useful, there are other situations in which the analyst can tell that a clustering does not meet her exploratory needs, without having a clear idea of how it should be improved. Alternative clustering methods, on the other hand, produce a set of clusterings which are chosen to be as diverse as possible while still fitting the data. This supports a more exploratory type of data analysis, but many such methods do not scale well to an interactive setting, and sometimes the notion of an alternative is too coarse-grained: An analyst may wish to preserve some parts of a clustering while discarding others.

To allow the user to provide fine-grained "non-constructive" feedback on a clustering, we introduce a simple rejection-based approach to interactive clustering, in which the analyst chooses to reject a subset of clusters and replace them with different ones. This framework contains alternative clustering as the special case in which the user rejects all clusters. The system returns another clustering, which is chosen fit the data as well as possible, while avoiding the creation of any cluster that is similar to the rejected ones. To reflect the notion of "rejecting" a cluster, we call this interaction mechanism TINDER (Technique for INteractive Data Exploration via Rejection).

We formalize this process in a Bayesian framework, in which we view the interaction procedure as a mechanism for prior elicitation. After the user rejects a set of clusters, we modify the prior distribution over model parameters to severely downweight regions of the parameter space that would lead to clusters that are similar to those previously rejected. This prior downweighting is achieved through a mutual information criterion, defined in such a way to prevent the rejection feedback from simply resulting in label permutation. In interactive settings, it is important that the response to the user's feedback be produced quickly, which

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16) August 14th, 2016, San Francisco, CA, USA.

suggests the use of a stochastic method, but unfortunately our penalty function does not decompose into a simple sum over data points. To surmount this, we propose an optimization method that introduces an auxiliary distribution, similar in spirit to variational methods, but that follows a Langrangian duality type argument rather than Jensen's inequality. The resulting objective function can then be optimized using a stochastic coordinate descent algorithm over minibatches of data points, which we show to be efficient in practice.

2. RELATED WORK

Previous work on interactive clustering methods exploits various types of user feedback. One type is *must-link* and *cannot-link* constraints between pairs of data points [24, 5]. Alternately, a second type of feedback is to request that entire clusters be *split* or *merged* [9, 3, 4]. A third type of feedback is for the analyst to explicitly choose the set of features to use in the clustering procedure [6, 11]. Similarly, interactive methods have been proposed for topic models using must-link / cannot-link [1], split/merge [15, 22], and feature-level feedback [17]. While all three types of feedback improve clustering quality, they require that the analyst have a certain level of knowledge about the data set and her information need, which might not be appropriate for a highly exploratory analysis. They can also be quite demanding in requiring active guidance from the user. We are unaware of previous work that uses cluster-level accept/reject feedback like we do.

In contrast, alternative clustering methods [13, 2, 7, 18, 10, 8] focus on generating a set of high-quality clusterings that are chosen to be different from each other, which the user can select between. Work in this area has generated diverse sets of clusters by randomly reweighting features [7], by exploring the space of possible clusterings using Markov Chain Monte Carlo [8], or by penalizing the objective function to encourage clusterings to be diverse [13, 18, 10]. Our framework for interactive clustering includes alternative clustering as a special case, bridging between interactive and alternative clustering. In particular, the objective function that we propose recovers the CAMI method [10] as a special case in which the user always rejects all clusters, and the objective function is optimized jointly over all clusterings, rather than one clustering at a time in response to user feedback. Additionally, the optimization method that we propose (Section 3) is different from the previous work and necessary for obtaining interactive performance.

Our work is similar in motivation to *diverse subset selection*, which is concerned with selecting subsets of data from a collection such that the inter-set diversity and the intra-set diversity is maximized, for example in summarization [14]. The application of diverse k-best summarization presented in [12] is a related problem to alternative clustering, if one considers a set of cluster centroids to be a summary of a data set. Finally, contrastive learning is aimed at fitting a latent variable model so that the latent variables explain the difference between one data set from another, for example, a data set of Chinese news articles versus a dataset of economic articles [25]. Our current work is somewhat analogous to this, in that to support interactive data exploration, we search for different latent variable explanations of a *single* data set.

3. INTERACTIVE CLUSTERING WITH REJECTIONS

Now we describe our rejection-based framework for interactive clustering. We begin with an overview of the interaction method. The data are first clustered according to a standard clustering algorithm. We present this clustering to the analyst for inspection, for example, by displaying the data points or the features that are most closely associated with each cluster. Then if the clustering does not meet the information need of the analyst, she can provide feedback. For each cluster, the analyst can either: (a) **reject** the cluster if it is not relevant to her information need, (b) **accept** the cluster if it is relevant, or else (c) **do neither**, expressing no opinion about the cluster. Once this feedback is complete, we cluster the data again, modifying the objective function of the clustering algorithm to penalize clusters that are similar to rejected clusters, and to reward clusters that are similar to accepted ones. This modified objective encourages the algorithm to return a new clustering that still fits the data well but that respects the user feedback. The process can be repeated as many times as desired. We call each iteration of this process a *feedback iteration*.

Now we describe the clustering method used in TINDER. We formalize the interaction mechanism as a type of Bayesian prior elicitation [21]. At each feedback iteration t, we perform Bayesian clustering with parameters θ, but with a different prior $\pi_t(\theta)$ that strongly downweights parameter vectors that are associated with rejected clusters, and strongly upweights parameters that are associated with accepted clusters. We perform clustering using a standard Bayesian mixture model. Let x denote a single data item, $h \in \{1, \dots K\}$ be a discrete latent variable that indicates the cluster membership of x, and the vector θ denote all of the model parameters, that is, the parameters of the prior distribution $p(h|\theta)$ over clusters, and the parameters of the conditional distribution $p(x|h, \theta)$ of data items given the cluster label. At feedback iteration t, the data is modelled as

$$p_t(x, \theta) = \sum_h p(x|h, \theta)p(h|\theta)\pi_t(\theta). \quad (1)$$

As the subscripting in (1) suggests, the prior distribution will change after every feedback iteration (in a way that we shall discuss in a moment), but the other parts of the probabilistic model will not.

For computational reasons, we perform maximum a posteriori (MAP) estimation. Let $\mathbf{x} = (x_1 \dots x_N)$ denote the data, where x_i is a single data point, and $\mathbf{h} = (h_1 \dots h_N)$ denote an assignment of cluster labels to all data points. Then, at each feedback iteration, MAP estimation computes the parameter estimate $\theta_t = \max_\theta \log p(\theta|\mathbf{x})$ and a soft cluster assignment $p(\mathbf{h}|\mathbf{x}, \theta_t)$ over cluster labels. (Note that this distribution has the same functional form across all iterations, and the parameter θ_t could be different in iteration t.) After reviewing the clustering, the analyst chooses a set of clusters to accept and reject. Let $A_t \subseteq \{1, \dots K\}$ be the indices of the clusters that the user has accepted and $R_t \subseteq \{1, \dots K\}$ be those the user has rejected. The sets A_t and R_t are disjoint. Cluster indices that do not appear in $A_t \cup R_t$ are those clusters for which the analyst has expressed no opinion.

Now we describe how TINDER produces a revised clustering at feedback iteration t. Following a Bayesian framework, we interpret the user feedback from clusterings $0 \dots t-1$ as an indirect source of information about the analyst's prior

beliefs over θ, that she was unable to encode mathematically into the prior distribution. Therefore we define a revised prior distribution $\pi_t(\theta)$ based on all the previous feedback, which is designed in such a way that the resulting clustering, which we denote $p(\mathbf{h}|\mathbf{x}, \theta_t)$, will respect the feedback. The prior $\pi_t(\theta)$ has the form

$$\pi_t(\theta) \propto \pi_0(\theta) \prod_{s=0}^{t-1} \exp\{-\beta f_s(\theta, \theta_s)\},$$

where f_s is a function that measures how well the parameter vector θ respects the feedback (A_s, R_s) from iteration s (lower is better). The parameter β is a temperature parameter.

For example, consider the case of "reject all" feedback, in which the user has rejected all previous clusters, that is, $R_s = \{1, \ldots, K\}$ for all s. This special case has been studied in the literature under the name of alternative clustering (Section 2). In this context, we want f_s to measure the degree of similarity between the cluster distribution $p(\mathbf{h}|\mathbf{x}, \theta)$ and the cluster distribution $p(\mathbf{h}|\mathbf{x}, \theta_s)$ that the user rejected, so that new parameters θ which produce clusters similar to those from θ_s will have lower probability. A naive choice for $f_s(\theta, \theta_s)$ would be to use the negative Kullback-Leibler divergence between the distributions $p(\mathbf{h}|\mathbf{x}, \theta)$ and $p(\mathbf{h}|\mathbf{x}, \theta_s)$. However, in the context of clustering, this metric suffers from the issue of label switching, i.e., merely permuting the cluster assignments can produce high divergence.

Instead, we begin by defining a joint distribution over the individual cluster labels h and h_s that would be assigned by the current clustering and the previous clustering to the same data point x. This joint distribution is

$$p_{\theta,\theta_s}(h, h_s, x) = p(h|x, \theta) p(h_s|x, \theta_s) \tilde{p}(x), \tag{2}$$

where $\tilde{p}(x) = N^{-1} \sum_i \delta_{x,x_i}$ is the empirical distribution over data points, for the Kronecker delta function δ. This now defines a bivariate marginal distribution

$$p_{\theta,\theta_s}(h, h_s) = \frac{1}{N} \sum_{j=1}^{N} p(h|x_j, \theta) p(h_s|x_j, \theta_s) \tag{3}$$

that measures the overall dependence between the two different clusterings, marginalizing out the data. In other words, p_{θ,θ_s} is the joint distribution over pairs of cluster labels that results from randomly choosing a data item x, and clustering it independently according to the distributions $p(h|x, \theta)$ and $p(h_s|x, \theta_s)$. The distribution p_{θ,θ_s} also yields marginal distributions $p_\theta(h) = \sum_{h_s} p_{\theta,\theta_s}(h, h_s)$ and $p_{\theta_s}(h_s) = \sum_h p_{\theta,\theta_s}(h, h_s)$ for each of the individual clusterings, which are simply the prior probabilities of the cluster labels from each clustering.

Now we can define f_s. We begin with the special case of reject all feedback. The distribution $p_{\theta,\theta_s}(h, h_s)$ measures the joint distribution between the new clustering and the previous one at iteration s, so to ensure that these two clusterings are different, we simply minimize their mutual information. This yields

$$f_s(\theta, \theta_s) = I(H; H_s) = \sum_{h=1}^{K} \sum_{h_s=1}^{K} p_{\theta,\theta_s}(h, h_s) \log \frac{p_{\theta,\theta_s}(h, h_s)}{p_\theta(h) p_{\theta_s}(h_s)}. \tag{4}$$

To handle accept feedback, f_s takes a similar form, but the sign is flipped for the clusters in A_s, so that f_s encourages similarity rather than dissimilarity. More specifically:

$$f_s(\theta, \theta_s) = \sum_{h_s \in R_s} \sum_{h=1}^{K} p_{\theta,\theta_s}(h, h_s) \log \frac{p_{\theta,\theta_s}(h, h_s)}{p_\theta(h) p_{\theta_s}(h_s)} - \sum_{h_s \in A_s} \sum_{h=1}^{K} p_{\theta,\theta_s}(h, h_s) \log \frac{p_{\theta,\theta_s}(h, h_s)}{p_\theta(h) p_{\theta_s}(h_s)} \tag{5}$$

Note that clusters for which the user has said "no opinion" are in neither A_s nor R_s, and therefore such clusters have no effect on π_t, and hence no effect on the clustering in subsequent feedback iterations. This completes the definition of π_t. Now, to compute the revised clustering, we perform MAP estimation on (1), which is equivalent to maximizing

$$L_t(\theta) = \sum_{j=1}^{N} \log p(x_j|\theta) - \beta \sum_{s=1}^{t-1} f_s(\theta, \theta_s) + \log \pi_0(\theta), \tag{6}$$

where β can now be interpreted as a weighting parameter to bring the terms to a common scale. Denote by θ_t the new MAP parameter estimate, i.e., $\theta_t = \max_\theta L_t(\theta)$. Then the new clustering that is displayed to the analyst is based on the soft assignment $p(\mathbf{h}|\mathbf{x}, \theta_t)$.

Examples..

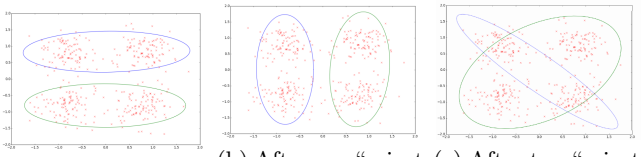
As an illustrative example, consider the 2D dataset shown in Figure 1(a), which is generated from a mixture of four isometric Gaussians. The ellipses in the figure show the clustering resulting from maximizing the likelihood of a mixture of two Gaussians using expectation maximization (EM) in the zeroth feedback iteration of TINDER. Starting from here, suppose the user rejects both clusters. Then Figure 1(b) shows the resulting clustering that TINDER generates in the next feedback iteration. Rejecting both clusters again results in the clustering in Figure 1(c). Therefore using TINDER, an analyst can obtain three quantitatively different explanations of the data in three feedback iterations.

Our per-cluster feedback framework recovers alternative clustering, in which the goal is to as explore as many diverse clusterings as possible, as the special case in which all previous clusters are rejected. By providing more specific feedback, the user can perform a more directed style of exploration, in which the user guides the clustering procedure toward a partitioning that interests her. By incorporating both alternative clustering and the more directed style of per-cluster feedback in the same framework, TINDER allows the analyst to flexibly alternate between more exploratory and more directed navigation through the space of possible clusterings.

Although we have described TINDER as a clustering method, that is, where $p(x, h|\theta)$ is a mixture model, the same logic can be applied to more general graphical models, e.g., ones that contain other latent variables in addition to x and h. All we require is that the model contains a discrete latent variable that we can use in the same way as the cluster labels h, and that MAP estimation of θ be tractable. We leave further exploration of this idea to future work.

Optimization.

In this section we discuss how to perform MAP estimation of θ, i.e., to efficiently optimize L_t. The gradient of L_t is easy to compute, so it is possible to apply standard optimization algorithms like conjugate gradient. However, for



(a) Initial clustering (b) After one "reject all" (c) After two "reject all"

Figure 1: Example of TINDER clusterings produced in three feedback iterations on synthetic data, showing (a) the initial clustering from expectation maximization (EM), and after (b) one round and (c) two rounds of "reject all clusters" feedback from the user.

an interactive algorithm, each feedback iteration needs to be relatively fast, because the computation is run while the user is waiting. To achieve interactive performance on large data sets, we would therefore prefer a stochastic gradient style of algorithm, in which each update to the parameters only depends on a small subset of data points. But the form of $f_s(\theta, \theta_s)$ makes this difficult. Notice that the distribution $p_{\theta,\theta_s}(h, h_s)$ contains a summation over data points within it, and this appears inside a log within f_s. Therefore the gradient of L_t does not decompose into a simple sum over data items.

Alternately, we could optimize L_t using the standard EM algorithm for MAP estimation. Recall that in this algorithm, the E step is unchanged from maximum likelihood, but the M step contains the log prior distribution as part of the objective. Applying this to L_t, the M step becomes

$$\theta_t \leftarrow \max_{\theta} \sum_j \sum_h q_j(h) \log p(h, x_j|\theta) - \beta \sum_s f_s(\theta, \theta_s) + \log \pi_0(\theta),$$

where q_j is the standard EM auxiliary distribution. It is not clear that this objective is any easier to optimize than (6), nor is it clear how to derive a stochastic gradient-style algorithm.

Instead we take a different approach, inspired by Lagrangian relaxation. To simplify the exposition, we will describe the optimization algorithm only for the case of "reject all" feedback, but the extension to the other types of feedback is straightforward. First we introduce an auxiliary random variable H, whose output is a cluster assignment, and whose distribution is given by a variational distribution $q_j(h)$ for each data point x_j. As in (3), we can induce a joint distribution over the random variable H and the random variable H_s whose distribution is given by $p(h_s|x_j, \theta_s)$. This joint distribution is

$$p_{q,\theta_s}(h, h_s) = N^{-1} \sum_j q_j(h) p(h_s|x_j, \theta_s).$$

Notice that this distribution, and therefore the resulting mutual information, which we denote $I_q(H; H_s)$, is a function of the variational distribution q. Then optimizing (6) is equivalent to

$$\max_{\theta,q} \log p_\theta(x) - \beta \sum_s I_q(H; H_s) + \log \pi_0(\theta). \quad (7)$$
$$\text{s.t. } \mathrm{KL}(q_j \,\|\, p(h|\theta, x_j)) = 0 \quad \forall j \in \{1, 2, \ldots N\},$$

where KL indicates the Kullback-Leibler divergence. Incorporating the constraint using a penalty term with parameter α leads to

$$\max_{\theta,q} \log p_\theta(x) - \beta \sum_s I_q(H; H_s) \quad (8)$$
$$- \alpha \sum_j \mathrm{KL}(q_j \,\|\, p(h|\theta, x_j)) + \log \pi_0(\theta).$$

If α is large enough, then the solution of (8) will be the same as for (7). Coordinate descent on (8) yields the EM-like algorithm:

"E"-Step:

$$q \leftarrow \max_q -\beta \sum_s I_q(H; H_s) - \alpha \sum_j \mathrm{KL}(q_j \,\|\, p(h|\theta, x_j)) \quad (9)$$

"M"-Step:

$$\theta \leftarrow \max_\theta E[\log p(\mathbf{x}, \mathbf{h}|\theta)]_q \quad (10)$$

This is not strictly an EM algorithm, because we lose the lower bound property that would have arisen if we had applied Jensen's inequality. However, if at the end of optimization procedure, we have that $q_j(h) = p(h|x_j, \theta)$ for all j (which will happen if α is set high enough, and can be easily checked), then θ is a local maximum of L_t.

Now we can optimize the objective in the "E" step by coordinate descent. The mutual information $I_q(H; H_s)$ still depends on all of the data points via $p_{q,\theta_s}(h, h_s)$, but now if we perform stochastic coordinate descent on each distribution q_j, then the value of $p_{q,\theta_s}(h, h_s)$ can be updated incrementally, so recomputing $I_q(H; H_s)$ does not require iterating through the entire data set. The "M" step is very fast, as it is exactly the same as the M step in the EM algorithm for maximum likelihood.

4. EXPERIMENTS

In this section, we evaluate the diversity and the quality of the clusterings produced by TINDER. Following previous work in alternative and interactive clustering [9, 3, 24, 6, 7, 8, 18, 10], we present an automatic evaluation in which we measure the quality of clusterings by comparing how well the clusters correspond to gold standard labels. An automatic evaluation allows us to compare the output of the learning algorithms directly without dealing with difficult and potentially confounding aspects of user interface design.

We evaluate TINDER for both per-cluster feedback (TINDER: Per Cluster), in which the user feedback is attempting to drive the system toward a given clustering, and in global mode (TINDER: Global), which is the alternative clustering setting in which the user is exploring the data set by rejecting all clusters. To replicate per-cluster feedback within an automatic evaluation, we simulate a user using the following heuristic. At each feedback iteration, the user provides feedback on one cluster at a time. If the cluster purity is below 50% with respect to the gold standard labels, the simulated user rejects the cluster otherwise the cluster is accepted. If none of the clusters are above this threshold, then the entire clustering is rejected. The reasoning here is that we are simulating a user whose information need is to find a clustering similar to that defined by the gold standard labels.

We compare TINDER to the popular *Decorrelated-kMeans* (Dec-kMeans) [18] algorithm for alternative clustering, which uses a penalized k-means objective to encourage the centroids from the previous clustering to be orthogonal to those from

the current clustering. Since this method in the default setting produces just two clusterings, we extended it by adding additional error and penalty terms to produce more than two clusterings at a time. We also compare to running EM using different random initializations, which will produce different clusterings because of its sensitivity to initialization. We call this method *random restarts*.

We use two data sets: a small collection of 640 face images of 20 people in different orientations from the CMU face dataset [20] and a large collection of 10,000 thumbnail images from CIFAR10 [19]. The CIFAR10 data is significantly larger than other data sets that have been used in the alternative clustering literature. The CIFAR10 data set is labeled with 10 classes. In the CMU face dataset, each image has three different types of labels: the identity of the person in the image, their gender, and the pose (orientation of the face). This provides three natural clusterings of the data. To obtain features, for the CIFAR10 dataset we use the the embedding generated by training the VGG network [23] on the CIFAR10 training set. For the CMU face dataset, we apply PCA to the raw pixel values and retain 90% of the variance from the original data.

To evaluate diversity, the Adjusted Rand Score (ARS) is used to measure the distance between two clusterings [16]; an ARS of 0 indicates no association between a pair of clusterings, and a score of 1 indicates a perfect match between the clusterings. To measure the diversity of a set of clusterings, we average the pairwise ARS over all pairs of clusterings in the set. To evaluate the quality of a clustering, we report its purity with respect to a set of ground truth labels. To evaluate the quality of a set of clusterings, we report the maximum purity of any clustering the set, reflecting the idea that, after examining a set of different clusterings, an analyst can choose the single clustering that she finds most useful.

We use a mixture of Gaussians (GMM) for modeling the CMU Face and the CIFAR10 datasets. In both cases, for the zeroth feedback iteration we set $\pi_0(\theta)$ to be one. We reabsorb the relaxed Lagrange multiplier α from Section 3 in β as well. Empirically, we found that TINDER performs well by simply setting β such that the penalty term, $\beta \sum_s f$, and the log-likelihood have the same order of magnitude. Dec-kMeans also has a similar weighing parameter λ, which we set according to the guidelines provided by the original authors [18].

All methods are allowed the same number of feedback iterations, i.e., all methods are evaluated on the same number of clusterings. To ensure this, first we run TINDER in per-cluster mode until the clustering stabilizes, that is, until the simulated user accepts all clusters. Then we run each of the other methods, including TINDER Global, for the same number of feedback iterations that was required by TINDER: Per Cluster. All methods are repeated 20 times from different random initializations, and we report the average maximum quality and the average diversity over the repetitions. For the CMU face data set, we use $K = 8$ clusters, whereas for CIFAR10, we use $K = 10$ clusters.

4.1 Results

Table 1 summarizes the clustering quality of the methods. The three columns for the CMU Face data report quality with respect to each of the three different types of gold standard labels. For TINDER: Per Cluster, the feedback from the simulated user is based on same set of gold standard

(a) Clustering 0 (no feedback)

(b) Clustering 1 (after one feedback iteration). For space, only two of the ten clusters are shown.

(c) Clustering 2 (after two feedback iterations). For space, only one of the ten clusters is shown.

(d) Clustering 5 (after five feedback iterations). For space, only four of the ten clusters are shown.

Figure 2: Example of TINDER clusterings on the CIFAR10 dataset.

labels that is used to evaluate quality; that is, the goal is to measure the effectiveness of per-cluster feedback at reaching a specific clustering that the user discovers during exploration. We find that TINDER: Per Cluster outperforms or matches the other methods, indicating that the more specific guidance provided by per-cluster feedback indeed leads to higher quality clusterings. On average TINDER requires four feedback iterations to stabilize. The quality of the clusters from the other three methods are similar to each other.

We report the clustering diversity in Table 2. For both the datasets, TINDER: Global clearly returns a much more diverse set of clusterings, outperforming both the baseline methods by a significant margin. On CIFAR10, Dec-kMeans oscillates between two similar clusterings and as a result performs worse than random restarts. These results indicate that overall, TINDER: Global returns a more diverse set of clusters of equivalent quality to the other alternative clustering methods. As expected, TINDER: Per Cluster results are not as diverse, because the goal of per-cluster feedback is to drive the method towards a specific target clustering, which necessarily reduces diversity.

To illustrate the effect of the feedback, we display in Figure 2 some of the clusters from TINDER: Global on the CIFAR10 dataset. TINDER: Global clusterings are not just able to find all the original CIFAR10 clusters but other meaningful clusters as well. In the figure, each of the rows represents a cluster and shows the top 6 images from that cluster ordered by their likelihood under the cluster. Figure 2(a) shows the initial clustering for $K = 10$ with no feedback. Clustering 1 (Figure 2(b)) is produced by TINDER after a single iteration of "reject all" feedback. We see that Clusters 2 and 3 from Clustering 0 (which contain deer and horses, respectively) are replaced in Clustering 1 by clusters 2' and 3', which contain

Table 1: Clustering quality, measured by purity to ground truth labels (higher is better).

	CIFAR10	CMU Face Person	CMU Face Gender	CMU Face Pose
Random Restarts	0.89	0.37	0.87	**0.44**
Dec-kMeans	0.90	0.37	0.86	0.42
TINDER: Global	0.89	0.37	0.89	0.40
TINDER: Per Cluster	**0.93**	**0.39**	**0.93**	**0.44**

Table 2: Diversity of returned sets of clusterings, measured by Adjusted Rand Score (lower is better).

	CIFAR10	CMU Face
Random Restarts	0.56	0.55
Dec-kMeans	0.83	0.38
TINDER: Global	**0.15**	**0.27**
TINDER: Per Cluster	0.88	0.59

large animals (Cluster 2') and horses with riders (Cluster 3'). The result of the next feedback iteration is shown in Clustering 2 (Figure 2(c)). We see that Cluster 9 has been replaced by Cluster 9', which contains images of birds and planes, which were scattered over multiple clusters in Clustering 0. Finally, after five feedback iterations, Clustering 5 (Figure 2(d)) includes clusters of ships (Cluster 4'), cats (Cluster 5'), birds (Cluster 6') and planes (Cluster 9"), which did not exist in Clustering 0. These new clusters replace Clusters 4-6 and 9 from Clustering 0, which have low purity.

As for running time, each feedback iteration of TINDER requires a few seconds for the CMU Face data set and under a minute for CIFAR10. Our implementation of Dec-kMeans performs comparably. Both methods take the same amount of time as standard EM without feedback. Finally, we observe that TINDER can be easily applied to any mixture model, not just a mixture of Gaussians. To demonstrate this, we also applied TINDER to a mixture of multinomials model for text data (see supplementary material).

5. CONCLUSION

In this paper we have presented a method for interactive clustering based on a particularly simple feedback mechanism, in which an analyst can reject individual clusters and request new ones. The interaction is formalized as a method of prior elicitation in a Bayesian model of clustering. We showed the efficacy of this method on two real world datasets. An interesting direction of future work would be to extend our approach to other graphical models for data exploration, such as topic models.

6. REFERENCES

[1] D. Andrzejewski, X. Zhu, and M. Craven. Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In *International Conference on Machine Learning (ICML)*, pages 25–32. ACM, 2009.

[2] E. Bae and J. Bailey. COALA: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *IEEE International Conference on Data Mining (ICDM)*, pages 53–62, 2006.

[3] M.-F. Balcan and A. Blum. Clustering with interactive feedback. In *Algorithmic Learning Theory*, pages 316–328. Springer, 2008.

[4] M.-F. Balcan, A. Blum, and S. Vempala. A discriminative framework for clustering via similarity functions. In *ACM Symposium on Theory of Computing*, pages 671–680. ACM, 2008.

[5] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *SIAM International Conference on Data Mining (SDM)*, volume 4, pages 333–344, 2004.

[6] R. Bekkerman, H. Raghavan, J. Allan, and K. Eguchi. Interactive clustering of text collections according to a user-specified criterion. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 684–689, 2007.

[7] R. Caruana, M. Elhawary, N. Nguyen, and C. Smith. Meta clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 107–118. IEEE, 2006.

[8] Y. Cui, X. Z. Fern, and J. G. Dy. Learning multiple nonredundant clusterings. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(3):15, 2010.

[9] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *ACM SIGIR conference on Research and Development in Information Retrieval*, pages 318–329. ACM, 1992.

[10] X.-H. Dang and J. Bailey. Generation of alternative clusterings using the CAMI approach. In *SIAM International Conference on Data Mining (SDM)*, 2010.

[11] S. Dasgupta and V. Ng. Which clustering do you want? inducing your ideal clustering with minimal feedback. *Journal of Artificial Intelligence Research*, 30:581–632, 2010.

[12] J. A. Gillenwater, R. K. Iyer, B. Lusch, R. Kidambi, and J. A. Bilmes. Submodular hamming metrics. In *Advances in Neural Information Processing Systems*, pages 3123–3131. 2015.

[13] D. Gondek and T. Hofmann. Non-redundant data clustering. In *IEEE International Conference on Data Mining (ICDM)*, 2004.

[14] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems*, pages 2069–2077, 2014.

[15] Y. Hu, J. Boyd-Graber, and B. Satinoff. Interactive topic modeling. In *Association for Computational Linguistics*, 2011.

[16] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[17] J. Jagarlamudi, H. Daumé III, and R. Udupa. Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European*

Chapter of the Association for Computational Linguistics, pages 204–213. Association for Computational Linguistics, 2012.

[18] P. Jain, R. Meka, and I. S. Dhillon. Simultaneous unsupervised learning of disparate clusterings. *Statistical Analysis and Data Mining*, 1(3):195–210, 2008.

[19] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images, 2009.

[20] T. M. Mitchell. Machine learning. *Computer Science Series (McGraw-Hill, Burr Ridge, 1997) MATH*, 1997.

[21] A. O'Hagan, C. E. Buck, A. Daneshkhah, J. R. Eiser, P. H. Garthwaite, D. J. Jenkinson, J. E. Oakley, and T. Rakow. *Uncertain judgements: eliciting experts' probabilities*. John Wiley & Sons, 2006.

[22] Q. Pleple. Interactive topic modeling. Master's thesis, University of California, San Diego, 2013.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning (ICML)*, pages 577–584, 2001.

[25] J. Y. Zou, D. J. Hsu, D. C. Parkes, and R. P. Adams. Contrastive learning using spectral methods. In *Advances in Neural Information Processing Systems*, pages 2238–2246, 2013.

Interacting with Massive Behavioral Data

Shih-Chieh Su
Qualcomm Inc.
5775 Morehouse Drive
San Diego, CA 92121 USA
shihchie@qualcomm.com

ABSTRACT

In this short paper, we propose the split-diffuse (SD) algorithm that takes the output of an existing word embedding algorithm, and distributes the data points uniformly across the visualization space. The result improves the perceivability and the interactability by the human.

We apply the SD algorithm to analyze the user behavior through access logs within the cyber security domain. The result, named the topic grids, is a set of grids on various topics generated from the logs. On the same set of grids, different behavioral metrics can be shown on different targets over different periods of time, to provide visualization and interaction to the human experts.

Analysis, investigation, and other types of interaction can be performed on the topic grids more efficiently than on the output of existing dimension reduction methods. In addition to the cyber security domain, the topic grids can be further applied to other domains like e-commerce, credit card transaction, customer service to analyze the behavior in a large scale.

CCS Concepts

•Human-centered computing → Visual analytics; Information visualization;

Keywords

data visualization, human interaction, dimension reduction, risk management

1. INTRODUCTION

When there are multiple measures of the each sample, the data is described in the a high dimensional space $\mathcal{H}$ by these measures. To make these high dimensional data points visible to human, a word embedding (or dimension reduction) technique is employed to map the data points to a lower dimensional space $\mathcal{L}$. Usually $\mathcal{L}$ is a two-dimensional (2D) or three-dimensional (3D) space. The word embedding technique of choice attempts to preserve some relationship among the data points in $\mathcal{H}$ after mapping them to $\mathcal{L}$.

For example, the multi-dimensional scaling (MDS) [5] $\mathbb{M}$ tries to preserve the distance between data points, during the mapping from $\mathcal{H}$ to $\mathcal{L}$. The stochastic neighbor embedding (SNE) [6] type of algorithms further emphasize the local relationship ahead of the global relationship. There are other dimension reduction techniques putting emphasis on different favored metrics over relationship. On a specific situation, one particular dimension reduction technique could be more suitable or more efficient than others.

The output from existing dimension reduction algorithm is a set of data points that are non-uniformly scattered around the visualization space, which has some drawbacks:

1. Some data points may overlap with others. Overlap makes the information less perceivable.

2. The data points are denser in some area. The heterogeneity makes human interaction with the data points more difficult.

2. METHODS

In order to better utilize the visualization space, we proposed to distribute the data points evenly over the visualization space. The cloud of data points is deformed in the same space defined by the dimension reduction algorithm of choice. This deformation is denoted as $\mathbb{S}$. In the meanwhile, it is desirable to preserve the point-wise relationship maintained by the dimension reduction algorithm. Our strategy in approaching this goal is prioritized as follows:

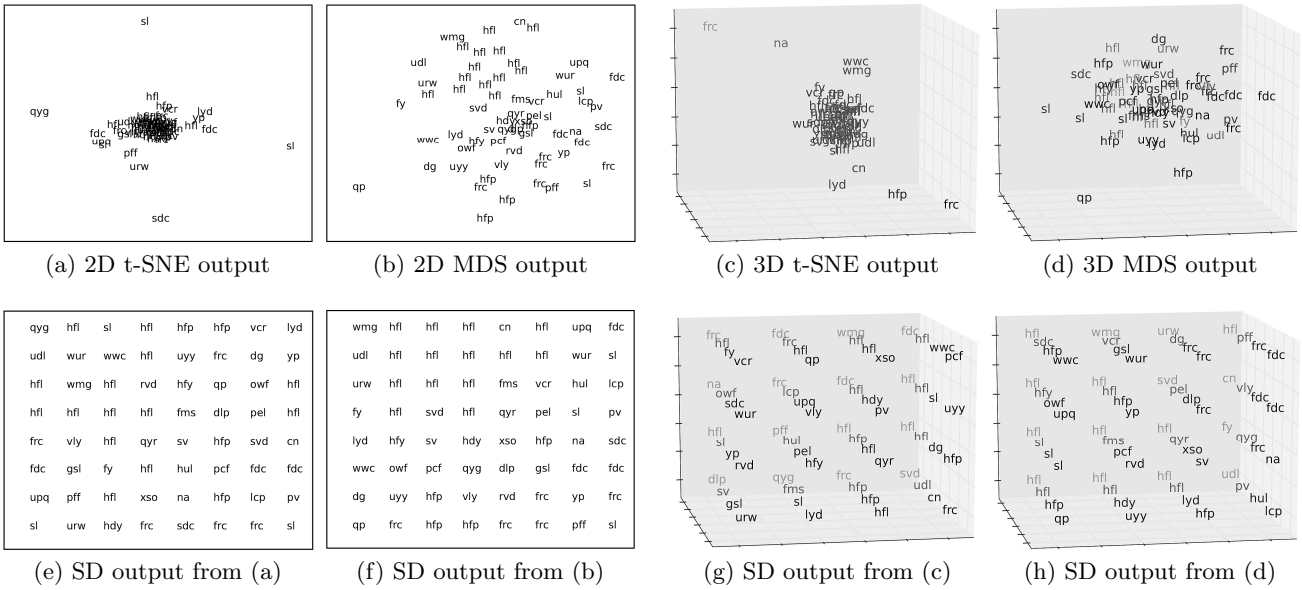
1. Points are equally spaced after the mapping $\mathbb{S}$.

2. Point-wise topology is preserved. $\mathbb{S}$ attempts to keep point p_j on the same side of point p_i as before the mapping.

3. Point-wise geometry is loosely followed. When p_i is far from p_j, $\mathbb{S}(p_i)$ is far from $\mathbb{S}(p_j)$.

The algorithm we propose is called the split-diffuse (SD) algorithm (Algorithm 1), which follows the strategies above. In our implementation, the SD algorithm first picks the x-axis as the dimension to split. As in Figure 2 (a), it splits the data points into two groups: the ones smaller than or equal to the median, and the ones larger than the median. Each group goes through this split step again over the y-dimension, as in Figure 2 (b). We recursively split the points

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16) August 14th, 2016, San Francisco, CA, USA.



(a) 2D t-SNE output (b) 2D MDS output (c) 3D t-SNE output (d) 3D MDS output

(e) SD output from (a) (f) SD output from (b) (g) SD output from (c) (h) SD output from (d)

Figure 1: The split-diffuse (SD) algorithm takes the output of any dimension reduction technique and distributes the data points evenly while maintaining the topology among them. Example inputs of 64 data points from (a) 2D t-SNE, (b) 2D MDS, (c) 3D t-SNE and (d) 3D MDS are distributed evenly in the same space as shown in (e)-(h) respectively.

Algorithm 1 Split-diffuse algorithm (square of power of 2)

Input: data points $\{p\}$ of length $2^h \times 2^h$, depth $d = 0$, allocation string $c =$ ''

split-diffuse ($\{p\}$, d, c)

  $k \leftarrow$ length of $\{p\}$

  if $k = 1$, **then**

    resolve $\mathbb{S}(p)$ from c

    return p

  end if

  $a \leftarrow mod(\text{depth}, 2)$

  $m \leftarrow$ median of $\{p\}$ in the dimension a

  return ([**split-diffuse** ($\{p : p \leq m|_{dim=a}\}$, d+1, c+'L')],
  [**split-diffuse** ($\{p : p > m|_{dim=a}\}$, d+1, c+'R')])

(a) first-level split (b) second-level splits

Figure 2: The split-diffuse algorithm over the 4×4 layout.

in x- and y-dimension iteratively, until there is only one point in current recursion.

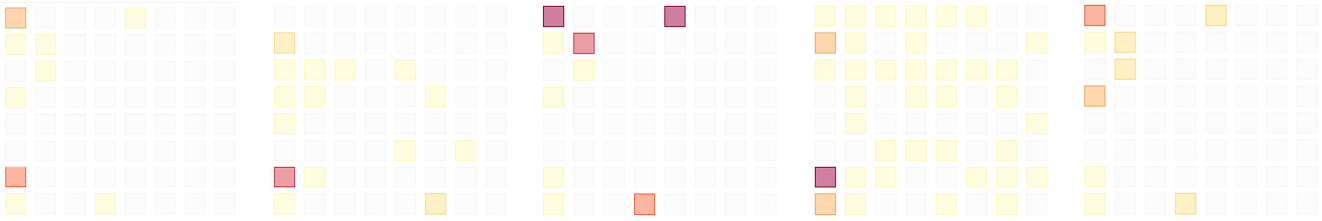
We keep track of the splitting path in string c. At the end of the recursion, the placement each single point p is resolved. The indexes of the SD-mapped points, $\mathbb{S}(p)$, are all integers, and forms a $2^h \times 2^h$ array. This means that the mapped data points are equally spaced in a $2^h \times 2^h$ square. To achieve this uniformity in the space $\mathcal{L}$, the data points are essentially diffused from the denser area to the coarser area by the SD algorithm — hence the name split-diffuse.

Some sample outputs from existing dimension reduction techniques are shown in Figure 1, as well as the corresponding SD outputs. Although we only present the results from t-SNE and MDS, the SD algorithm can be applied to outputs of other techniques such as the principal component analysis (PCA) [3], isomap [4], spectral embedding [1], and totally random trees embedding [2].

3. INTERACTING WITH THE DATA

The motivation to better utilize the visual space comes from the need to interact with massive amount of data. Consider the case that there are millions of items being shipped to a city every month. How can we easily observe the difference in the monthly shipping patterns? When vectoring each item as a data point, putting all the data points on a chart makes the chart hard to read. Instead, using clustering algorithms to group the points and showing the representatives is a better way to present the shipping pattern. Still, with the existing dimension reduction techniques (Figure 1 (a)-(d)), it is difficult to visually compare the difference and interact with the representative points for more detail.

In our use case, we apply the SD algorithm to help analyzing behavioral content in the cyber security domain. The goal of the system is to detect behavioral anomaly based on the access logs. Topics are generated on the content of the logs in a word vector space of $19K+$ dimensions. MDS is applied to reduce the dimension. As shown in Figure 1



(a) current activities of an entity (b) historical activities of this entity (c) risk against the historical activities of this entity (d) historical activities of the peers (e) risk against the historical activities of the peers

Figure 3: The topic grids. The self risk in (c) is derived from comparing the current activities (a) and the historical activities (b) of a specific entity. The peer risk in (e) is derived from comparing the current activities (a) and the peers' activities (d) of a specific entity.

(b) and (f), topics are represented by the most relevant keywords, encrypted. Topics close to each other may share the same representative keyword. The SD algorithm follows to generate the topic grids and visualize different metrics about the behavior of a user (Figure 3).

When not directly displaying the detail keywords about a topic, the topic grids requires less space. At the same time, the human expert still can easily keep track of the topics based on their indexes over all dimensions and compare the difference between different sets of topic grids. Human interaction, which is the ultimate goal of the uniform placement of the data points, can be done more easily on the topic grids than on the raw dimension reduction output as in Figure 1 (a)-(d). For example, the mouse over event on a grid pops up the topical summary, and the click event to overlay the detailed topical activities.

It is also useful to monitor the behavior change over time. In such cases, we reserve a dimension in $\mathcal{L}$ as the time axis. For a 2D space $\mathcal{L}$, a 1D version of SD algorithm is applied to maintain the point-wise topology. The cumulative activities have a shape of curtain. Meanwhile, we can pile up the 2D topic grids on the time axis over the 3D $\mathcal{L}$, as shown in Figure 4. With normal or usual behavior, it is expected to see the consistent hot grids at the same locations over time.

4. FUTURE WORK

In addition to the cyber security domain, the topic grids can be applied to other domains having free-form text logs to analyze the behavior described by the logs. Some possible use cases include e-commerce, credit card transaction, customer service, or others with large volume of behavioral data to be analyzed.

It is also possible to apply the topic grids to the structured data, on which an arbitrary clustering algorithm can generate cluster centers. The data points are then organized into these cluster centers, the same way we use the topic to represent the log entries related to it.

5. REFERENCES

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[2] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

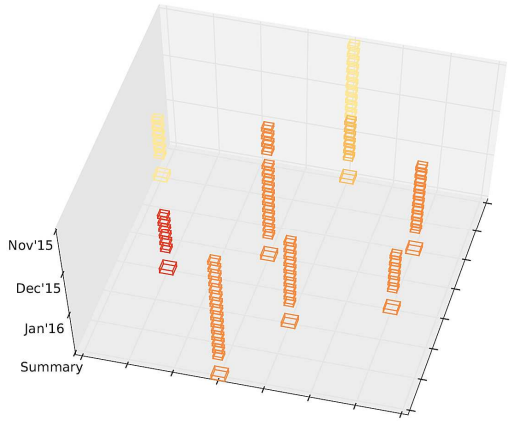
[3] K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[4] J. B. Tenenbaum, V. D. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[5] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[6] L. Van der Maaten and G. E. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

(a) Topic curtain



(b) Topic shower

Figure 4: Other formats of the topic grids

VIT-PLA: Visual Interactive Tool for Process Log Analysis

Sen Yang[1], Xin Dong[1], Moliang Zhou[1], Xinyu Li[1], Shuhong Chen[1], Rachel Webman[3], Aleksandra Sarcevic[2], Ivan Marsic[1] and Randall S. Burd[3]

[1]Rutgers University
Piscataway, NJ, USA
{sy358, xd48, mz330, xl264, sc1624, marsic}@rutgers.edu

[2]Drexel University
Philadelphia, PA, USA
aleksarc@drexel.edu

[3]Children's National Medical Center
Washington, DC, USA
{rwebman, rburd}@childrensnational.org

ABSTRACT
Techniques for analyzing and visualizing process or workflow data have been developed and applied in a wide range of domains. Visual analysis of large process logs and integration of statistical analysis, however, have been limited. We introduce the Visual Interactive Tool for Process Log Analysis (VIT-PLA) that provides a simplified process log visualization and performs statistical correlation analysis on process attributes. We demonstrate its use by applying it to an artificial dataset and running a preliminary analysis of trauma team task data collected from a medical emergency department.

Keywords
Interactive Workflow Data Visualization; Trace Alignment; Trace Clustering; Correlation Analysis

1. INTRODUCTION

1.1 Motivation
Many contemporary information systems record activity logs, including personal calendars and electronic health records (EHR). Process mining techniques attempt to extract non-trivial knowledge and insights from these activity logs and use them for further analyses [1]. Most research in process mining has focused on workflow discovery and process execution visualization [1][2]. Whe visualized, real-world workflow often produces "spaghetti-like" graphics that are difficult to analyze and do not provide useful observations or insights. In addition to graphical visualization, other efforts have also been made to produce different visualizations for process executions or workflow data [3][4][5][6][7][8][9]. Although these systems have been shown to work well with focused processes and relatively small event logs, little work has been done with large process logs with many execution traces (typically hundreds or thousands of different process cases). Simply displaying all traces at once does not make a useful visualization. We observed that only several dozen traces can fit intelligibly on one screen at a time. Even if the symbols were distinguishable, the amount of displayed data make it inconvenient for human interpretation. When working with large workflow datasets, it is often useful to obtain a concise visualization that summarizes the data into an easily interpretable format. We present an approach for visualizing a summary of large process logs by aggregating the data with a trace clustering method. Process traces are clustered based on the similarity or proximity between their elements (i.e. process tasks). Each cluster is represented using a "representative" or "average" trace extracted from the corresponding cluster. Using this approach, we are able to usefully visualize large process logs. To help users better understand the clusters, we also included tools for running statistical tests on the clusters and their associated process attributes. These statistical test results can reveal significant and interesting correlations between process executions and process attributes. We implemented these approaches in a Java-based application, named VIT-PLA.

KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16), August 14th, 2016, San Francisco, CA, USA.

1.2 Related Work
Recent advances have been made in the development of workflow data visualization techniques. EventFlow [3] visualizes temporal events on a timeline and can simplify workflow executions into an aggregated display. Outflow [5] aggregates events into a graph with integrated statistics. Frequence [6] and Care Pathway Explorer [7] are user interfaces for information exploration that integrate interactive visualizations with data mining to find frequent event sequence patterns. Dotted Chart [8] uses colored dots to visualize process traces in a fast and simple implementation. The trace alignment plugin for the ProM framework [9] is designed to align process traces so as to optimize interpretability and facilitate exploration. Despite extensive work on interactive visualization, little has been done to directly integrate statistical analysis into these applications. Some data visualization applications can show general statistics [5][8], but few can provide more sophisticated ones [4]. CoCo [4] can be used to find similarities and differences between two groups ("cohorts") of process traces and to highlight their significant distinguishing features (e.g. activity order, frequency, and duration).

From the perspective of workflow visualization, Eventflow [3] and ProM's Trace Alignment [9] plugin are closest related to our work. Neither are suitable for visualizing large process logs with many traces, because both visualize all activities in the log at once. Without data aggregation and summarization strategies, the size of the dataset that can be handled is always limited. From our previous experience with Eventflow and ProM, visualizations using a standard sized computer monitor (24") generally become uninterpretable when the number of unique process traces exceeds 100. EventFlow can be used to visualize logs with >100 process traces, but only if there are many repeated traces [21]. Eventflow visualizes the activities on a timeline without advanced processing of the data. ProM visualizes the alignment and also clusters the

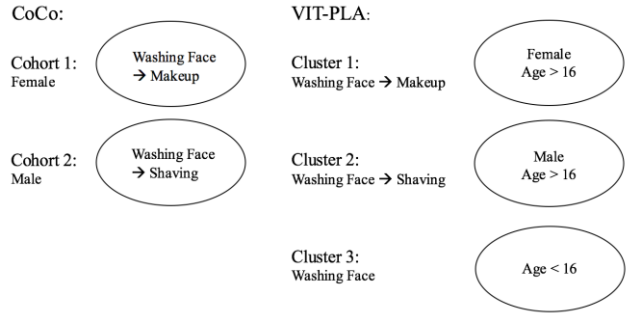


Figure 1. A simple example showing the differences between the statistical analysis in CoCo and VIT-PLA. This example describes a morning skincare ritual. The workflow includes three different activities (washing face, makeup, and shaving) and two different attributes (gender and age).

process traces, but does not provide any statistical analyses that can help the user better understand their data. When visualizing clusters of process traces, ProM shows all traces in each cluster without any data aggregation or simplification. In contrast, our approach displays each cluster's cluster "prototype" [8], i.e., an execution trace that is representative of the other traces in the cluster (the representative trace is not necessarily one of the original process traces in the input log). This strategy enables visualization of large process logs. This visualization also helps to identify key characteristics of each cluster and key differences between clusters.

From the perspective of statistical analysis, CoCo is closest related to our work. Both CoCo and VIT-PLA seek to correlate trace structural features (e.g., sequential order of activities, their frequencies and durations) with process attributes (e.g., patient gender, age, etc.). The two approaches to statistical analysis are different (Figure 1). CoCo first splits the data into strictly two cohorts based on a background attribute (in this case gender). It then finds significant associations between the cohorts' trace structures and attributes. It may identify a structural pattern (e.g., "Washing Face → Makeup") as significantly belonging to one cohort (female), as opposed to the opposite (male). In contrast, our implementation first separates the data into clusters based on trace structure, and then associates cluster membership with background attributes. For example, the sequence "Washing Face → Makeup" is executed mostly by females over age 16.

Unlike CoCo that can only make these associations based on cohort pairs, our system uses multinomial or binomial logistic regression to make associations based on multiple clusters. VIT-PLA allows for more comprehensive attribute-structure correlation, bringing the previously unusable age attribute into the analysis (see example above [Figure 1]). In this way, VIT-PLA's approach reveals potential relationships missed by CoCo's binary analysis.

Our statistical analysis is important because it facilitates the discovery of significant correlations between clusters and background attributes. Given the trace attributes, we may determine what workflow practices (represented by the cluster prototype) are more likely to be observed, which is useful information for analyzing the workflow data and extracting insights.

1.3 Contribution

Our main contribution is a novel approach to producing summarized visualizations of large process logs and directly integrating statistical analyses into the visualization. These features help users discover attributes associated with specific sequence progressions and deviations within the dataset.

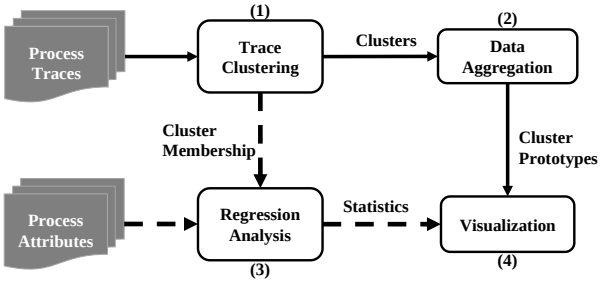


Figure 2. Flowchart outlining the core methods implemented in VIT-PLA and their corresponding inputs and outputs.

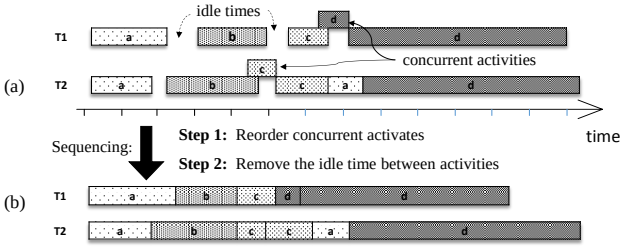


Figure 3. Two steps of sequencing the traces with concurrent activities (such as *d* in T_1 and *c* in T_2) and idle times (white spaces between activities). (a) Example process traces before sequencing. (b) The same process traces after sequencing.

The paper is organized as follows. Section 2 introduces our approach to process trace visualization and attribute analysis. Section 3 discusses our implementation and user interface design. Section 4 shows preliminary results from using VIT-PLA on an artificial dataset and a trauma resuscitation process log. Section 5 summarizes the paper and discusses the limitations of our current work.

2. METHODOLOGY

The core methods implemented in VIT-PLA can be summarized as follows (Figure 2): (1) clustering of process traces (workflow data) based on proximity of data objects, (2) aggregation of process traces and selection of cluster prototype, (3) regression analysis to explore underlying knowledge, (4) interactive visualization of process traces and statistical analysis results. This section will describe (1), (2), and (3); (4) will be discussed in Section 3.

2.1 Data Preprocessing: Sequencing of Traces

Process sequencing is necessary before more advanced processing. Activities coded in a process log usually have start and end timestamps (some logs may not include end time) for each activity. Idle time may exist between activities, and some activities may be executed concurrently (Figure 3(a)). In process mining, process traces are usually sequenced by ascending order of the start time of activities (Figure 3(b)).

2.2 Summary Visualization of Process Logs

2.2.1 Process Trace Clustering

Our approach uses clustering techniques to simplify the process trace visualizations. Clustering provides an abstraction from the original data objects to generalized data representatives, i.e. cluster prototypes. In most data mining problems, data clusters are calculated based on the data objects' feature set. However, to aggregate process traces that follow an underlying workflow model, we cluster the traces based on the similarity of their

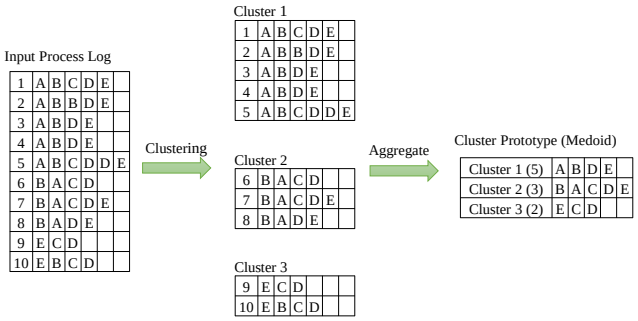


Figure 4. An example showing data clustering and aggregation. The cluster prototype used here is cluster medoid.

constituent tasks in terms of task type and sequential order of execution [10]. That is to say, our sole feature used for clustering is the structure of each trace's task sequence, not the process attributes.

In VIT-PLA, the clustering algorithm we use is agglomerative hierarchical clustering [15] with Ward's method [22] as clustering criterion. We calculate the similarity of process traces based on Edit Distance [8] (a.k.a. Levenshtein Distance [11]). If activity duration information is also available, the similarity can be calculated with "Duration-Aware Edit Distance" [16], a metric derived from Edit Distance that penalizes dissimilarity between durations of the same activity type.

2.2.2 Cluster Prototype and Trace Alignment

After clustering, each cluster can be characterized by a cluster prototype (Figure 4). Because it is not practical to visualize all the data objects on a single computer screen, a substantial reduction in the data size is needed. The deployment of cluster prototypes helps compress the dataset.

Several candidates can be considered as cluster prototype, such as the widely-used cluster centroid [14], the center of a cluster. There is, however, a great chance that there may not be an actual data point at the cluster's center. In this case, the centroid location is calculated from the data in the cluster with the aim of minimizing the sum-squared distance to other points.

Note that for categorical data and event-based data, the notion of a center (centroid) does not apply [14]. For example, the centroid of categorical data (e.g. {orange, apple, banana}) cannot be determined. In this case, we may use the cluster medoid, the most representative data object in the cluster, i.e. a data point with minimal average dissimilarity to all other objects in the cluster. The medoid, however, may not be adequate if the cluster does not contain an "appropriate" representative.

To ensure that the chosen sequence is representative of the cluster, we used the consensus sequence as the cluster prototype even though it may not be an observed trace from the data. The consensus sequence, a concept derived from aligning biological sequences (e.g. DNA) in bioinformatics, is a sequence of the most frequent residues found in the alignment matrix's columns. In process mining, consensus sequences may be considered the "average" or "common" sequence of tasks [9] (Figure 5). To find the consensus sequence for each cluster, trace alignment [9][16] needs to be performed using traces from each cluster respectively. Trace alignment reformats the original data by placing the same or similar activities of all traces to the same column of the alignment matrix. If a matching activity cannot be found, a gap symbol "-" is inserted. Bose and Van der Aalst [9] have shown how to use trace alignment techniques to visualize and analyze process traces

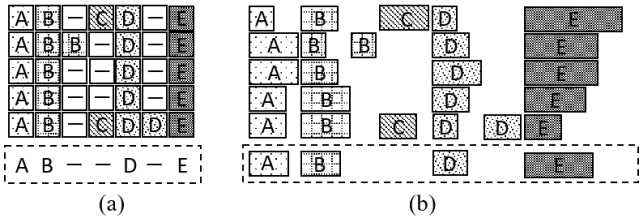


Figure 5. An example of two types of trace alignment: (a) Context-Aware and (b) Duration-Aware. The sequences at the bottom of (a) and (b) are consensus sequences derived from the data. A gap symbol "-" or white space is inserted if a match cannot be found. The five process traces shown here are from Cluster 1 in Figure 4.

(Figure 5(a)). In our previous work, we extended their work by introducing a duration-aware trace alignment algorithm [16] that also takes activity duration into consideration. In our implementation, the alignment algorithm can work for data either with or without activity durations (Figure 5).

2.3 Association between Trace Clusters and Trace Attributes

In addition to visualization, VIT-PLA also provides statistical analysis functions. The goal of our statistical analyses is to help the user discover the underlying associations between data cluster membership and trace attributes. This goal is accomplished using either multinomial or binary logistic regression. The user chooses between these two statistical methods depending on the domain question being asked. Multinomial logistic regression works for binary comparison between two clusters (one-vs.-one cluster comparison), while binomial logistic regression works for binary comparison between one cluster and the rest of the clusters (one-vs.-rest). Using both logistic regression models can help discover attributes associated with particular clusters.

2.3.1 Multinomial logistic regression

In multinomial logistic regression [12], let K denote the number of independent variables, and let J denote the number of discrete categories of the dependent variable, where $J \geq 2$. In our case, the independent variables correspond to the trace attributes and the dependent variables correspond to the trace cluster membership. The number of trace attributes is K and the number of clusters is J. By default, we define the last category (the Jth cluster) to be the reference category, against which logits of the first $J-1$ categories are compared. Let C denote cluster membership. Represented formally:

$$\ln\left(\frac{P(C=i)}{P(C=J)}\right) = \ln\left(\frac{P(C=i)}{1-\sum_{j=1}^{J-1}P(C=j)}\right) = \beta_{i0} + \beta_{i1}x_{i1} + \beta_{i2}x_{i2} + \cdots + \beta_{iK}x_{iK}, \quad i = 1, \ldots, K-1 \quad (1)$$

where x_i are trace attributes, and β_i are regression coefficients for each of the trace attributes. In VIT-PLA, users can also choose which cluster to use as the reference category.

2.3.2 Binomial logistic regression

Binary logistic regression [12] is a special case of multinomial logistic regression, in which there are only two categories ($J = 2$). In our problem, one category is the target cluster of interest and the other category is all other clusters. Let K denote the total number of independent variables and C denote cluster membership. Represented formally:

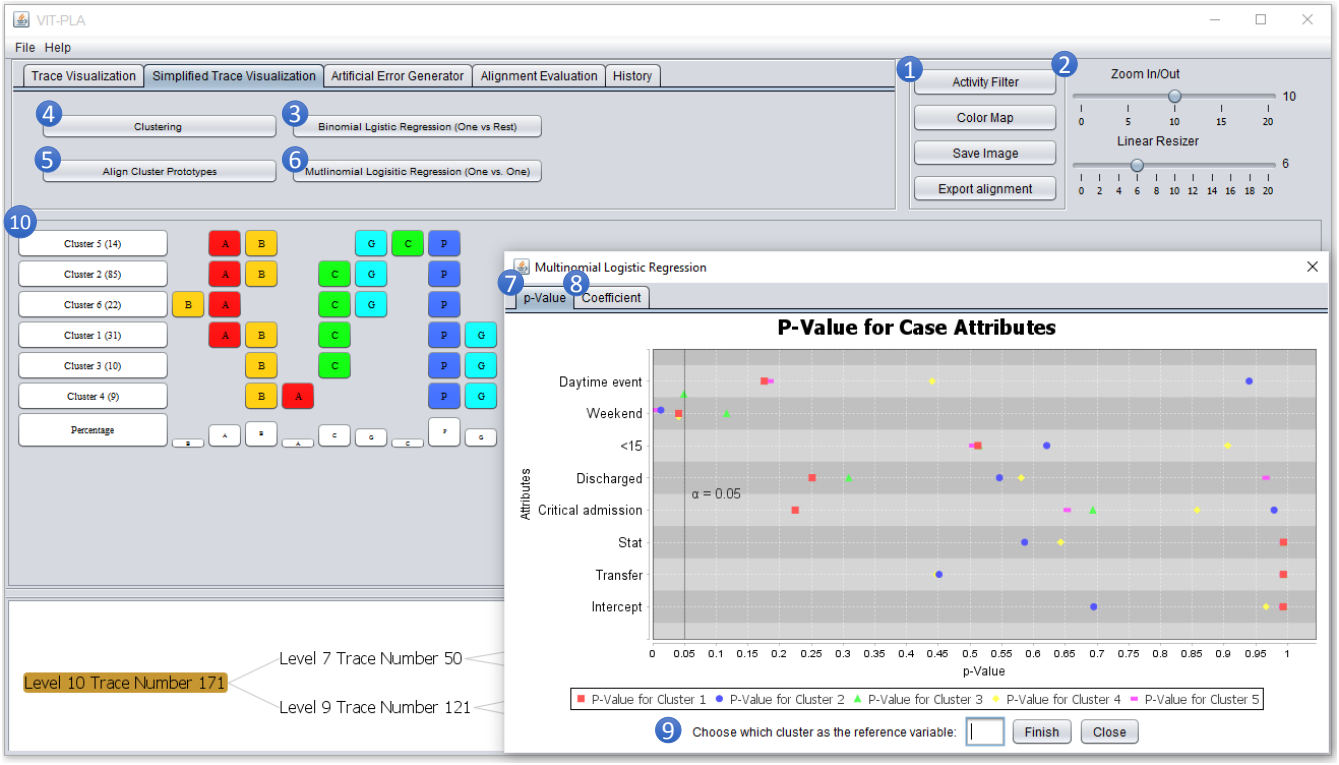


Figure 6. VIT-PLA Graphical User Interface showing aggregated data, hierarchical clustering results, and statistics from the multinomial logistic regression analysis. The data shown here is the same as the data in our 2nd case study. Please note that there are other functions of VIT-PLA that are not displayed in this figure.

$$\ln\left(\frac{P(C=i)}{P(C\neq i)}\right) = \ln\left(\frac{P(C=i)}{1-P(C=i)}\right) = \beta_{i0} + \beta_{i1}x_{i1} + \beta_{i2}x_{i2} + \cdots + \beta_{iK}x_{iK}, \quad i = 1, \ldots, K \quad (2)$$

where the parameters have the same meaning as in Eq.1.

2.3.3 Hypothesis Test

To identify which trace attributes are significantly associated with cluster membership, we use the Wald test [13] for logistic regression, which is defined as:

$$W = \frac{(\hat{\beta}_i - \beta_i)}{\widehat{se}(\hat{\beta}_i)}$$

where $\hat{\beta}_i$ is the regression coefficient for trace attributes x_i; $\beta_i = 0$ is the null hypothesis, i.e. the trace attribute x_i has a corresponding coefficient of zero; se is standard error. In our implementation, we use a normal distribution and z-values for calculating p-values. The null hypothesis can be rejected when p-value is less than or equal to alpha, the significance level which is most often set at 0.05.

3. VISUAL INTERFACE DESIGN

During software development, we received feedback from domain experts and continuously improved our design. In this section, we describe the first prototype of VIT-PLA. The visual interface design (Figure 6) was developed with three main goals:

G1. Interactive visualization of raw process traces, the basic visualization functionality.

G2. Simplified visualization of process traces (for large data applications).

G3. Visualization of trace cluster vs. trace attribute association statistics.

Although VIT-PLA has many other functions, the rest of this paper focuses on how its design achieves these three goals.

3.1 G1: Three Common Ways to Visualize Raw Process Traces

VIT-PLA provides three common ways of visualizing raw process traces. We refer to the data as "raw process traces" to distinguish goal G1 from G2, where the data is visualized in an aggregated format. The three visualization methods are:

1) *Simple stack of activities in the process traces* (Figure 7(a) without activity duration, and Figure 7(b) with activity duration). This approach is one of the simplest ways to visualize process traces. Activities are stacked based on their occurrence time. Activity information can be accessed with a mouse click on the corresponding symbol. This visualization is easily interpretable and computationally efficient, but it cannot provide deep insights into the data.
2) *Overlay of the process execution on the timeline* (Figure 8). Activities are scaled based on duration and aligned to the timeline according to their start and end times. The advantage of this visualization approach is that it clearly shows the concurrent activities in each process.
3) *Process trace alignment* (Figure 9(a) context-aware alignment and Figure 9(b) duration-aware alignment). The context-aware trace alignment algorithm is based on Bose and Van der Aalst's work [9] and the duration-aware trace alignment algorithm proposed in our previous research [16]. The duration of each activity in the consensus sequence (bottom

Figure 11. Simple stack (a) Process executions are stacked (b) Process executions are stacked and symbol blocks are scaled based on activity duration. Each row represents a single trace and each block represents a single activity. The data comes from Cluster 1 in Figure 4.

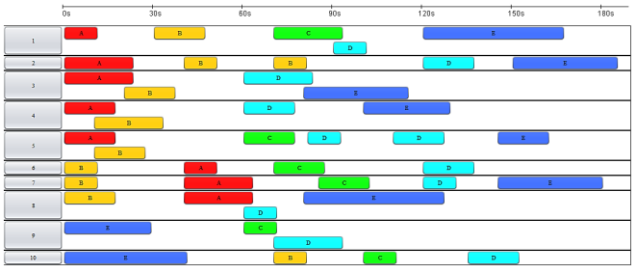


Figure 12. Visualize process traces on a timeline. The top scale is the timeline with second as the unit. Each row, separated by a bold line, represents a single process. Each block represents a single activity. Symbol blocks that are vertically stacked in one process are activities occurring simultaneously. The data comes from the input log in Figure 4.

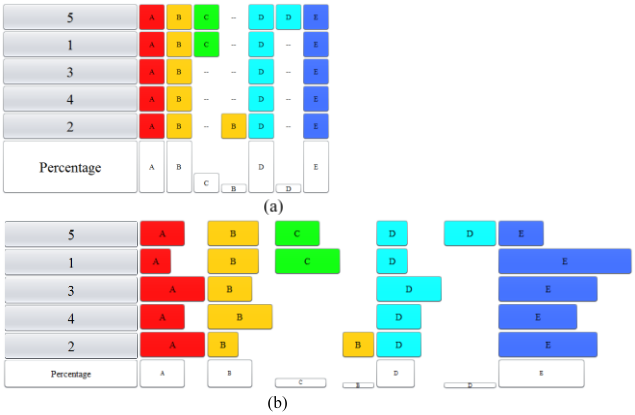


Figure 13. Alignment (a) Process trace alignment (b) Duration-aware trace alignment. Each row represents a single process and each block represents an activity. The bottom line of each figure is the consensus sequence. Dashes or spaces are introduced to achieve alignment of the activities. The data comes from Cluster 1 in Figure 4.

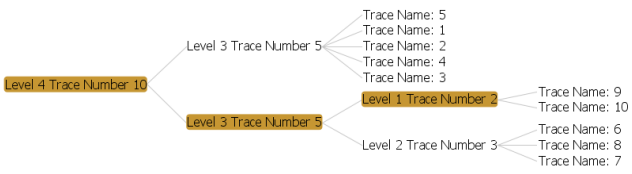


Figure 14. Hierarchical Tree Structure (we cited the same source code from ProM [9] here and made modifications showing only the number of clusters specified by the user). The result is based on the data in Figure 4.

line of Figure 9(b)) of duration-aware trace alignment is the mean activity duration of the corresponding column. Compared with the previous two visualizations, the alignment view makes it easier to interpret process traces and extract insights. When considering algorithm execution time, our previous research found that for a moderately-sized dataset (e.g. 50,000 activities, ~1,000 traces and ~50 activity for each trace), the alignment can be effectively calculated in 25.5±1.5 seconds [16]. This time is not instantaneous (which would be ideal), but is still reasonable.

3.2 G2: Simplified Visualization of Process Traces

The first interactive visualization feature in G2 is the selection of cluster number (clicking button ① in Figure 6 and inputting cluster number *k* in the pop-up dialogue). A hierarchical tree structure with *k* clusters will be shown at the bottom panel (Figure 6 and Figure 10) where the non-leaf (a.k.a. internal) nodes show the current height (a.k.a. depth) and process traces included under this node. *k* leaf nodes correspond to the *k* clusters and display all the process IDs in the cluster.

After clustering, each cluster is represented with its own cluster prototype. By default, the cluster prototypes are visualized as activity stacks (Figure 11). The prototypes can also be visualized in alignment view (Figure 6 and Figure 12) by clicking on the button "Align Cluster Prototype" (② in Figure 6). Another interactive function allows the user to check the pre-aggregated traces under a certain cluster. This feature may be accessed by clicking on the buttons showing the cluster information (③ in Figure 6).

3.3 G3: Visualization of Statistics of Trace Clusters vs. Trace Attributes.

Users can access statistics of trace clusters and trace attributes by clicking on the button "Multi-Logistic Regression" (⑤ in Figure 6)

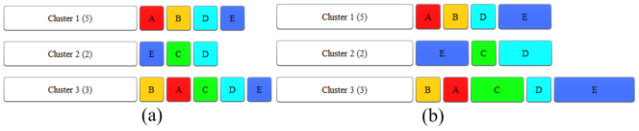


Figure 15. Simplified visualization of raw process traces. Each row is a cluster's prototype. The information in the white block before the prototypes shows the cluster ID that each prototype represents and the number of process traces in that cluster. (a) Cluster prototypes are consensus sequences calculated from context-aware alignment (Figure 9(a)); (b) Cluster prototypes are consensus sequences calculated from duration-aware alignment (Figure 9(b)). The data comes from Figure 4.

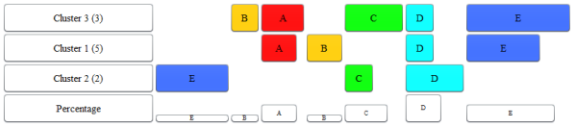
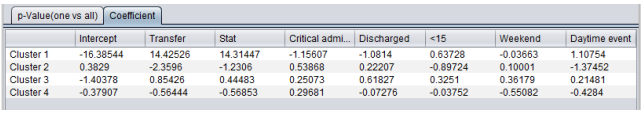


Figure 16. Alignment view of the cluster prototypes in Figure 15(a). The data comes from Figure 4.



	Intercept	Transfer	Stat	Critical admi...	Discharged	<15	Weekend	Daytime event
Cluster 1	-16.38544	14.42526	14.31447	-1.15607	-1.0814	0.63728	-0.03663	1.10754
Cluster 2	0.3829	-2.3596	-1.2306	0.53868	0.22207	-0.89724	0.10001	-1.37452
Cluster 3	-1.40378	0.85426	0.44483	0.25073	0.61827	0.3251	0.36179	0.21481
Cluster 4	-0.37907	-0.56444	-0.56853	0.29681	-0.07276	-0.03752	-0.55082	-0.4284

Figure 17. Statistics for regression coefficients

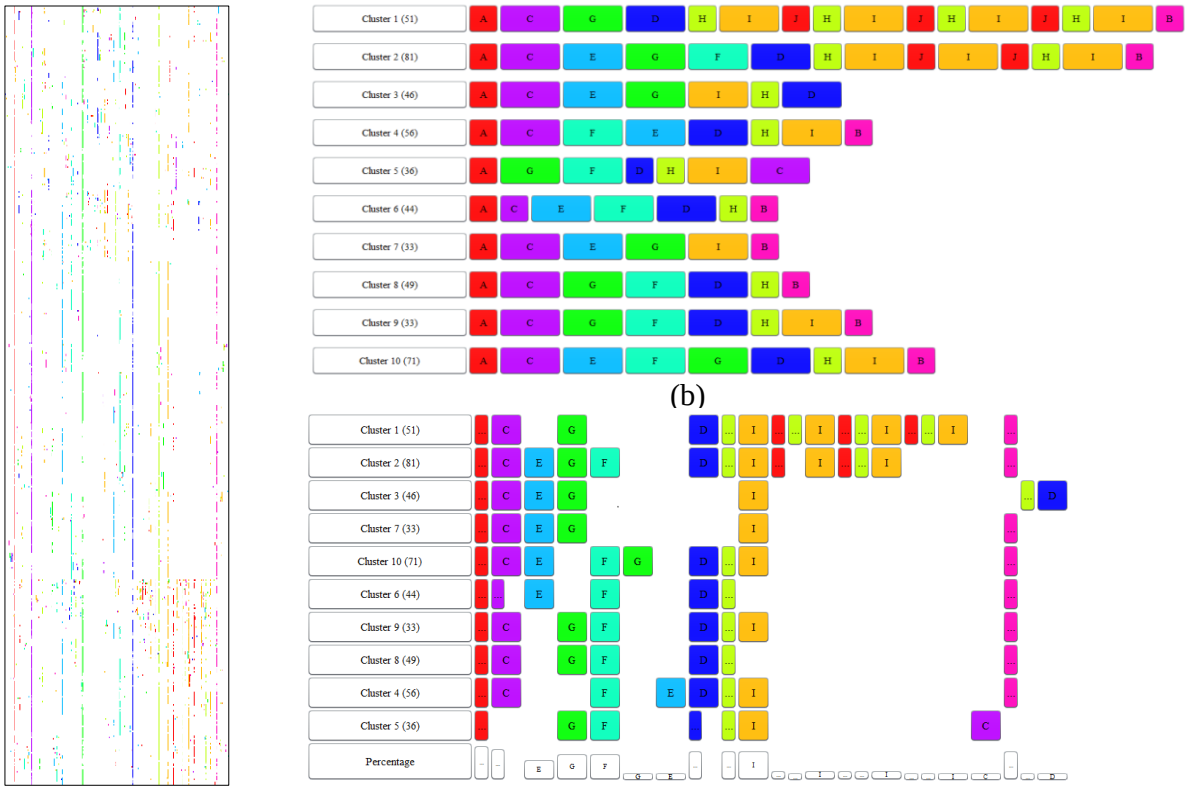


Figure 18. Visualization of artificially generated dataset. (a) Alignment view of all 500 process traces; (b) Simplified visualization of 500 process traces using 10 cluster prototypes; (c) Alignment view of 10 cluster prototypes.

or on "Binomial Logistic Regression" (④ in Figure 6). The number of clusters is decided by the user. The significance tests for trace attributes on trace clusters (*p*-value statistics) are shown in a chart (⑥ in Figure 6, JFreeChart library [18] is used). The horizontal axis represents the *p*-value, while the vertical axis represents the trace attributes. The *p*-value of different clusters is denoted with different shapes and colors. Because alpha = 0.05 is widely used as the significance level, we placed a highlighted line at this level. When performing multinomial logistic regression, the reference category is set to the last-numbered category by default. Users, however, may change the reference category manually (⑦ in Figure 6). In addition to *p*-values for each trace attribute, the regression coefficients of the logistic regression model are also listed in a table (⑧ in Figure 6 and Figure 13).

3.4 Additional supportive functions

In addition to the three main goals, VIT-PLA also includes several useful supportive functions. The Activity Filter (⑨ in Figure 6) allows the user to include and exclude activities in the visualization and analysis. The Color Map (⑨ in Figure 6) allows the user to recolor the activity symbols. The Zoom Slider (⑩ in Figure 6) enables the user to resize the activity symbols in the visualization panel (the sliders in the top-right corner control the size of the activity symbols).

4. PRELIMINARY CASE STUDY

4.1 Case Study I: Artificial Data

4.1.1 Data Description

This dataset was artificially generated using the Process Log Generator (PLG) [17]. It includes 500 process traces consisting of 10 different activity types. The drawback of this artificial data is that it does not have background attributes associated with each process trace. For this reason, we only focus on the simplification of trace visualization when using this dataset.

4.1.2 Results and Discussion

The visualization of 500 process traces without data aggregation strategies can lead to extremely large and complex visualization results (Figure 14(a)). When represented this way, the symbols are too small to identify, making it difficult to extract useful information. To improve visualization, we used clustering to aggregate the original dataset into a small number of representative process traces (Figure 14(b). In this example, we arbitrarily chose 10 clusters, a manageable number of clusters to understand). The visualization becomes clearer when put into the alignment view (Figure 14(c)). From these two simplified visualizations (Figure 14(b) and Figure 14(c)), it is easy to extract some interesting insights: (1) the sequential order of consensus tasks (tasks that occur more than or equal to 50% in the column) is "ACEGFDHIB"; (2) the pattern "HIJ" is repeated in two of the ten clusters (cluster 1 and cluster 2); (3) activity C is performed late in one cluster (cluster 5); and (4) activity D is performed late in one cluster (cluster 3) and omitted in another (cluster 7).

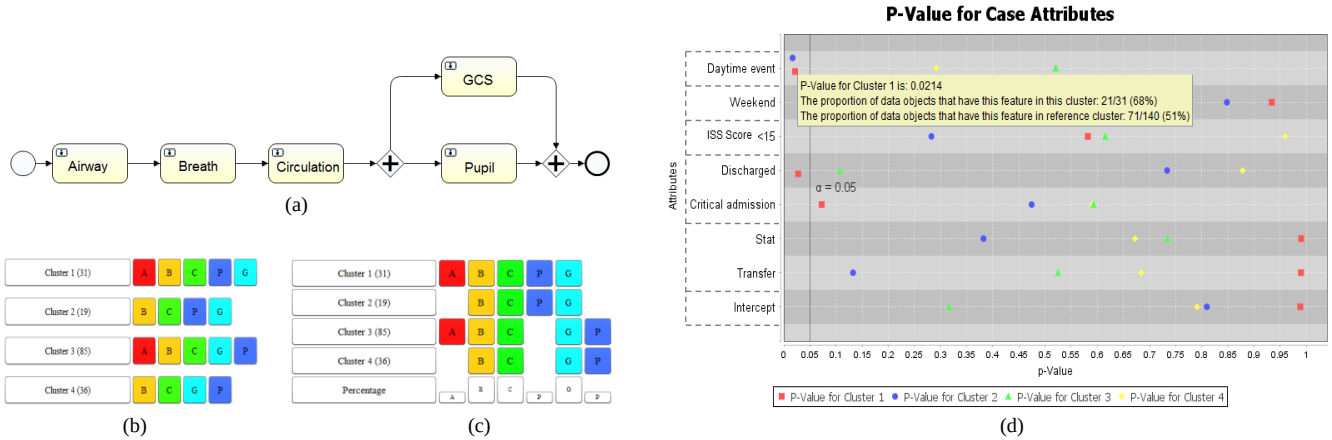


Figure 19. (a) Workflow model (drawn based on BPMN) given by domain expert describing the initial evaluation of trauma, (b) Simplified visualization of 171 traces using four cluster prototypes, (c) Alignment view of four cluster prototypes (d) *p*-value for binomial logistic regression coefficients

4.2 Case Study II: Trauma Resuscitation Workflow Data

4.2.1 Data Description

We used a trace log obtained from video analysis of 171 child trauma resuscitations between May and August 2013 at Children's National Medical Center in Washington, DC. An event log of five activities typically performed during the initial evaluation was created and used as the dataset for this case study. We obtained the workflow model for these activities from domain experts (Figure 15(a)). Activities "Airway, Breath, Circulation" follow a sequential order. Activities "GCS" and "Pupil check" are parallel and should be performed after the previous three activities. We also obtained from the medical chart review several patient and resuscitation attributes (including pre-hospital triage level, the resuscitation's time of day and day of week, Injury Severity Score [ISS], and patient admission status after the resuscitation) (Table 1). This dataset is not a "large process log," but we chose it for our preliminary analysis to demonstrate how our approach can be integrated with medical domain knowledge.

4.2.2 Results and Discussion

4.2.2.1 Data Interpretation from Visual Analysis

Four cluster prototypes were generated (Figure 15(b) and (c)). Prototypes of clusters 1 and 3 conform to our expert model, but clusters 2 and 4 do not. From the alignment view of prototypes, we can observe that the sequential order of activity GCS (G) and pupil assessment (P) is interchangeable, which conforms with the parallel structure in our expert model. Visualizations of pre-aggregated traces for each prototype are not displayed, but users can visualize the traces by clicking on the cluster button at the front of each row (Figure 15(b) and (c)).

Table 1 Process trace attributes

Attribute List	Values		
Weekend Event	1	0	
Daytime Event	1	0	
ISS Score	<15	≥15	
Activation Level[a]	Attending Stat	Stat	Transfer
EDDISPGroup[b]	Non-critical Admission	Critical Admission	Discharged

[a.] Activation level = pre-hospital triage level

[b.] EDDISPGroup = admission status of patients after ED care

With the attribute data for these process traces, we can perform statistical analysis to explore the underlying correlation between the trace attributes and trace cluster membership. The following are examples of the statistical findings, followed by feedback from domain experts:

Observation #1: Attribute "Daytime Event" is statistically significant (*p*-value = 0.021, red square point in row "Daytime event" in Figure 15) for cluster 1. The regression coefficient of Daytime Event is 1.108 (Figure 13). This attribute is statistically significant because the proportion of data objects that have this feature (daytime = 1) in this cluster is 12/31 (68%), while the proportion of data objects that have this feature (daytime = 1) in the reference category (all other cluster) is 71/140 (51%).

Observation #2: Attribute "Daytime Event" is statistically significant (*p*-value = 0.017, blue circle point in row "Daytime event" in Figure 15) for cluster 2. The regression coefficient of Daytime Event is −1.375 (Figure 13). This attribute is significantly significant because the proportion of data objects that have this feature (daytime = 1) in this cluster is 6/19 (31%), while the proportion of data objects that have this feature (daytime = 1) in the reference category (all other cluster) is 86/152 (57%).

Medical expert feedback: For the care of injured patients, improved outcomes are associated with compliance with the Advanced Trauma Life Support model [19], represented here as the expert model. We find that one cluster (cluster 1) whose cluster prototype follows the model occurs more often during the day and another cluster (cluster 2) whose cluster prototype deviates from the model occurs more often at night. This association finding supports previous work showing decreased compliance with trauma protocols at night [20].

4.2.2.2 Domain Expert Feedback on VIT-PLA Design:

To evaluate the quality of our design, we had two medical domain experts evaluate a prototype of VIT-PLA. Both positive and negative feedback was received.

Both domain experts liked the visualization's flexibility and interactivity. They found that its data clustering, activity filtering, symbol resizing, and recoloring functions were very useful. They

were also found that with the knowledge uncovered by the program's statistical analysis was useful. One domain expert found it useful to switch between the aggregated data and the original traces, and also commented on the helpfulness of the cluster's "average sequence".

Most negative comments focused on our approach for statistical analysis. One domain expert felt that data-driven clustering approach lacked consistency because its result varied when different clustering algorithms or similarity metrics were used. Also, the domain expert found that some small clusters did not have sufficient data to support the statistical hypothesis test correlating trace clusters and trace attributes.

5. SUMMARY AND FUTURE WORK

As process mining finds increased usage in many domains, visual analytic tools for process sequences are in high demand. We introduced VIT-PLA, a visual and interactive workflow data analysis tool that is able to visualize large process logs. With these visualizations and integrated statistical testing, VIT-PLA is able to obtain results not revealed by simple observation.

The limitation of our current work is that we only implemented the hierarchical clustering approach with two process trace proximity metrics. In our future work, we will evaluate other clustering algorithms (e.g. KNN, feature-based k-means, HMM-based clustering). Also, the determination of cluster number, a typically non-trivial task, is still manual. In the future, we plan on building a function that suggests cluster number based on some cluster metric.

6. ACKNOWLEDGMENTS:

This research is supported by National Institutes of Health under grant number 1R01LM011834-01A1.

7. REFERENCES

[1] Van Der Aalst, Wil. Process mining: discovery, conformance and enhancement of business processes. Springer Science & Business Media, 2011

[2] Van der Aalst, Wil, Ton Weijters, and Laura Maruster. "Workflow mining: Discovering process models from event logs." Knowledge and Data Engineering, IEEE Transactions on 16.9 (2004): 1128-1142.

[3] Monroe, Megan, et al. "Temporal event sequence simplification." Visualization and Computer Graphics, IEEE Transactions on 19.12 (2013): 2227-2236.

[4] Malik, Sana, et al. "Cohort comparison of event sequences with balanced integration of visual analytics and statistics." Proceedings of the 20th International Conference on Intelligent User Interfaces. ACM, 2015.

[5] Wongsuphasawat, Krist, and David Gotz. "Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization."Visualization and Computer Graphics, IEEE Transactions on 18.12 (2012): 2659-2668.

[6] Perer, Adam, and Fei Wang. "Frequence: interactive mining and visualization of temporal frequent event sequences." Proceedings of the 19th international conference on Intelligent User Interfaces. ACM, 2014.

[7] Perer, Adam, Fei Wang, and Jianying Hu. "Mining and exploring care pathways from electronic medical records with visual analytics." Journal of biomedical informatics 56 (2015): 369-378.

[8] Song, Minseok, and Wil MP van der Aalst. "Supporting process mining by showing events at a glance." 7th Annual Workshop on Information Technologies and Systems. 2007.

[9] Bose, RP Jagadeesh Chandra, and Wil MP van der Aalst. "Process diagnostics using trace alignment: opportunities, issues, and challenges."Information Systems 37.2 (2012): 117-141

[10] Bose, RP Jagadeesh Chandra, and Wil MP van der Aalst. "Context Aware Trace Clustering: Towards Improving Process Mining Results." SDM. 2009.

[11] Levenshtein, Vladimir I. "Binary codes capable of correcting deletions, insertions, and reversals." Soviet physics doklady. Vol. 10. No. 8. 1966.

[12] Czepiel, Scott A. "Maximum likelihood estimation of logistic regression models: theory and implementation." Available at czep. net/stat/mlelr. pdf(2002).

[13] Wasserman, Larry. All of statistics: a concise course in statistical inference. Springer Science & Business Media, 2013.

[14] Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. Introduction to data mining. Vol. 1. Boston: Pearson Addison Wesley, 2006.

[15] Jain, Anil K., and Richard C. Dubes. Algorithms for clustering data. Prentice-Hall, Inc., 1988.

[16] Sen Yang, Moliang Zhou, Rachel Webman, JaeWon Yang, Aleksandra Sarcevic, Ivan Marsic and Randall S. Burd, "Duration-Aware Alignment of Process Traces", Accepted for Industrial Conference on Data Mining (ICDM 2016), New York, NY, July 13-17, 2016

[17] Burattin, Andrea, and Alessandro Sperduti. PLG: a Process Log Generator. Tech. rep, 2010.

[18] http://www.jfree.org/

[19] van Olden, Ger DJ, et al. "Clinical impact of advanced trauma life support.", The American journal of emergency medicine 22.7 (2004): 522-525.

[20] Carter, Elizabeth A., et al. "Adherence to ATLS primary and secondary surveys during pediatric trauma resuscitation." Resuscitation 84.1 (2013): 66-71.

[21] Du, Fan, et al. "Coping with Volume and Variety in Temporal Event Sequences: Strategies for Sharpening Analytic Focus."

[22] Ward Jr, Joe H. "Hierarchical grouping to optimize an objective function."*Journal of the American statistical association* 58.301 (1963): 236-244.