Editors & Organizers: Polo Chau, Jilles Vreeken, Matthijs van Leeuwen, Christos Faloutsos

Proceedings of the
ACM SIGKDD 2014 Full-day Workshop on

Interactive Data Exploration and Analytics

IDEA

2014

New York City, USA
Aug 24, 2014

poloclub.gatech.edu/idea2014

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

IDEA'14, August 24, 2014, New York, NY, 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 (KU Leuven)
Christos Faloutsos (Carnegie Mellon University)

Program Committee

Adam Perer (IBM, USA)
Andreas Holzinger (Medical University Graz, Austria)
Antti Oulasvirta (Aalto University, Finland)
Antti Ukkonen (Aalto University, Finland)
Arno Knobbe (Universiteit Leiden, the Netherlands)
Arno Siebes (Universiteit Utrecht, the Netherlands)
Cody Dunne (UMD, USA)
Dafna Shahaf (Stanford, USA)
Esther Galbrun (Boston University, USA)
Fei Sha (University of Southern California, USA)
Geoff Webb (Monash University, Australia)
George Forman (HP Labs)
Hanghang Tong (City University of New York and Arizona State University, USA)
Jaakko Hollmen (Aalto University, Finland)
Jaegul Choo (Georgia Tech)
Jefrey Lijffijt (Aalto University, Finland)
Kai Puolamäki (Finnish Institute of Occupational Health, Finland)
Klaus Mueller (Stony Brook University, USA)
Leman Akoglu (Stony Brook University)
Lisa Singh (George Town, USA)
Michael Berthold (University of Konstanz, Germany)
Nan Cao (IBM, USA)
Nikolaj Tatti (Aalto University, Finland)
Olivier Thonnard (Symantec)
Parikshit Ram (Georgia Tech, USA)
Pauli Mietinnen (Max Planck Institute for Informatics, Germany)
Saleema Amershi (Microsoft Research)
Stefan Kramer (University Mainz, Germany)
Thomas Gärtner (University of Bonn, Germany)
Thomas Seidl (Aachen University, Germany)
Tijl De Bie (University of Bristol, UK)
Tina Eliassi-Rad (Rutgers)
U Kang (KAIST)
Zhicheng 'Leo' Liu (Stanford)

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 exploratively 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, the main program of IDEA'14 consists of sixteen papers covering various aspects of interactive data exploration and analytics. Nine papers were accepted for oral presentation, with eight more selected for poster presentation with accompanying interactive demos. These papers were selected from a total of 23 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 Ben Shneiderman and Aditya Parameswaran for engaging keynote presentations on the fundamental aspects of interactive data exploration and visualization.

Polo Chau & Jilles Vreeken & Matthijs van Leeuwen & Christos Faloutsos
Saarbrücken, July 2014

Table of Contents

Invited Talks

Research Papers

Invited Talk

Information Visualization for Knowledge Discovery: Big Insights from Big Data

Ben Shneiderman
Human Computer Interaction Lab
University of Maryland, College Park
ben@cs.umd.edu

Abstract

Interactive information visualization tools provide researchers with remarkable capabilities to support discovery from Big Data resources. Users can begin with an overview, zoom in on areas of interest, filter out unwanted items, and then click for details-on-demand. The Big Data initiatives and commercial success stories such as Spotfire and Tableau, plus widespread use by prominent sites such as the New York Times have made visualization a key technology.

The central theme is the integration of statistics with visualization to support user discovery. Our work focuses on temporal event sequences such as found in electronic health records (www.cs.umd.edu/hcil/eventflow), and social network data such a twitter discussion patterns (www.codeplex.com/nodexl). The talk closes with 8 Golden Rules for Big Data.

Bio

Ben Shneiderman is a Distinguished University Professor in the Department of Computer Science and Founding Director (1983-2000) of the Human-Computer Interaction Laboratory at the University of Maryland. He is a Fellow of the AAAS, ACM, and IEEE, and a Member of the National Academy of Engineering, in recognition of his pioneering contributions to human-computer interaction and information visualization. His contributions include the direct manipulation concept, clickable web-link, touchscreen keyboards, and dynamic query sliders for Spotfire, development of treemaps, innovative network visualization strategies for NodeXL, and temporal event sequence analysis for electronic health records.

Ben is the co-author with Catherine Plaisant of *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (5th ed., 2010). With Stu Card and Jock Mackinlay, he co-authored *Readings in Information Visualization: Using Vision to Think* (1999). His book *Leonardo's Laptop* appeared in October 2002 (MIT Press) and won the IEEE book award for Distinguished Literary Contribution. His latest book, with Derek Hansen and Marc Smith, is *Analyzing Social Media Networks with NodeXL* (2010).

Invited Talk

Human-Powered and Visual Data Management

Aditya Parameswaran
Department of Computer Science
University of Illinois (UIUC)
adityagp@illinois.edu

Abstract

This talk will consist of two parts. The first part will be on an ongoing project: Fully automated algorithms are inadequate for many data analysis tasks, especially those involving images, video, or text. Thus, we need to combine crowdsourcing with traditional computation, to improve the process of understanding, extracting and managing data. In this part, I will present a broad perspective of our research on this topic. I will then present details of one of the problems we have addressed: filtering large data sets with the aid of humans. For more details, see: i.stanford.edu/~adityagp/scoop.html

The second part will be on a project that is just starting off: Data scientists rely on visualizations to interpret the data returned by queries, but finding the right visualization remains a manual task that is often laborious. We propose a system that partially automates the task of finding the right visualizations for a query. The output will comprise a recommendation of potentially "interesting" or "useful" visualizations, where each visualization is coupled with a suitable query execution plan. I will discuss the technical challenges in building this system and preliminary results, and outline an agenda for future research. For more details, see http://goo.gl/FHZY61 (to appear at VLDB '14)

Bio

Aditya Parameswaran is an Assistant Professor in Computer Science at the University of Illinois (UIUC). He is currently spending the year visiting MIT CSAIL, after completing his Ph.D. from Stanford University in Sept. 2013, advised by Prof. Hector Garcia-Molina. He is broadly interested in data analytics, with research results in human computation, visual analytics, information extraction and integration, and recommender systems. Aditya is a recipient of the Arthur Samuel award for the best dissertation in Computer Science at Stanford (2013), the SIGMOD Jim Gray dissertation award (2014), the Key Scientific Challenges Award from Yahoo! Research (2010), two best-of-conference citations (VLDB 2010 and KDD 2012), the Terry Groswith graduate fellowship at Stanford (2007), and the Gold Medal in Computer Science at IIT Bombay (2007).

VizLinc: Integrating information extraction, search, graph analysis, and geo-location for the visual exploration of large data sets [*]

Joel C. Acevedo-Aviles, William M. Campbell, Daniel C. Halbert, Kara Greenfield
MIT Lincoln Laboratory, Human Language Technology Group, Lexington, MA, USA
{joel, wcampbell, daniel.halbert, kara.greenfield}@ll.mit.edu

ABSTRACT

In this demo paper we introduce *VizLinc*; an open-source software suite that integrates automatic information extraction, search, graph analysis, and geo-location for interactive visualization and exploration of large data sets. VizLinc helps users in: 1) understanding the type of information the data set under study might contain, 2) finding patterns and connections between entities, and 3) narrowing down the corpus to a small fraction of relevant documents that users can quickly read. We apply the tools offered by VizLinc to a subset of the New York Times Annotated Corpus and present use cases that demonstrate VizLinc's search and visualization features.

Keywords

VizLinc, visualization, visual analytics, graph analysis, data exploration, information extraction, search, geo-location

1. INTRODUCTION

Information extraction refers to the task of automatically extracting structured information from unstructured documents. Sub-tasks like named entity, relationship, and terminology extraction are extremely useful to characterize the content of large text corpora and give data analysts a sense of what information might be present in such corpora. For many applications, characterizing individual documents is not enough. Linking relevant information across documents is the key to harnessing the informative power of a large heterogeneous corpus.

In this paper we introduce *VizLinc*; an open-source software suite that integrates automatic information extraction, search, graph analysis, and geo-location for interactive visualization and exploration of large data sets. VizLinc helps users in: 1) understanding the type of information the data set under study might contain, 2) finding patterns and connections between entities, and 3) narrowing down the corpus to a small fraction of relevant documents that users can quickly read. VizLinc is self-contained, does not require connections to online components, and scales to tens of thousands of documents. All software is publicly available through GitHub.[1]

According to the survey presented in [11] a large number of data analysis and visualization tools are available for analyzing structured data, but tools for modeling and visualizing semi- or unstructured data are still underrepresented. Commercial visual analytics (VA) systems such as *Tableau*[2] and *Spotfire Desktop*[3] are designed to connect to a variety of structured data sources. VizLinc, on the other hand, is designed with the purpose of characterizing and exploring large collections of *unstructured* documents. Like *Visual Analytics*[4] and *Centrifuge*[5], VizLinc uses graph modeling techniques to represent relationships between data items; a feature not present in most of the systems studied in [11]. *Palantir* [6] features a super set of the main visualizations that VizLinc uses (geographical map and network) and it is also suitable for unstructured data.[10] VizLinc, however, is an extensible open-source, and free software suite whereas Palantir is a commercial product. Contrary to most open-source and commercial VA tools, VizLinc does not offer the uni-variate or bi-variate statistical analysis tools often found in software of its kind.[4][11] This is something that will be addressed in future releases.

The rest of this paper is organized as follows. Sections 2 and 3 present an overview of the main components of the VizLinc software suite and the technology used to implement it. Sections 4 to 6, offer a look into VizLinc's features and usage. Lastly, we apply our techniques to a data set and present use cases to demonstrate VizLinc's capabilities in section 7.

2. SYSTEM OVERVIEW

VizLinc is a software suite composed of two main applications: the *Ingestion Tool* and the *User Interface (UI)*. The Ingestion Tool takes a set of documents as input, extracts information from unstructured text, and stores the extracted information in the format that the UI needs to allow users to search, visualize, and explore the documents' content. The process of converting the input documents into information-rich data structures, or simply the *ingestion process*, will be covered in detail in section 3. For now, it suffices to say that data ingestion is carried out entirely by the In-

[*]This work was sponsored by the Defense Advanced Research Projects Agency under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

[1] https://github.com/mitll/vizlinc
https://github.com/mitll/vizlinc_db
https://github.com/mitll/vizlinc_ingester

[2] http://www.tableausoftware.com/

[3] http://spotfire.tibco.com/discover-spotfire/spotfire-overview/spotfire-desktop

[4] http://www.visualanalytics.com/

[5] http://centrifugesystems.com/

[6] http://www.palantir.com/

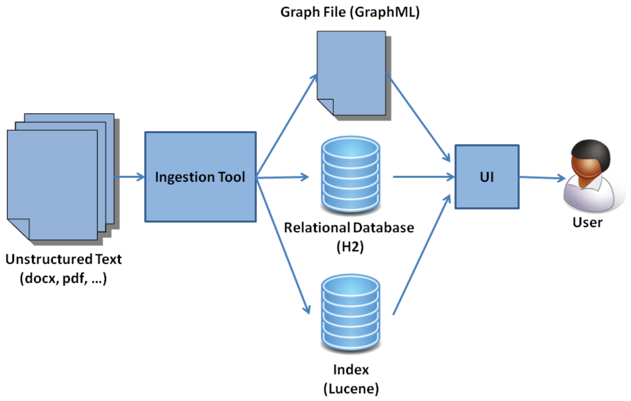


Figure 1: VizLinc main components. The Ingestion Tool processes text in a variety of formats and produces the metadata the UI requires as input for search and visualization.

gestion Tool in two major stages: *Information Extraction* and *Metadata Generation*. During Information Extraction, mentions of people, locations, and organizations are identified in text. Once identified, locations are geo-coded and people are linked based on their co-occurrence patterns in the data set. In addition, the text is indexed for search and retrieval. The metadata generation step takes the extracted information and stores it in an H2[7] relational database, a GraphML[8] graph file, and a Lucene[9] index. The UI takes the database, graph file, and index as inputs and presents a graphical user interface for interactive visualization and exploration of the original data set. For the rest of this paper, we will refer to the UI simply as *VizLinc*. Figure 1 depicts the interaction between the aforementioned components.

The Ingestion Tool is written in the Groovy programming language. *Groovy* is an agile and dynamic language for the Java Virtual Machine[10]. The graph database we use for data ingestion has a robust implementation in Groovy thus making it an ideal choice for the Ingestion Tool. For ease of use, we have developed a graphical user interface in Java Swing that calls the appropriate Groovy classes as needed.

As the reader probably inferred by this point, the UI is written in the Java programming language. Specifically, the UI is a Gephi [11] plugin. *Gephi* is an interactive visualization and exploration platform for large graphs [1] and powers all graph-related features in the UI. Gephi, in turn, is based on the *Netbeans Platform*[12], a generic framework for rapid development of Java Swing applications. One of the key distinctions of software built upon the NetBeans Platform is modularity [2]. This distinction made the integration of VizLinc's UI with Gephi a seamless one. For the rest of this paper, we will refer to the UI component simply as VizLinc.

3. DATA INGESTION

Data ingestion refers to the sequence of processing steps that generate the necessary metadata for later visualization and exploration in VizLinc. Figure 2 shows what these steps are and the order in which they are executed.

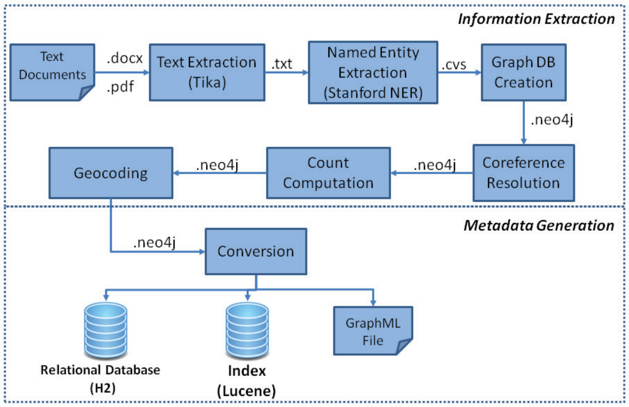


Figure 2: The data ingestion process.

VizLinc admits text in a variety of formats including Microsoft Office formats(.docx, .doc, .xls, ...), Portable Document Format (PDF) and HyperText Markup Language (HTML).[13] For this reason, the first step in the ingestion pipeline is extracting the text contained in the input documents. We use the tools provided by *Apache Tika*[14] for this purpose. Other content, such as images, is ignored.

Once text is extracted, we perform named entity recognition on each document using the *Stanford NER* [3] recognizer. During this step named entities, specifically people, locations, and organizations, are identified and extracted. Each instance of an entity in a text is called a *mention*. All mentions, information about the documents in which they appear, and their positions within those documents are stored in a *Neo4j*[15] graph database. The need to store and retrieve links between the metadata that will be generated in subsequent steps, makes this data representation an intuitive and efficient one for our purposes [7]. The graph database is augmented as ingestion progresses and, by the end of the pipeline, stores the results of all the steps performed during this process.

Mentions can have different forms yet refer to the same entity. For instance, the person entity *John Fitzgerald Kennedy* might be referred to as "John F. Kennedy", "Kennedy", and "JFK". VizLinc aims at discovering interesting patterns in text by unveiling connections between the entities mentioned. To this end, it is critical to find all the mentions of an entity both within a document and across documents in the corpus. The task of finding all expressions that refer to the same entity is denominated *coreference resolution* and is the goal of the fourth step in the ingestion pipeline. In VizLinc, approximate string matching and a simple set of rules are brought together to: 1) find all mentions of an entity within a document(e.g., "John F. Kennedy", "Kennedy") and assign them a single canonical form (e.g., "John F. Kennedy") that

[7] http://www.h2database.com/
[8] http://graphml.graphdrawing.org/
[9] http://lucene.apache.org/
[10] http://groovy.codehaus.org/
[11] http://gephi.org/
[12] https://netbeans.org/features/platform/

[13] For a list of all supported formats, see https://tika.apache.org/1.4/formats.html.
[14] http://tika.apache.org/
[15] http://www.neo4j.org/

is then used to 2) link all mentions of the same entity across documents.

Ingestion proceeds by calculating the number of times each entity is mentioned in the entire data set and the number of documents in which each appears. These values are stored in the graph database for later presentation in the graphical user interface.

One of VizLinc's main features is a map that displays the locations mentioned in a selected subset of the document set under analysis. To render this possible, locations have to be resolved to latitude/longitude coordinates that can be then highlighted in the map. This is precisely what the *Geocoding* step depicted in figure 2 does. Internally, we have used a number of approaches to carry out this step for our data sets of interest. In the version that we have publicly released, the user can point the Ingestion Tool to an online geocoding server. The *Geocoding* step marks the end of the Information Extraction stage.

During the Metadata Generation stage, particular data structures are created and saved to disk. Most of the metadata generated is stored in an *H2* database. *H2* is an open source database engine written in the Java programming language[7]. Its speed, ability to run without a server, and seamless integration with Java applications were the main reasons why we chose it over other database engines.

Co-occurrences of person entities in documents are encoded in the form of edges between nodes of a graph. For that reason, we store this information in a GraphML file. *GraphML* is a comprehensive file format for graphs which consists of a language core to describe the structural properties of a graph and a flexible extension mechanism to add application-specific data[8]. Each node in the graph represents a person entity mentioned in the data set. An edge exists between a pair of nodes if the corresponding entities co-occur in more than two documents. Co-occurrence is a symmetric relation therefore edges in the graph are undirected. In sections 5.5 and 7 we will discuss how this co-occurrence network can be used to find coherent groups and "important" people [5].

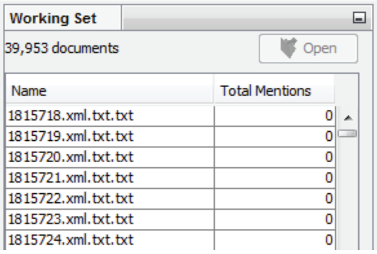
Lastly, *Lucene* is used to generate an index that stores the entire text content of the input documents in a format suitable for string searching. *Apache Lucene* is a high-performance, full-featured text search engine library written entirely in Java. Lucene is an open source project available for free download[9].

4. VIZLINC INPUT

When run for the first time, VizLinc prompts the user for the system paths of the database, index, and graph file generated by the Ingestion Tool. Additionally, VizLinc requires a tile source to populate its map. Two tile source types are supported in the current version. If users have pre-generated tiles and saved them as images, the directory in which they were saved can be specified. Otherwise, an HTTP map server can be specified through a URL. Once the input is specified and loaded, users can visualize and explore their data sets. At any point, users can point VizLinc to a different data set or tile source.

5. DATA CHARACTERIZATION

Upon loading our data in VizLinc we can immediately get a sense of the composition of our text corpus. Figure 3 shows a snapshot of the UI's main components or *views*. In the following sub-sections we describe each of these views and how they can be used for data exploration.



39,953 documents

Name	Total Mentions
1815718.xml.txt.txt	0
1815719.xml.txt.txt	0
1815720.xml.txt.txt	0
1815721.xml.txt.txt	0
1815722.xml.txt.txt	0
1815723.xml.txt.txt	0
1815724.xml.txt.txt	0

Figure 4: Working Set view

5.1 Working Document Set

The *working set* is the set of documents currently being visualized in VizLinc. At first, this set consists of all documents in the data set but as search queries are applied, this set gets narrowed down to a relevant subset of the corpus (see section 6). Keep in mind that one of the main goals of VizLinc is to empower users to quickly filter out those documents that might not contain relevant information.

The *Working Document Set* view lists the specific documents that are part of the working set. This view is shown in figure 4.

At any point in time, the number shown across the top of the view represents how many documents are being analyzed and represented in all views. The entries under the *Total Mentions* column will be explained in section 6.

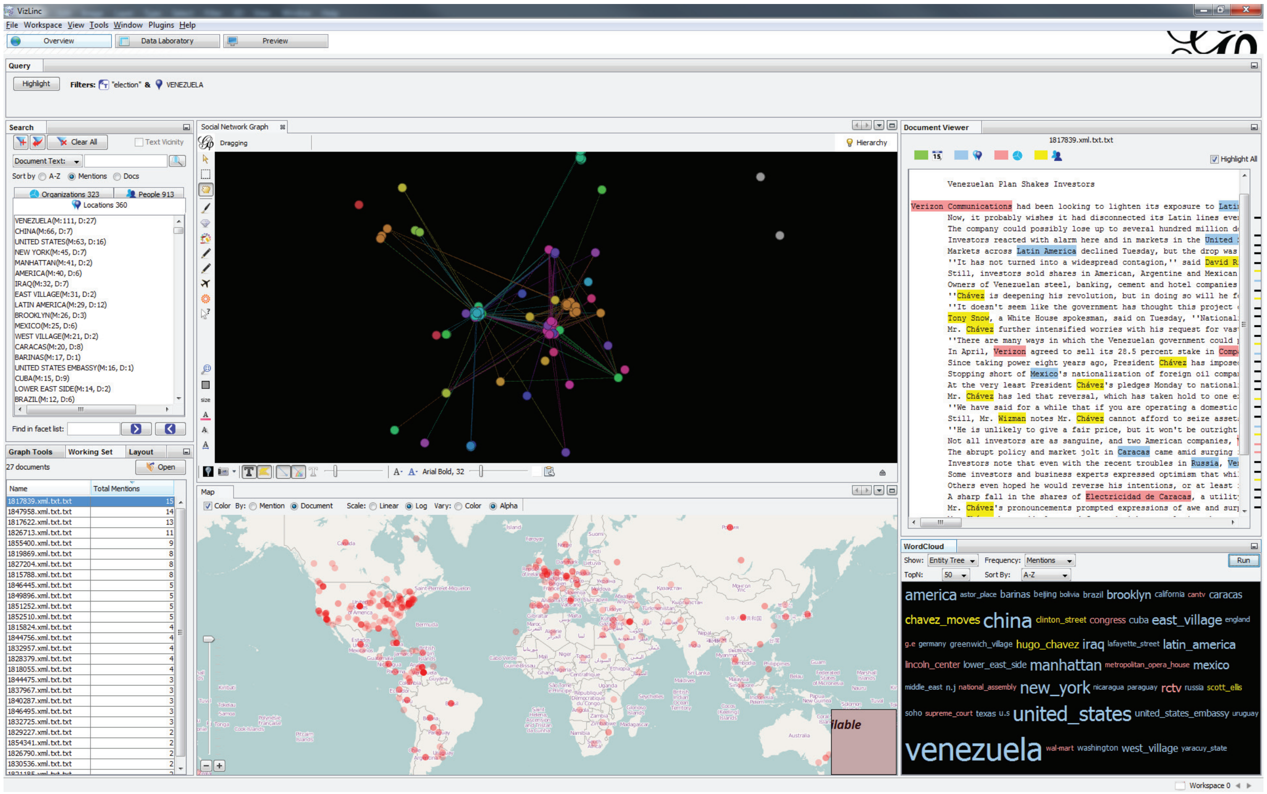
5.2 Document Viewer

Ultimately, users should be able to easily read the informative sections of a document, as determined by the search query, and draw relevant conclusions. Selecting a document name in the *Working Set* view and clicking on the *Open* button will show the document's content in the *Document Viewer*. This view displays the text extracted from the selected document in its raw form. All formatting information, other than capitalization and spacing, is discarded in the ingestion process. Figure 5 shows this view.

If the *Highlight All* check box is selected, the Document Viewer highlights all the mentions found in the document. A color code is used to distinguish between people, locations, and organizations. If this check box is not selected only those mentions that match the search query are highlighted.

5.3 Search View

Figure 6 shows the Search View. As the name implies the *Search* view allows users to search for particular terms or entities in the data set. However, that is not its sole utility. This view also lists all the people, locations, and organizations automatically extracted during the ingestion process. The number that appears next to the entity type is the number of entities of that type present in the working set. Each entity is shown along with its mention and document count. The mention count is the number of times an entity is referred to in the working set whereas the document count is the number of documents in which an entity is mentioned. Both counts are shown in this view only as it pertains to the current working set, i.e., for all the documents that match



Filters: "election" & VENEZUELA

Sort by ○ A-Z ● Mentions ○ Docs

Organizations 323 | People 913 | Locations 360

VENEZUELA(M:111, D:27)
CHINA(M:66, D:7)
UNITED STATES(M:63, D:16)
NEW YORK(M:45, D:7)
MANHATTAN(M:41, D:2)
AMERICA(M:40, D:6)
IRAQ(M:32, D:7)
EAST VILLAGE(M:31, D:2)
LATIN AMERICA(M:29, D:12)
BROOKLYN(M:26, D:3)
MEXICO(M:25, D:6)
WEST VILLAGE(M:21, D:2)
CARACAS(M:20, D:8)
BARINAS(M:17, D:1)
UNITED STATES EMBASSY(M:16, D:1)
CUBA(M:15, D:9)
LOWER EAST SIDE(M:14, D:2)
BRAZIL(M:12, D:6)

Find in facet list:

27 documents

Name	Total Mentions
1817839.xml.txt.txt	15
1847958.xml.txt.txt	14
1817622.xml.txt.txt	13
1826713.xml.txt.txt	11
1855400.xml.txt.txt	9
1819869.xml.txt.txt	8
1827204.xml.txt.txt	8
1815788.xml.txt.txt	8
1846445.xml.txt.txt	5
1849896.xml.txt.txt	5
1851252.xml.txt.txt	5
1852510.xml.txt.txt	5
1815824.xml.txt.txt	4
1844756.xml.txt.txt	4
1832957.xml.txt.txt	4
1828379.xml.txt.txt	4
1818055.xml.txt.txt	4
1844475.xml.txt.txt	3
1837967.xml.txt.txt	3
1840287.xml.txt.txt	3
1846495.xml.txt.txt	3
1832725.xml.txt.txt	3
1829227.xml.txt.txt	2
1854341.xml.txt.txt	2
1826790.xml.txt.txt	2
1830536.xml.txt.txt	2

☑ Color By: ○ Mention ● Document Scale: ○ Linear ● Log Vary: ○ Color ● Alpha

1817839.xml.txt.txt

☑ Highlight All

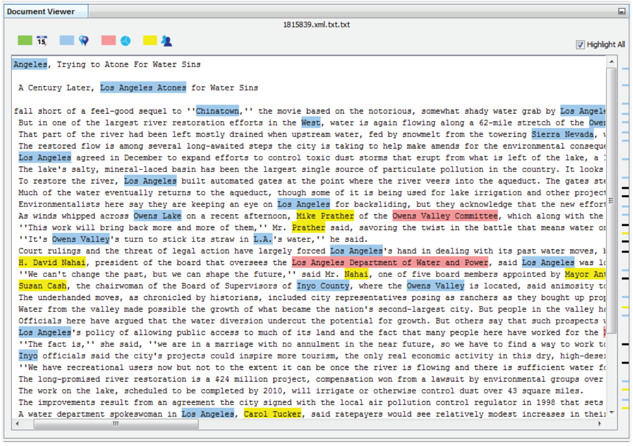
```
        Venezuelan Plan Shakes Investors

Verizon Communications had been looking to lighten its exposure to Lati
        Now, it probably wishes it had disconnected its Latin lines eve
        The company could possibly lose up to several hundred million d
        Investors reacted with alarm here and in markets in the United
        Markets across Latin America declined Tuesday, but the drop was
        ''It has not turned into a widespread contagion,'' said David R
        Still, investors sold shares in American, Argentine and Mexican
        Owners of Venezuelan steel, banking, cement and hotel companies
        ''Chávez is deepening his revolution, but in doing so will he f
        ''It doesn't seem like the government has thought this project
        Tony Snow, a White House spokesman, said on Tuesday, ''National
        Mr. Chávez further intensified worries with his request for vas
        ''There are many ways in which the Venezuelan government could
        In April, Verizon agreed to sell its 28.5 percent stake in Comp
        Since taking power eight years ago, President Chávez has impose
        Stopping short of Mexico's nationalization of foreign oil compa
        At the very least President Chávez's pledges Monday to national
        Mr. Chávez has led that reversal, which has taken hold to one e
        ''We have said for a while that if you are operating a domestic
        Still, Mr. Wizman notes Mr. Chávez cannot afford to seize asset
        ''He is unlikely to give a fair price, but it won't be outright
        Not all investors are as sanguine, and two American companies,
        The abrupt policy and market jolt in Caracas came amid surging
        Investors note that even with the recent troubles in Russia, Ve
        Some investors and business experts expressed optimism that whi
        Others even hoped he would reverse his intentions, or at least
        A sharp fall in the shares of Electricidad de Caracas, a utilit
        Mr. Chávez's pronouncements prompted expressions of awe and sur
```

Show: Entity Tree Frequency: Mentions
TopN: 50 Sort By: A-Z

america astor_place barinas beijing bolivia brazil brooklyn california cantv caracas
chavez_moves china clinton_street congress cuba east_village england
g.e germany greenwich_village hugo_chavez iraq lafayette_street latin_america
lincoln_center lower_east_side manhattan metropolitan_opera_house mexico
middle_east n.j national_assembly new_york nicaragua paraguay rctv russia scott_ells
soho supreme_court texas u.s united_states united_states_embassy uruguay
venezuela wal-mart washington west_village yaracuy_state

Figure 3: VizLinc: user interface



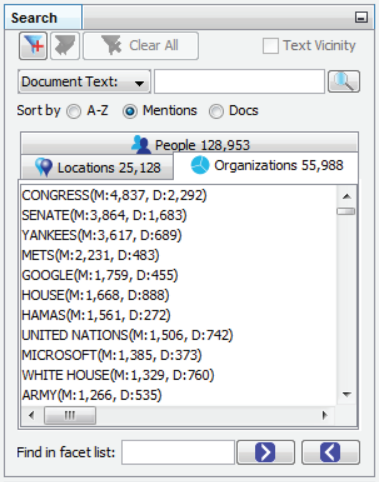
1815839.xml.txt.txt

```
Angeles, Trying to Atone For Water Sins

 A Century Later, Los Angeles Atones for Water Sins

fall short of a feel-good sequel to ''Chinatown,'' the movie based on the notorious, somewhat shady water grab by Los Angele
 But in one of the largest river restoration efforts in the West, water is again flowing along a 62-mile stretch of the Owe
 That part of the river had been left mostly drained when upstream water, fed by snowmelt from the towering Sierra Nevada, 
 The restored flow is among several long-awaited steps the city is taking to help make amends for the environmental conseque
 Los Angeles agreed in December to expand efforts to control toxic dust storms that erupt from what is left of the lake, a 
 The lake's salty, mineral-laced basin has been the largest single source of particulate pollution in the country. It looks
 To restore the river, Los Angeles built automated gates at the point where the river veers into the aqueduct. The gates st
 Much of the water eventually returns to the aqueduct, though some of it is being used for lake irrigation and other projec
 Environmentalists here say they are keeping an eye on Los Angeles for backsliding, but they acknowledge that the new effor
 As winds whipped across Owens Lake on a recent afternoon, Mike Prather of the Owens Valley Committee, which along with the
 ''This work will bring back more and more of them,'' Mr. Prather said, savoring the twist in the battle that means water o
 ''It's Owens Valley's turn to stick its straw in L.A.'s water,'' he said.
 Court rulings and the threat of legal action have largely forced Los Angeles's hand in dealing with its past water moves, 
 H. David Nahai, president of the board that oversees the Los Angeles Department of Water and Power, said Los Angeles was l
 ''We can't change the past, but we can shape the future,'' said Mr. Nahai, one of five board members appointed by Mayor An
 Susan Cash, the chairwoman of the Board of Supervisors of Inyo County, where the Owens Valley is located, said animosity t
 The underhanded moves, as chronicled by historians, included city representatives posing as ranchers as they bought up pro
 Water from the valley made possible the growth of what became the nation's second-largest city. But people in the valley h
 Officials here have argued that the water diversion undercut the potential for growth. But others say that such prospects 
 Los Angeles's policy of allowing public access to much of its land and the fact that many people here have worked for the 
 ''The fact is,'' she said, ''we are in a marriage with no annulment in the near future, so we have to find a way to work t
 Inyo officials said the city's projects could inspire more tourism, the only real economic activity in this dry, high-dese
 ''We have recreational users now but not to the extent it can be once the river is flowing and there is sufficient water f
 The long-promised river restoration is a $24 million project, compensation won from a lawsuit by environmental groups over
 The work on the lake, scheduled to be completed by 2010, will irrigate or otherwise control dust over 43 square miles.
 The improvements result from an agreement the city signed with the local air pollution control regulator in 1998 that sets
 A water department spokeswoman in Los Angeles, Carol Tucker, said ratepayers would see relatively modest increases in thei
```

Figure 5: Document Viewer



CONGRESS(M:4,837, D:2,292)
SENATE(M:3,864, D:1,683)
YANKEES(M:3,617, D:689)
METS(M:2,231, D:483)
GOOGLE(M:1,759, D:455)
HOUSE(M:1,668, D:888)
HAMAS(M:1,561, D:272)
UNITED NATIONS(M:1,506, D:742)
MICROSOFT(M:1,385, D:373)
WHITE HOUSE(M:1,329, D:760)
ARMY(M:1,266, D:535)

Figure 6: Search view. Here we show the most mentioned organizations in the New York Times dataset as extracted by VizLinc's Ingestion Tool.

the current query. If there is no query the counts correspond to the whole data set. All lists can be sorted alphabetically, by decreasing mention count, or decreasing document count. The search-related features of this view will be covered in section 6. Figure 6 shows the list of organizations extracted from the New York Times data set (see section 7.1) sorted by mention count.

5.4 Map

The *Map* view places the locations present in the working set on a geographic map. A small circular waypoint is drawn for each location. Users can navigate the map by zooming and panning. The color or alpha value of each circle can represent either the mention or the document frequency of the corresponding location. Adjusting the alpha value of waypoints based on frequency is particularly useful when there are a large number of locations in the working set. The most frequent locations become clearly visible whereas locations with few mentions/documents fade into the background.

5.5 Graph

The *Graph* view shows the co-occurrence network of all the people mentioned in the working set. Nodes in the graph represent person entities and edges represent document co-occurrence between the linked entities. This is a direct visualization of the graph generated during data ingestion.

VizLinc contains all of Gephi's visualization, analysis, and exploration capabilities In addition, we have made some useful graph analytics accessible through the *Graph Tools* view. The following sections describe those analytics.

Node Centrality

The centrality of a node measures its relative importance within a graph. In the context of VizLinc, centrality can be an indicator of how important a person is in the social structures described in the working set. We have included two centrality metrics: Eigenvector Centrality and PageRank[6]. Both metrics are based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. Users can choose to represent the nodes' centrality score through modifying their size and/or color.

Clustering

The clustering feature groups related nodes in a graph and colors them accordingly. Clustering is based on the *InfoMap* algorithm which attempts to find community structure based on the flow of information in the graph.[8]

N-Hop Network

Highlighting a node in the graph and clicking on the *1-Hop Network* button displays a network consisting of the selected node, all of its neighbors, and all the edges that exist between them. Note that this graph would not necessarily contain all the people in the working document set as normally as the "seed" node does not need to be one of the terms in the search query. Similarly, clicking on the *2-Hop Network* button would generate and display a graph containing all the neighbors of the nodes in the 1-hope network and the edges between all of them.

5.6 Word Cloud

The Word Cloud provides an aggregated view of the most frequent entities. The canonical names of the N most frequent entities are laid out in a grid and their font size is adjusted so that it is proportional to their mention or document count.

6. SEARCH

So far, we have discussed how VizLinc can be used to ingest documents containing text, summarize the entities mentioned in those documents, and visualize their co-occurrence patterns and geographical placement. In this section we will talk about how VizLinc can direct users to relevant sections of a document through searching.

A search query acts as a document filter and can contain one or more terms. Documents that match all the terms in the query are kept in the working set whereas those that don't are eliminated. As a result, all views are updated to display only those entities present in the new working set. Narrowing down the working set and updating all views

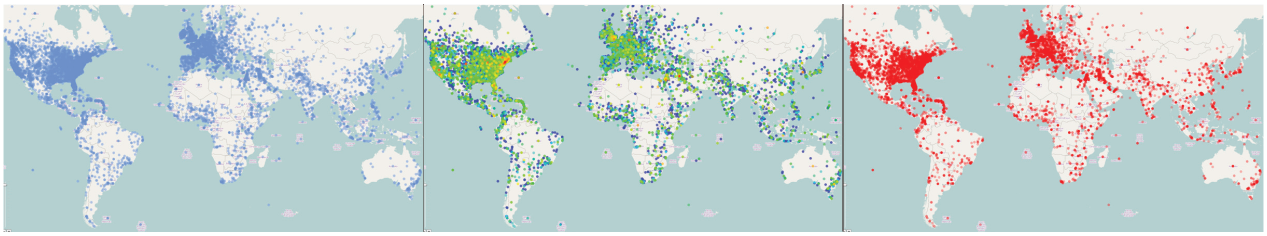
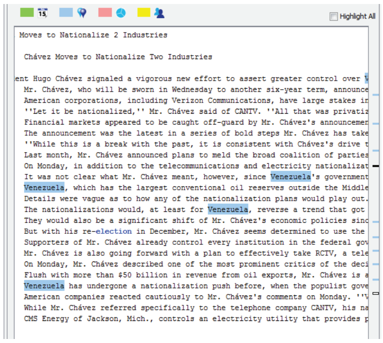


Figure 7: Map showing three different parameter settings: a waypoint per location (left); color scale representing the location's mention count, where blue correspond to the lowest value and red to the highest (center); and alpha value representing mention count, where the highest frequency locations are rendered solid and the lowest frequency locations are not drawn.



☐ Highlight All

```
Moves to Nationalize 2 Industries

 Chávez Moves to Nationalize Two Industries

ent Hugo Chávez signaled a vigorous new effort to assert greater control over V
 Mr. Chávez, who will be sworn in Wednesday to another six-year term, announce
 American corporations, including Verizon Communications, have large stakes in
 ''Let it be nationalized,'' Mr. Chávez said of CANTV. ''All that was privatiz
 Financial markets appeared to be caught off-guard by Mr. Chávez's announcemen
 The announcement was the latest in a series of bold steps Mr. Chávez has take
 ''While this is a break with the past, it is consistent with Chávez's drive t
 Last month, Mr. Chávez announced plans to meld the broad coalition of parties
 On Monday, in addition to the telecommunications and electricity nationalizat
 It was not clear what Mr. Chávez meant, however, since Venezuela's government
 Venezuela, which has the largest conventional oil reserves outside the Middle
 Details were vague as to how any of the nationalization plans would play out.
 The nationalizations would, at least for Venezuela, reverse a trend that got
 They would also be a significant shift of Mr. Chávez's economic policies sinc
 But with his re-election in December, Mr. Chávez seems determined to use the
 Supporters of Mr. Chávez already control every institution in the federal gov
 Mr. Chávez is also going forward with a plan to effectively take RCTV, a tele
 On Monday, Mr. Chávez described one of the most prominent critics of the deci
 Flush with more than $50 billion in revenue from oil exports, Mr. Chávez is a
 Venezuela has undergone a nationalization push before, when the populist gove
 American companies reacted cautiously to Mr. Chávez's comments on Monday. ''V
 While Mr. Chávez referred specifically to the telephone company CANTV, his na
 CMS Energy of Jackson, Mich., controls an electricity utility that provides p
```

Figure 8: Document content showing the matches to the query `String:elections & Location:Venezuela`

accordingly constitutes the basis of discovering patterns and relevant information in VizLinc.

Not only can users target a sub-set of all documents but they can also quickly navigate to the sections within those documents where the target entities are mentioned. This is done by opening one of the documents in the working set. The Document Viewer then shows the content of that document and highlights all the instances of the query terms using different colors for each term type. In addition, color-coded markers for each line containing a match are shown along the right side of the Document Viewer for quick access to the relevant document sections (see figure 8).

In the following sections, we discuss the two search features VizLinc offers.

6.1 String Search

String search refers to the process of finding all the documents that contain a specific string. *Lucene* is used both for indexing the text during ingestion and to search the documents through VizLinc. Matching documents become the working set and all views are updated to reflect the entities mentioned in them. The nature of this type of search implies that its results could mixed documents referring to different entities. For instance a search for the string "washington" will return documents mentioning Washington D.C. (location), George Washington (person), and Washington State University (organization).

6.2 Entity Search

An entity-based search query contains one or more named entities (i.e., locations, persons, and organizations) and returns all the documents that mention those entities. This type of search differs from string search in that a query entity could resolve to several different strings if they are all different ways to mention the same entity. This is the result of within/across document co-referencing during ingestion. For instance, a search for Person:John M. Smith might resolve to mentions "John Smith", "John", or "Mr. Smith". This feature might represent a significant advantage over string searching if users have a particular entity in mind. For instance, if we are interested in those documents that mention the state of Washington, a search for Location:Washington will exclude documents that mention President Washington and not the location.

There are many ways to execute an entity search in VizLinc. In the Search view, users can select an entity from the list and click the *Add Filter* button on the view's toolbar. Alternatively, users can drag the entity name from the list and drop it in the *Query* view. Entities can also be added to a query from the graph, map or word cloud by right-clicking on their representation (node, waypoint, or label, respectively) and selecting *Add to Query* from the context menu.

7. CASE STUDY

In this section, we present a few hypothetical use cases based on the results obtained by processing and analyzing real data with VizLinc. These use cases and results should give readers some insight about the type of patterns and information VizLinc can reveal. All hypotheses drawn from our visualization and stated in this section were later confirmed by examining the content of the appropriate documents.

7.1 New York Times Articles (2007)

The New York Times Annotated Corpus [9] contains over 1.8 million articles written and published by the New York Times between January 1, 1987 and June 19, 2007 with article metadata. The data set we processed is a subset composed of all articles written in the year 2007. This subset is composed of 39,953 documents containing thousands of entities.

Figure 9 shows the map, graph, and word cloud for the whole New York Times 2007 data set. The map shows that most location mentions are concentrated in the U.S.A. and Western Europe. It is hard to make sense of the graph when it contains so many nodes. Upon closer examination thanks to VizLinc's graph navigation and clustering features, we can see that clusters belong to different categories. Politicians, artists and sports personalities all have their own clusters. The word cloud shows the 50 most salient terms in the data set. Not surprisingly, locations New York, United States, New York City, Iraq, Manhattan and Washington are heavily mentioned. Organizations like Congress, Senate, Yankees (New York Yankees), and Google also form part of the list of most mentioned entities.

We will rely on a hypothetical use case and potential action path to illustrate VizLinc search capabilities on the New York Times data set. Let us say that we are interested in elections around the world. A first approach would be to do a search for the string "elections". The working set decreases from nearly 40,000 to 834 documents that contain that string. The list of documents can now be sorted by the total number of mentions and we could browse the contents of the top hits. Instead, we will take a look at the map and location list to see what locations co-occur the most with the term "elections". The reader should keep in mind that, after executing a query, all views are updated to show different visualizations of the content of the matching document set only i.e., the *new working set*. The entity list in the *Search* view shows that "Iraq", "United States", and "Israel" are the most mentioned locations in conjunction with "elections". Examining the map shows activity in many other parts of the world including the major countries in South America. Let us say that Venezuela piques our interest, so we add `Location:Venezuela` to the query from the map view.

The working set now contains 12 documents that could be browsed within minutes if so desired. The resulting graph shows the people mentioned in these documents and it is much more suitable for visual analysis than the original one.

To get a sense of the importance of each individual in the working set as described by the co-occurrence relation defined in previous sections, we re-size the nodes in the graph according to their centrality score. Also, we can cluster this new sub-graph to reveal any community structures present. Figure 10 shows part of the resulting graph.

The graph suggests that one of the most central people is Hugo Chavez. Hugo Chavez was the president of Venezuela in 2007 and had been re-elected the previous year. This is not new information but it demonstrates VizLinc's ability to find central people with respect to some user-defined context. Clustering resulted in three major communities; a subset is shown in Figure 10. Upon examination, it can be noticed that the three clusters illustrated group three different types of actors: USA political and media figures (top-left), South American political figures (top-right) and artists (bottom-left). From this point on, if we were interested in the sentiment and opinions of U.S. politicians towards the government of Venezuela we could add members of that community to the query. If what is relevant to us is stories about South American leaders and the government of Venezuela we would add members of the second community to my query and examine the resulting documents. Authors and entertainers appear in the graph due to spurious co-occurrences in articles that contain lists spanning a variety of unrelated topics.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced VizLinc and described how it combines information extraction, graph analysis, and geo-location for visualization and exploration of text corpora. We have also presented a case study, centered on a compilation of articles from the New York Times, to demonstrate VizLinc's features.

Now that we have achieved our principal goal of creating a complete framework for data ingestion, visualization, and exploration, our future work will focus on making each component more generic and robust. Modules such as the ones that generate the graph and perform coreference resolution, yielded reasonable results on the data for which VizLinc was initially intended. However, these modules turned out to be rather simplistic for most of the text genres we have tested so far. Multiple-term searches could also be improved by restricting the distance at which both terms can appear in a document. This will avoid documents in which terms co-occur but are in fact unrelated. Expanding queries to support "and" and "or" operations is also a subject for future work. With a platform in place, we can now take a task centric approach and assess whether the techniques and user interactions VizLinc enables are appropriate to the successful completion of a particular task. Finally, we understand that user-defined algorithms and entity types will be required to analyze certain data sets efficiently. Therefore we would like to include a mechanism that would allow users to add these custom components with ease.

9. REFERENCES

[1] M. Bastian, S. Heymann, M. Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. *ICWSM*, 8:361–362, 2009.

[2] T. Boudreau, J. Tulach, and R. Unger. Decoupled design: building applications on the netbeans platform. In *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*, pages 631–631. ACM, 2006.

[3] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[4] J. R. Harger and P. J. Crossno. Comparison of open-source visual analytics toolkits. In *IS&T/SPIE Electronic Imaging*, pages 82940E–82940E. International Society for Optics and Photonics, 2012.

[5] A. Özgür, B. Cetin, and H. Bingol. Co-occurrence network of reuters news. *International Journal of Modern Physics C*, 19(05):689–702, 2008.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[7] M. A. Rodriguez and P. Neubauer. The graph traversal pattern. *arXiv preprint arXiv:1004.1001*, 2010.

[8] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. In *Proceedings of the National Academy of Sciences*, page 1118, 2001.

[9] E. Sandhaus. The new york times annotated corpus ldc2008t19. *Linguistic Data Consortium*, 2008.

[10] B. Wright, J. Payne, M. Steckman, and S. Stevson.

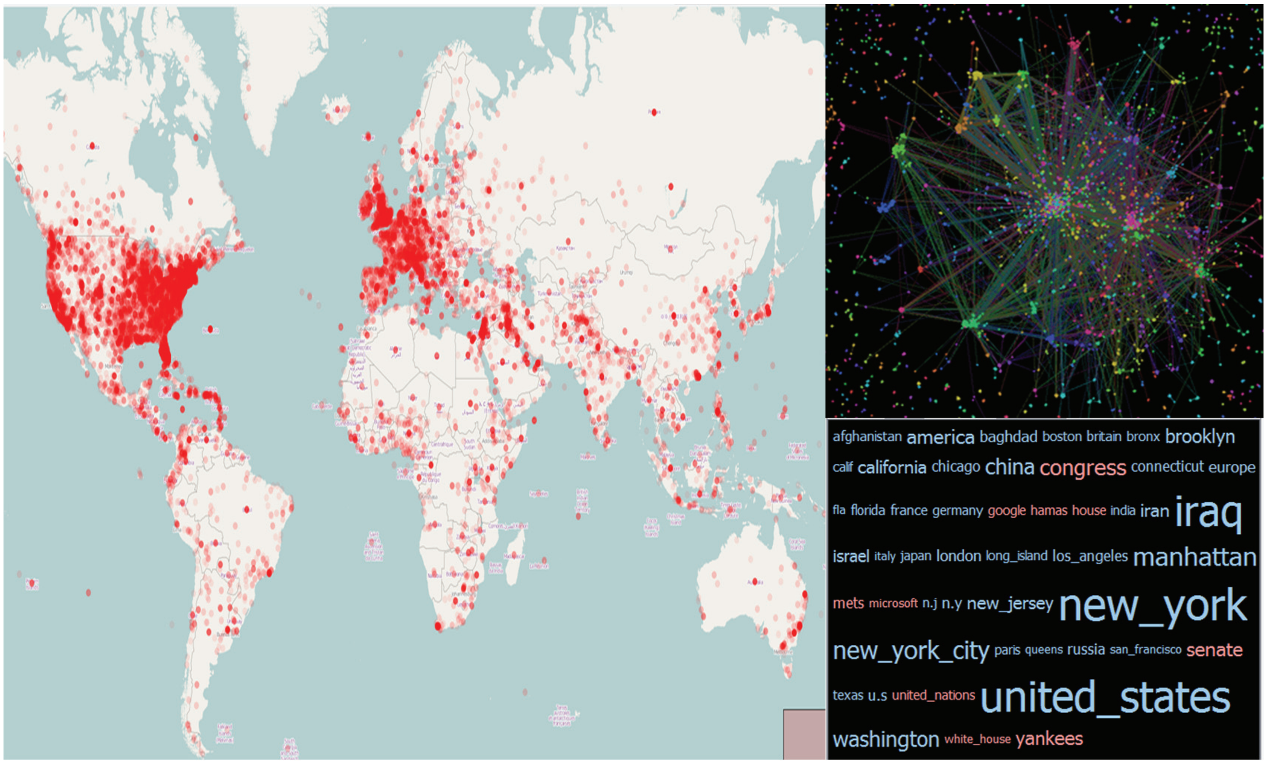

afghanistan america baghdad boston britain bronx brooklyn
calif california chicago china congress connecticut europe
fla florida france germany google hamas house india iran iraq
israel italy japan london long_island los_angeles manhattan
mets microsoft n.j n.y new_jersey new_york
new_york_city paris queens russia san_francisco senate
texas u.s united_nations united_states
washington white_house yankees

Figure 9: Map, co-occurrence graph, and word cloud for the New York Times 2007 data set.

Palantir: A visualization platform for real-world analysis. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*. IEEE, 2009.

[11] L. Zhang, A. Stoffel, M. Behrisch, S. Mittelstadt, T. Schreck, R. Pompl, S. Weber, H. Last, and D. Keim. Visual analytics for the big data eraâĂŤa comparative review of state-of-the-art commercial systems. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 173–182. IEEE, 2012.

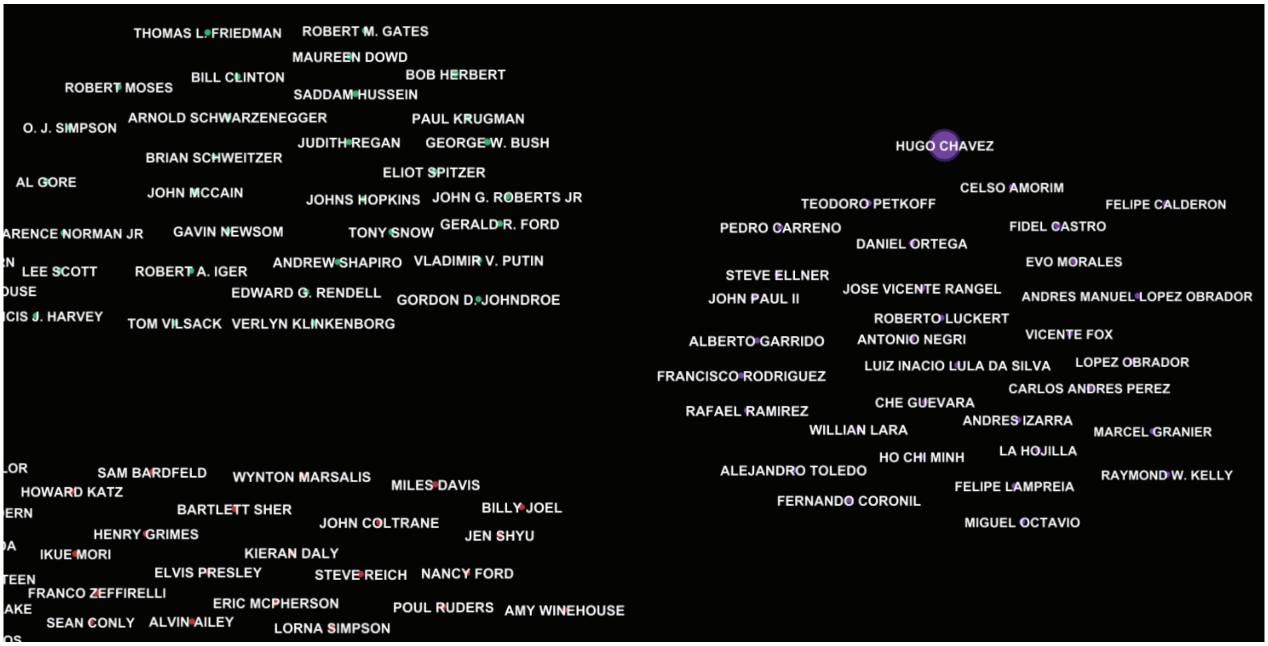


Figure 10: Section of the graph after executing the query `String:elections & Location:Venezuela` The nodes have been re-sized according to their centrality scores and clustering has been run on this sub-graph. Clusters shown group three types of people: USA political figures (top-left), South American political figures (top-right) and artists (bottom-left)

Better Logging to Improve Interactive Data Analysis Tools

S. Alspaugh
University of California,
Berkeley
alspaugh@eecs.berkeley.edu

Archana Ganapathi
Splunk, Inc.
aganapathi@splunk.com

Marti A. Hearst
University of California,
Berkeley
hearst@berkeley.edu

Randy Katz
University of California,
Berkeley
randy@eecs.berkeley.edu

Abstract

Interactive data analysis applications have become critical tools for making sense of our world. We present a set of recommendations to improve the quality and quantity of user activity data logged from interactive data analysis systems. Such data is invaluable for improving our understanding of the data exploration process, for implementing intelligent user interfaces, for evaluating data mining and visualization techniques, and for characterizing how the broader ecosystem of data analysis tools are used in practice.

Currently, much of the data logged by data analysis systems is intended for the purpose of debugging and system performance monitoring, not for understanding user behavior. As a result, researchers have to rely on labor-intensive techniques for extracting useful information from low-level event streams, or on collecting data through observation, interviews, experiments, and case studies.

We present recommendations – derived from personal experience as well as examples from the literature – for logging user activity in interactive data analysis tools, to ensure that better information is collected, and ultimately, to enhance human problem-solving abilities and speed the pace of discovery. We illustrate these recommendations using examples from three widely-used but distinct systems for analyzing data: Tableau, an interactive visualization product, Excel, a spreadsheet application, and Splunk, an enterprise log management and analysis platform.

1. INTRODUCTION

Despite longstanding research interest in data exploration and automation, there is a scarcity of automatically logged, high-quality activity records of data exploration activities at an appropriate level of granularity, available for study by researchers and developers. In our experience, one reason is that much of the data logged by data analysis systems is intended for the purpose of debugging and system performance monitoring, not for understanding user behavior [12, 16, 28]. As Horvitz et al. note in their paper on Bayesian user modeling, "...it is critical to gain access to a stream of user actions. Unfortunately, systems and applications have not been written with an eye to user modeling." [16] As a result, much effort has been devoted to devising ways to extract useful usability information from UI events, as reviewed by Hilbert and Redmiles in their extensive survey on the topic [14].

As an alternative to analyzing automatically logged user actions, much of the research on understanding and improving how users analyze data relies either on author intuition born of first-hand experience [4, 6, 17, 33] or on observational studies using manually-recorded and synthesized information that is hard to share and compare [18, 19]. This lack of automatically logged activity records hinders research into improving tools for data exploration and analysis.

Currently, to understand what users are doing, build user models to improve interfaces, yield predictions about user actions, and make recommendations to users, researchers have few options. One option is to take great pains to extract high-level information out of low-level event logs [12, 14, 16, 28]. Another option is to manually observe user behavior, interview experts, read through the literature, and synthesize all of their observations "by hand," then encode this synthesized information into their tools [1, 17, 22, 27].

If user actions were instead encoded in a machine-digestible format at an appropriate level of granularity, researchers could create software to automatically detect these patterns, much like clickstream analysis and webmining [3, 11, 13, 32]. As Hilbert and Redmiles conclude, "more work is needed in the area of transformation and data collection to ensure that useful information can be captured in the first place, before automated analysis techniques ...can be expected to yield meaningful results." [14]

This position paper encourages better practices for logging data to enable studying user behavior in interactive data and visualization systems. Our recommendations are based primarily on our personal experiences trying to build user models using traces from an enterprise-scale log analysis system, but also on a review of the literature and conversations with industry personnel. Our recommendations can be summarized as follows:

- Design to capture high-level user actions.
- Capture provenance of all events.
- Observe intermediate user actions.
- Obtain the analyzed data's metadata and statistics.
- Work towards log standardization.
- Collect user goals and feedback.

In Section 2 we first provide motivating examples of research and applications that could be enabled if better user activity data were logged from interactive data analysis systems. We then give our recommendations for collecting this improved data in Section 3. Section 4 discusses ramifications and other issues.

2. WHY DO WE NEED BETTER LOGGING?

This section motivates the need for better logging of interactive data analysis systems: characterizing the exploration process, implementing intelligent user interfaces, evaluating analysis tools and interfaces, and understanding the analysis ecosystem as a whole. These purposes are not only of interest to researchers who wish to understand these topics, but also to industry practitioners, who can use this information to design their products to make them better suited to their users. The section concludes with examples from the related area of web behavior mining, which is further along and could hold useful lessons.

Characterizing the exploration process: Two interesting theoretical models of the data exploration process have been put forward recently that would benefit greatly from better logging as proposed here.

(1) De Bie and Spyropoulou propose a formalization to unify the concept of interestingness and help automate data exploration across a range of data mining techniques [8]. In their formalization, users express interests and beliefs about the data in terms of mathematical patterns and probability distributions. To put this formalization into practical use, rather than asking end-users to specify these directly, data exploration tool developers will likely want to determine the beliefs and patterns users find useful for a particular domain and then expose those to the user in a more easily interpretable form. Doing so would require detailed and annotated records of exploratory activities in a wide variety of scenarios.

(2) As another example, Perer and Shneiderman propose a framework, called SYF (Systematic, Yet Flexible), for guiding users through data exploration [27]. It operates within interactive analysis and visualization interfaces, guiding users by providing an overview of recommended analysis steps, suggesting unexplored states, and allowing users to annotate and share a record of their activities. To implement SYF within a given tool, developers must register their recommended exploration steps with the SYF framework. To derive these systematic steps, Perer and Shneiderman suggest that developers try "[i]nterviewing analysts, reviewing current software approaches, and tabulating techniques common in research publications.". An additional useful approach for establishing these steps would be to mine detailed activity records from data analysis and visualization systems.

Implementing intelligent user interfaces: SYF is one example of an intelligent user interface. These assist the user by offloading some of the complexity in working with the tool at hand, often by automated means. Other examples include adaptive or adaptable [5, 24], predictive [29], and mixed-initiative interfaces [15], as well as automated user assistants [16, 23]. Automated interfaces often rely on statistical models of user behavior and thus require accurate accounting user actions at a level that corresponds to the variables being modeled.

For example, Wrangler has a mixed-initiative interface that makes suggestions to help users clean their data based on frequencies of user actions [17]. Wrangler was originally based on a transformation language with a small number of operators. To identify this list of transforms and pair them with interface gestures, the authors were able to capitalize on their extensive first-hand experience, as well as prior work on languages for data cleaning. However, for data exploration rather than data cleaning, it is not clear what set of transforms and visualizations should be supported. Previous work has relied on author intuition and experience with particular situations to determine what actions to support [4, 6, 33]. However, these could be better determined by having detailed activity records from data exploration and visualization tools with direct-manipulation interfaces, logged at an appropriate level of granularity [12].

Evaluating analysis tools and interfaces: More generally, researchers and practitioners evaluate interfaces to understand user behavior, performance, thoughts, and experience, compare design alternatives, compute usability metrics, and certify conformance with standards [14]. To achieve these goals using events logged from current UI systems, researchers have devised a wide range of techniques: synchronizing data gathered from different sources, transforming, comparing, summarizing, and visualizing event streams, and abstracting low-level log events into high-level modeled events.

An alternative to these automated techniques is to perform carefully controlled laboratory evaluations or focused long-term studies of specific tools in isolation [18, 31, 20, 26, 19]. These usually involve watching videos of study subjects performing a task, interviewing subjects about their experience, and evaluating how well they performed the task.

While this research is valuable, some drawbacks of these techniques are that they don't scale well, they generate results that are not amenable to comparison or combination with data from other studies, and the process of recording the data is too open to subjective interpretation. High-quality automatically logged interaction data would circumvent each of these problems, although at the expense of missing the big picture that these techniques provide.

Understanding the analysis ecosystem: In addition to improving upon individual tools and interfaces, developers and researchers want to understand the entire data analysis pipeline. In practice, users leverage multiple tools to explore and visualize their data depending on their needs. For example, a data scientist might use Hadoop and R for statistical work. A product manager might create their visualizations in Tableau using web analytics reports generated in Splunk. A customer analyst might extract numbers from Salesforce to crunch in Excel. To get a complete picture of each of these user's exploration and visualization needs, it would help greatly to be able to track their activities across each of the tools they use. However, researchers are currently limited to gathering cross-tool data via long-term time-intensive interviews with industry practitioners [25, 30].

Inspirations from other domains: When it comes to automating data collection about user behavior and optimizing interfaces in response, work in mining website interaction data and search engine clickstreams may help point the way forward. Much effort has gone into designing tools to collect extensive data about web users and analyze it in increasingly sophisticated ways. A full survey of such work is beyond the scope of this paper; here we provide a few examples for illustration. Researchers have designed advanced and unobtrusive tracking software that can be implemented using standard web technologies [3]. Information gleaned from this type of data can be used to infer user goals to determine, for instance, if the user is interested in purchasing a product or merely researching it [13]. Such data can also be used as implicit feedback on the presented interface, for example, regarding the quality of a ranking returned by a search engine [11].

We hope that in the near future, interactive data analysis and visualization users will benefit from similar efforts. We contribute to this vision through our detailed recommendations regarding what data to collect from interactive data analysis and visualization tools.

3. RECOMMENDATIONS

In this section we detail a set of proposals for collecting data from interactive data analysis and visualization systems to support the research and applications described in Section 2. We illustrate our proposals with examples from three systems: Tableau, an interactive data visualization product, Excel, a spreadsheet application, and Splunk, an enterprise log management and analysis platform.

3.1 Overview of logging basics

Here we review the basic types of information that should be logged.

Events: Units of information in a log are often called events, even if they weren't generated from an event-driven program. However, GUIs and other interactive programs are usually event-driven.

An event in a log is some piece of information that is recorded any time work of interest is run on the system. Work of interest could include functions called, queries run, GUI handlers triggered, threads executed, and so on. Exactly what information is logged for each event and the format it appears in varies widely – it may include information such as function parameters, execution duration, caller, and source code location. For example, an event logged by Tableau is shown in Table 1. Such events are typically logged for debugging and performance monitoring purposes. Later we discuss specifically what types of events and associated information should be logged for user modeling.

Timestamps: Events should always be accompanied by a timestamp that describes the date, time, and timezone information. Timestamps are important for understanding the order and rate of events but are not always reliably accurate reflections of when an event truly occurred. This is usually not a problem when dealing with logs from a single machine but can be extremely challenging to deal with in a distributed setting. A discussion of how to deal with this problem is beyond the scope of this paper; we refer the reader to other work [14].

User ID: Ideally each event should be connectable to information about the user "responsible" for initiating the event, in the sense that their interaction with the program "caused" the event. For some events, the "user" responsible may be the system itself, for example, in the case of garbage collection. In general, determining causality is not trivial, but for the events of interest for user modeling, it should be straightforward.

Version and configuration: It is critical to provide some information that ties each event recorded to metadata about the version and configuration of the interface that generated that event. This is important because exactly what information is logged and the format it is logged in tends to change across versions and configurations. Without this information, it can become unnecessarily difficult to parse the logged data, and ambiguities may be introduced. Ideally, even changes to minor features of the interface would be versioned, to facilitate A/B testing.

3.2 Design to capture high-level user actions

As Horvitz et al. state, "a critical problem in developing probabilistic and decision-theoretic enhancements for user interface applications is establishing a link between user actions and system events." Thus, it is important not only to log the actions that happen in the system but the actions that the user takes. In other words, log the operations that are applied to the data at the level of the user's perspective, not just the executed code that the user's command calls. To model user behavior and cognition, we are primarily interested in the former, but the systems engineers who typically write the logging code are primarily interested in the latter.

In some cases, this may be a conceptually simple fix. For example, Tableau had [1] functionality "called *Show Me Alternatives* and *Show Me*, which are respectively a dialog of commands that automatically build views from scratch and a button that is a shortcut to the default choice for the dialog" [21]. In their paper describing this functionality, the authors have a discussion about their efforts to evaluate their interface using the Tableau logs. They note that, "the log files do not differentiate between *Show Me* and *Show Me Alternatives*. These commands are implemented with the same code and the log entry is generated when the command is successfully executed." This exactly describes the problem with logging events from the perspective of what the system executes versus from the perspective of the user taking the action. It complicates the work of trying to understand how users are interacting with a tool and especially complicates trying to build statistical models of user behavior.

In other cases, logging user-level actions may be more difficult. For example, the authors of the Lumière project to build an automated assistant in Excel found transforming system events into modeled events to be challenging [16]. To establish the link between low-level atomic events and the higher-level semantics of user actions they built a special events system to analyze the atomic event streams and map them into higher-level observations.

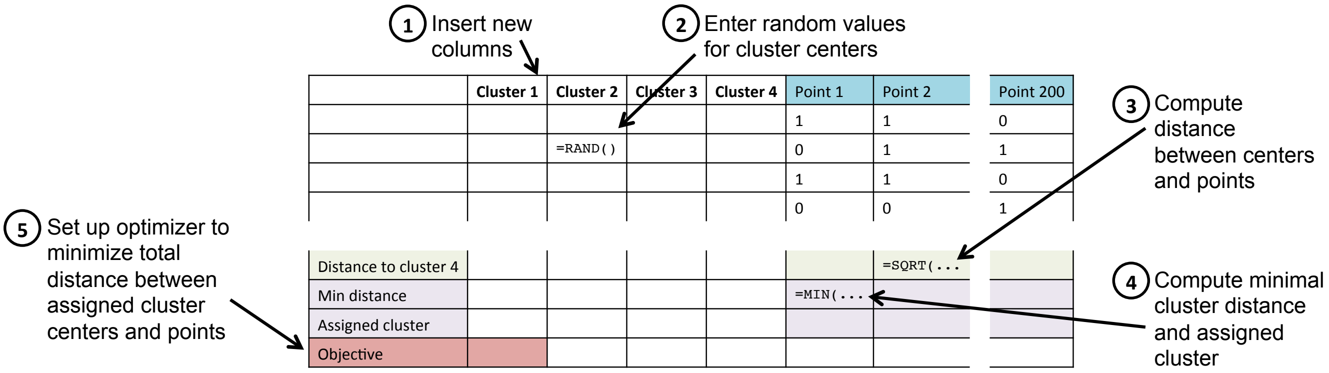
Figure 1 and Figure 2 illustrate this problem graphically, using Excel as an example. Figure 1 shows the steps required to perform k-means clustering on data in Excel from a user's perspective [10]. The user may be doing this in order to segment their customer base into different markets for the purpose of releasing targeted advertising. Market segmentation is the user's intention, as shown in the uppermost level of Figure 2. Ideally, we could record that the user is performing clustering – this is the user's task, shown in the second level. However, in reality, the best information we can capture is the stream of the user's activities in the GUI. This is the third level of information shown in Figure 2. In practice, the information captured tends to be low-level system events – the lowest level of Figure 2. Using events from a lower-level to infer what action is being taken at the higher-level can be painful if insufficient information is recorded. It may rely on human-defined rules or on statistical inference, as in the events system of the Lumière project.

HARVEST is a visual analytics system specifically designed to deal with this problem and capture the provenance of "in-

[1] *Show Me Alternatives* is no longer part of Tableau.

```
2013-09-24 22:06:34.197 (-,-,-,-) 0d08:  UpdateThumbnail called, WorksheetWnd.cpp, line 1685
```

Table 1: Example of an event logged by Tableau. This event records that a function was called, giving its location in the source.


1 Insert new columns
2 Enter random values for cluster centers
3 Compute distance between centers and points
4 Compute minimal cluster distance and assigned cluster
5 Set up optimizer to minimize total distance between assigned cluster centers and points

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Point 1	Point 2	Point 200
					1	1	0
		=RAND()			0	1	1
					1	1	0
					0	0	1
Distance to cluster 4						=SQRT(...	
Min distance					=MIN(...		
Assigned cluster							
Objective							

Figure 1: Shown above are the steps a user takes to perform k-means clustering on their data in Excel. Ideally, the interface would be designed to easily capture the user's task, which is clustering. Barring that, the task may have to be inferred from the sequence of user actions, shown here.

sights" [12]. It does this by exposing interface elements for performing tasks that directly correspond to "semantic actions," like bookmark, brush, filter, and sort. In other words, the interface is designed so that the actions the researchers are interested in tracking must be explicitly indicated by the user. Contrast this to a system that allows the user to select a view with many potential elements of interest at once, so that it is difficult to discern what the user most wanted, or to a system that requires the user to achieve their task through several direct but low-level actions like clicking and dragging, as in the example above. Designing the interface to facilitate easier capturing of high-level user behavior is a useful approach that should be considered wherever possible.

If this is not possible or desirable for a particular application, researchers can rely on techniques for combining sequences of events into high-level tasks [28]. For example, if the system was designed such that the user interacts with it at a relatively low-level (such as clicking and dragging, or entering functions into an interactive shell), it will not be possible to log the user's high-level tasks because those will not be reflected in their activity stream. In this case, it is important to keep the end use case of the interaction data in mind while developing the application and the logging code. Ideally, tool builders would design the system that identifies from low-level events the high-level tasks employed in user models in tandem with the application, rather than as an afterthought.

3.3 Capture provenance of all events

Information about the point in the application where an event occured and how it was issued should be logged along with the event itself. This information reflects the event's provenance or point of origination. The following real-life anecdote demonstrates the importance of this.

Splunk records an event every time it executes a query, as shown in Table 2. The query, highlighted in bold, is an important source of information about the analysis actions users perform on their data. Some of these queries may be written interactively by a human user. The user may issue these queries by typing it into the Splunk shell at their command line interface, or by typing it into the search bar in Splunk's in-browser GUI, or by hitting a button in an application that then issues the query to Splunk's back-end, or by loading a web page in a browser-based dashboard that triggered the query to execute. Other queries may be written by a user once, but then set up by that user to execute programmatically, via a script, for instance. Still other queries may have been issued from system code to fulfill some function of the system, and were written by a programmer of the system, not by a human user of that system. (Usually in this case, the user will be indicated to be the system, as in the example given above.)

To correctly model a user's cognitive state and extract artifacts like user sessions, it is critical that each time a query is logged, it is possible to easily and unambiguously recover this origination or application context information. Otherwise, the record of what actions a user took may be tainted with actions that they did not actually take, or actions that a user actually took may be inadvertently discarded. Our first-hand experience attests that it is extremely difficult to recover this information if it is not captured when the data is originally logged. Statistical learning techniques like clustering and classification can aid in probabilistically identifying origin information post facto, but rather than relying on such techniques, it is better to plan ahead and capture this information when the data is initially logged.

As another example, in Tableau, if a user creates a scatter plot by dragging a variable onto a shelf and receiving the default view versus selecting a scatter plot from the *Show Me Alternatives* menu, it should be recorded not only that the user created a scatter plot but also how the user created the scatter plot i.e., the provenance.

Even then, we have observed in our work with Splunk logs that sometimes users perform actions that (unintentionally) circumvent efforts to accurately model their behavior, for example, by writing external scripts to interact with a browser-

```
09-28-2012 18:28:01.134 -0700 INFO AuditLogger - Audit:[timestamp=09-28-
2012 18:28:01.134, user=splunk-system-user, action=search, info=granted ,
search_id=‘scheduler__nobody__testing__RMD56569fcf2f137b840_at_1348882080_101256’,
search=‘search index=_internal metrics per_sourcetype_thruput | head 100’, autojoin=‘1’,
buckets=0, ttl=120, max_count=500000, maxtime=8640000, enable_lookups=‘1’, extra_fields=‘’,
apiStartTime=‘ZERO_TIME’, apiEndTime=‘Fri Sep 28 18:28:00 2012’, savedsearch_name=‘‘sample
scheduled search for dashboards (existing job case)’’]
```

Table 2: Example of an event logged by Splunk. A query is highlighted in bold. Queries represent useful records of user activity. These queries are written in the Splunk query language, modeled after UNIX pipes and utilities.

based GUI (i.e., a bot). But this should be rare if the system is designed to encourage correct and easy-to-track use, such as providing programmatic access to its analysis capabilities via an API, if that is what users need.

3.4 Observe intermediate user actions

Modelers and researchers of user behavior may also be interested in user activities that are not “submitted” to the system. This includes information such as

- text that a user types in a search box and then deletes and then types again (not just the text that is finally submitted when they hit enter),
- data on where the user’s mouse hovers, and
- interface selections that the user makes that are done client-side and not sent to the back-end.

For example, Splunk provides a browser-based interface for visualization. The bulk of data transformation operations, however, occur on the Splunk servers, which is where the logs are written. In order to capture the full extent of user behavior, such as extracting data, aggregating it, then toggling between a pie chart and a bar chart, it is necessary for the client to send this client-side activity information back to the system to be logged. Otherwise the system will not “see” this activity, since it does not pass to the back-end in the course of normal operation [2].

Developers of automated help systems may be particularly interested in such intermediate behavior because it may indicate confusion on the part of the user. For example, the Lumière project to create an automated assistant in Excel, modeled events such as “menu surfing,” “mouse meandering,” and “menu jitter.”

3.5 Obtain analyzed data’s metadata & stats

If possible, with the user’s permission, metadata and statistics about the data over which the user is operating should be logged. Metadata includes information like schema (column names and data types) and provenance. Statistics includes things like descriptive statistics (describing the empirical distribution of the data), correlations, and measures such as entropy and cardinality. This would allow an inference model that supports an intelligent interface’s predictions and suggestions to incorporate variables that reflect information about the user’s data. It would also allow product managers to identify important user personas and their needs. For example, a company may be able to recognize by tracking this information that 35% of the users of their system use their browser-based GUI to analyze email marketing data specifically, and further observe that these users often follow very similar analysis workflows. This may spur the company to create a specialized product targeted towards these users needs that conveniently encapsulates these workflows. Less hypothetically, Splunk provides on top of its framework targeted “apps” – pre-built dashboards and tools designed for certain types of users with certain data sets. However, these apps are currently designed based on information manually gleaned from extensive interaction with customers, not based on data gathered through Splunk.

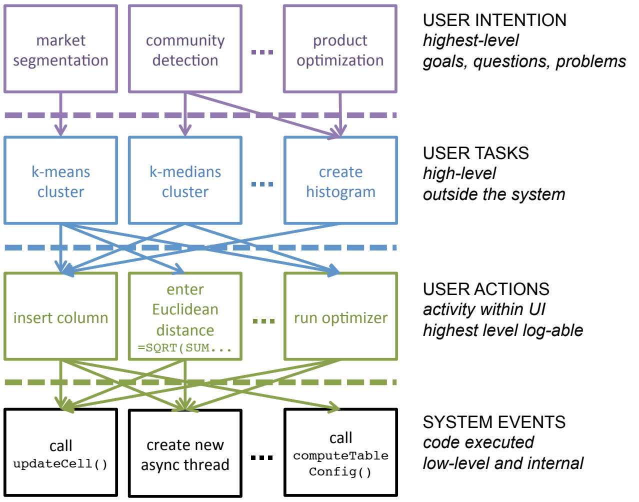


Figure 2: Users intentions motivate their actions, but may be hard to know (top row). When trying to understand user behavior especially with respect to data exploration and visualization, we are often interested in the high-level task the user is performing (second row). In the best case, we can log the actions the user takes via the UI (third row). Sometimes what is logged are low-level system events, which can make it very hard to reconstruct the user’s behavior (bottom row).

We acknowledge that collecting this data is a challenging proposition in many scenarios because users may be unwilling to provide information about their data, which may include operational, propriety, or personally identifying information. The “Show Me” paper by researchers at Tableau discusses the challenge this poses to developers trying to use logs to evaluate interface innovations [21]

3.6 Work towards log standardization

Effort should be made where possible to log information that facilitates combination with and comparison to other systems. This would allow researchers and developers to understand what functionality is missing from existing tools, better characterize portions of the analysis pipeline that are currently less well understood, such as exploration, and identify how best to integrate new data mining techniques into existing workflows. Ideally, there will one day be cross-system instrumentation that would allow researchers to understand the entire ecosystem of data analysis tools and visualization and the roles they play in users' daily workflows. To accomplish this, it may benefit the community to develop an open standard for logging, similar to the Common Information Model, which defines a way for objects in an IT environment and relationships between them to be described so as to facilitate management of systems, networks, applications and services independent of manufacturer or provider [9]. Examples of standard schemas in other domains that may serve as inspiration includes IEC61850 for electric grids and SensorML for sensor data.

3.7 Collect user goals and feedback

The user's goal, or the task they are trying to perform, as well as their position and their level of expertise, are all likely important factors that will likely greatly impact what interface elements an adaptive interface should show or what recommendations should be given. This has long been recognized as important for determining what visualizations to automatically generate for a user [7]. Where possible, information about goals, expertise, and other relevant context could be solicited from the user. However, if this solicitation requires the user do additional work that does not benefit them, it is highly unlikely to be successful. One possible solution could be, for instance, asking the user to select an answer from a list at a natural inflection point in their workflow, to reduce the inconvenience to the user (for example, as is often done when one unsubscribes from a mailing list). More research will be required to determine how to do this in a way that is not annoying and that still yields useful information. The information we recommend collecting in the rest of this paper could facilitate such research.

If an adaptive, predictive, customized, or mixed-initiative interface is implemented that provides suggested actions or tasks to the user, the interface should also provide the user with the opportunity to comment on, rate, rank, and mark as interesting or uninteresting each suggested action, as suggested by Perer and Shneiderman [27]. This data should be recorded to improve the underlying model used to generate the suggestions and can also be provided to the user. Such data also becomes very useful to the user as an artifact of their analysis and exploration process.

4. CONCLUSION

We conclude our recommendations with a brief discussion of implementation considerations. As suggested previously, one reason high-quality user activity records may not be collected from data analysis systems is that often, understanding and modeling user behavior is not a first priority for developers of such applications, who instead focus on logging for system debugging and performance monitoring. Horvitz et al. note that "establishing a rapport with the Excel development team was crucial for designing special instrumented versions of Excel with a usable set of events." [16] Similarly, in our experience with Splunk, we found that some of the interaction data that we were interested in, particularly the visualization activity that occurs on the client side, was not data that the development team had previously needed to log [2].

Hilbert and Redmiles have raised the concern that requiring more data about user behavior "places an increased burden on application developers to capture and transform events of interest." [14] We acknowledge that there will be costs associated with more thorough and high-quality logging of user activity, but we argue that as the examples in Section 2 demonstrate, this effort will be well worth it for the wide variety of applications and research it enables. Such work will ultimately benefit the end-user of data analysis and visualization systems, particularly during data exploration, by enhancing human problem-solving abilities and speeding the pace of discovery.

There are a number of concerns related to collecting data from or about the user, particularly if it is personally identifying information or sensitive operational data from an organization. A full discussion of these concerns and their possible solutions are outside the scope of this paper. We argue though that through careful planning and working to develop trusted collaborations with the users who will be the ultimate benefactors of such efforts, researchers and tool builders should be able to improve their practices for logging interaction data from data analysis and visualization tools. Obtaining permission from users before logging their data, and allowing them to see what data is logged and edit and remove portions that they do not wish to have remain in the logs provides important protections. Policies to permanently remove data after a fixed amount of time, after useful information has been derived from the specifics and placed into general models can further help to protect individual privacy. Some users have already shown themselves willing to share data about their usage and behavior with companies in the interest of improving their user experience and the company's product. As already noted, it is important that such data be collected with users' full knowledge and consent.

5. REFERENCES

[1] Visual analysis best practices. Technical report, Tableau Software, 2014.

[2] Sara Alspaugh, Beidi Chen, Jessica Lin, Archana Ganapathi, Marti Hearst, and Randy Katz. Analyzing log analysis: An empirical study of user log mining. In *Conference on Large Installation System Administration (LISA)*, 2014.

[3] Richard Atterer, Monika Wnuk, and Albrecht Schmidt. Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In *ACM Conference on World Wide Web (WWW)*. ACM, 2006.

[4] Abraham Bernstein et al. Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification. *Knowledge and Data Engineering*, 2005.

[5] Andrea Bunt, Cristina Conati, and Joanna McGrenere. Supporting interface customization using a mixed-initiative approach. In *ACM Conference on Intelligent User Interfaces (IUI)*, 2007.

[6] Stephen Casner. Task-analytic approach to the automated design of graphic presentations. *Graphics (TOG)*, 1991.

[7] Stephen Casner. Task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics (TOG)*, 1991.

[8] Tijl De Bie and Eirini Spyropoulou. A theoretical framework for exploratory data mining: Recent insights and challenges ahead. pages 612–616, 2013.

[9] Inc. Distributed Management Task Force. Common information model. `http://www.dmtf.org/standards/cim`.

[10] John Foreman. *Data Smart: Using Data Science to Transform Information into Insight*. John Wiley and Sons, Inc., 2014.

[11] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems (TOIS)*, 23(2):147–168, 2005.

[12] David Gotz et al. Characterizing users' visual analytic activity for insight provenance. *Information Visualization*, 2009.

[13] Qi Guo and Eugene Agichtein. Ready to buy or just browsing?: detecting web searcher goals from interaction data. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2010.

[14] David M Hilbert and David F Redmiles. Extracting usability information from user interface events. In *ACM Computing Surveys (CSUR)*, 2000.

[15] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Human Factors in Computing Systems (CHI)*, 1999.

[16] Eric Horvitz et al. The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Conference on Uncertainty in Artificial Intelligence*, 1998.

[17] Sean Kandel et al. Wrangler: Interactive visual specification of data transformation scripts. In *Human Factors in Computing Systems (CHI)*, 2011.

[18] Youn Ah Kang et al. Evaluating visual analytics systems for investigative analysis: Deriving design principles from a case study. In *Visual Analytics Science & Technology (VAST)*, 2009.

[19] Youn Ah Kang et al. Characterizing the intelligence analysis process: Informing visual analytics design through a longitudinal field study. In *Visual Analytics Science & Technology (VAST)*, 2011.

[20] H Lam, E Bertini, P Isenberg, C Plaisant, and S Carpendale. Empirical Studies in Information Visualization: Seven Scenarios. In *Transactions on Visualization and Computer Graphics (TVCG)*, pages 1520–1536, 2012.

[21] J. Mackinlay et al. Show me: Automatic presentation for visual analysis. *Visualization and Computer Graphics*, 2007.

[22] Jock Mackinlay. Automating the design of graphical presentations of relational information. *Graphics (TOG)*, 1986.

[23] Pattie Maes. Agents that reduce work and information overload. *Communications of the ACM*, 1994.

[24] Joanna McGrenere, Ronald M Baecker, and Kellogg S Booth. An evaluation of a multiple interface design solution for bloated software. In *Conference on Human Factors in Computing Systems (CHI)*, 2002.

[25] Sean Kand others. Enterprise data analysis and visualization: An interview study. In *Visual Analytics Science & Technology (VAST)*, 2012.

[26] Adam Perer and Ben Shneiderman. Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis. In *Conference on Human Factors in Computing Systems (CHI)*, 2008.

[27] Adam Perer and Ben Shneiderman. Systematic yet flexible discovery: guiding domain experts through exploratory data analysis. In *Conference on Intelligent User Interfaces (IUI)*, 2008.

[28] Swapna Reddy, Ya'akov Gal, and Stuart Shieber. Recognition of users' activities using constraint satisfaction. In *Conference on User Modeling, Adaptation, and Personalization (UMAP)*, 2009.

[29] Wolfram Research. Wolfram predictive interface. `http://reference.wolfram.com/mathematica/guide/WolframPredictiveInterface.html`.

[30] Michael Sedlmair et al. Evaluating information visualization in large companies: challenges, experiences and recommendations. In *3rd BELIV Workshop (BELIV)*, 2010.

[31] Ben Shneiderman and Catherine Plaisant. Strategies for evaluating information visualization tools. In *Beyond Time And Errors: Novel Evaluation Methods For Visualization Workshop (BELIV)*, 2006.

[32] Jaideep Srivastava et al. Web usage mining: discovery and applications of usage patterns from web data. *SIGKDD Explorations Newsletter*, 2000.

[33] Robert St. Amant et al. Intelligent support for exploratory data analysis. *Journal of Computational and Graphical Statistics*, 1998.

Explorable Visual Analytics

Knowledge Discovery in Large and High–Dimensional Data

Saman Amirpour Amraii
CREATE Lab, Robotics Institute
Carnegie Mellon University
and
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA
samirpou@cs.cmu.edu

Michael Lewis
School of Information Sciences
and
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA
ml@sis.pitt.edu

Randy Sargent
CREATE Lab, Robotics Institute
Carnegie Mellon University
Pittsburgh, PA
randy.sargent@cs.cmu.edu

Illah Nourbakhsh
CREATE Lab, Robotics Institute
Carnegie Mellon University
Pittsburgh, PA
illah@cs.cmu.edu

ABSTRACT

Visual analytic tools are invaluable in the process of knowledge discovery. They let us explore datasets intuitively using our eyes. Yet their reliance on human cognitive abilities forces them to be highly interactive. The interactive nature of visual analytic systems is facing new challenges with the emergence of big data. Massive data sizes are pushing against the boundaries of current visualization capabilities. Also the emergence of complex datasets is asking for new ways of navigation in the high–dimensional space. EVA (Explorable Visual Analytics) is an in-progress work for developing a web–based tool for visual exploration of large and complex datasets. EVA tries to handle large data sizes through utilizing local GPU resources and a novel client/server architecture. It also provides an easy navigation mechanism for exploring high–dimensional data. This paper presents our experiments in knowledge discovery with EVA, using US Census employment dataset as our testbed. We hope our experiences result in designing guidelines and techniques for the future visual analytic tools of the big data era.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: General; H.4 [**Information Systems Applications**]: General; H.1.2 [**User/Machine Systems**]: Human Information Processing

Keywords

visual analytics, data exploration, visualization, dimension reduction, data mining

1. SENSEMAKING AND BIG DATA

A data explosion is happening, promising invaluable opportunities in scientific and technological progress, yet this vast potential relies not only on our ability to collect and access this data, but also on us being able to understand it. Despite this fact, it seems that knowledge discovery from raw data has not still reached its full power. We are producing much more data than we can explore leading to massive amounts of untapped data waiting for future discoveries. But what makes knowledge discovery hard?

There are in general two major approaches to do knowledge discovery: either we use mathematical methods (e.g. machine learning) or we use human judgment by directly looking at data (e.g. visual analytics). Mathematical methods are profoundly powerful tools yet they still rely on human intuition for the following reasons. First, mathematical methods are a collection of tools. Finding the right tool, using the right models, tuning its parameters and feeding the right feature space into it are often done by human experts. Second, mathematical methods are not context-aware. It is this extra knowledge that usually leads human experts to find the right features or ask the right questions. Third, mathematical methods are not good at providing explanations. A famous example is a Neural Network which is great at finding patterns but does not provide any explanation for how does it find it. And last but not least, mathematical methods are best practiced by mathematicians and computer scientists while most data experts are from other fields, not proficient enough in using these tools on their own data. These facts force us to keep the human in the knowledge discovery loop. Therefore the important question to answer becomes how do people make sense of the data?

Jerome Bruner [10] argued that children posses three modes of representation, (1) interactive, (2) visual and (3) symbolic, and they use these modes to understand a new object or system. In other words we act, we see, and we ask to make sense of something new. For example, upon encountering a new object, the child uses her hands to play with the object, looks at it to find out what happens when she touches it and in the more abstract level she may even ask a question to acquire new sources of knowledge. This process is then repeated until the child amasses enough knowledge about the object until she can build a reliable mental model representing it (Figure 1). It can be argued that even scientists upon facing a new system, be it a simple object or a complex dataset, go through the same process in order to build a mental model of it. This multi-modal exploration of data is an essential step in building the right intuition and plays a significant role in choosing and applying the right rigorous methods in the following steps. For example in a classification problem using machine learning tools, data scientists usually first draw the raw data and do some basic interactions with the data (e.g. scaling). This step provides the initial guidance which then translates into choosing the right model/machine learning tool. This process of building a mental model of the data is called sensemaking. It is only after acquiring this intuition that we can apply our mathematical tools in their full power and extract meaning and knowledge out of the raw data. It is worth mentioning that the model presented in Figure 1 has a hidden assumption: the feedback we see from interacting with an object should be almost instantaneous. If we devise a new theory and test it on the object/data but receive our answer after several hours, we will not be able to effectively build a mental model as we lose our train of thought after only a few seconds. Therefore query latency can have a major impact on the sensemaking process.

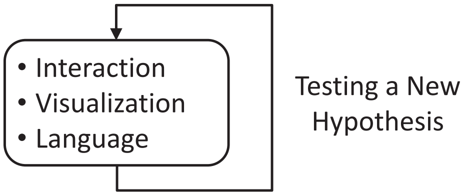


Figure 1: Multi-Modal Exploration: how people understand an object or a system.

Up until now, this sensemaking process has been done intuitively, usually through conventional visualization techniques (e.g. plotting). But the emergence of vast and high dimensional datasets is raising challenging issues not addressable by our current data analytic approaches. For example, current datasets are getting so large that asking even the simplest questions from them may take hours or days of computation. Even after accessing the data, usual visualization techniques may not work due to issues like overplotting. Furthermore, it is not even possible to fully visualize datasets that have hundreds or thousands of dimensions. Another issue is the lack of hypotheses for analyzing the data. Due to decreasing trend of storage prices, we are acquiring and storing an ever increasing amount of data without knowing which portions of that might be useful in a future analysis. Facing with these datasets, even finding the right questions becomes a part of data exploration process.

EVA (Explorable Visual Analytics) is an effort to seek for design guidelines and analytic tools which are capable of visualizing, exploring and analyzing large and complex datasets. Our hope is to promote a set of practices which lead to faster and easier data driven knowledge discovery. To achieve this goal, EVA attempts to facilitate hypothesis generation and query refinement through a series of consecutive multi-modal exploration loops. We also seek new computational techniques which can scale appropriately with the data size and complexity.

Section 2 gives some examples of how researchers are approaching large and complex datasets and what are the challenges they are faced with. In Section 3 we introduce EVA and give an example of using EVA for knowledge discovery on real data. Section 4 discusses some of the lessons we have learned so far in exploratory data analysis and suggests some possible approaches that might expand our ability to do knowledge discovery in large and high–dimensional data.

2. NEW APPROACHES IN VISUAL ANALYTICS

2.1 Knowledge Discovery, Visualization, and Big Data

The process of knowledge discovery is a fundamental aspect of science in general. A rich model for describing this process is presented in [22]. The authors argue that scientists navigate in a four dimensional space in order to extract meaning from their observations. The first dimension in this paradigm is called *data representation*. This is where an abstract representation of data is being formed from a set of features. The second dimension is *hypothesis space*. Here, the scientist generates new assumptions on the possible causal relationships. Then she moves to the third dimension of *experimental space* in order to test those hypotheses. It should be noted that the experiments themselves live in an experimental framework that defines the boundaries of valid experiments and expectable outcomes. Therefore the fourth dimension is *experimental paradigm space* where the scientist can choose a completely different class of experiments for her task. In visual analytics tools, a knowledge discovery process can be modeled based on the first three dimensions. A specific visualization is an example of the data representation space. The ability to interact with the data is happening in the experimental space. Finally, the visualization/exploration choices form a series of decisions in the hypothesis space. By using visual space for doing data representation, we have a tangible and more direct connection to the actual data. By forming a visual query, we actually form a hypothesis in our mind and when we do a visual search, we are experimenting with data in order to confirm or reject our hypothesis. This process has been explained in literature in various ways. Fry [13] presents this process in seven steps. First, we should acquire the data. Then we have to parse it and make it machine readable. This is then followed by filtering in which we select a subset of data that is relevant to our work. We then mine for useful information which usually means some sort of mathematical transformation. The results are then represented in an initial visualization. Then comes the refinement and finally interaction steps in which we explore the visualization and improve it by redoing the previous steps until we extract or discover the desired knowledge. A more general perspective on knowledge discovery is pursued in the field of visual analytics [17, 18]. The goal of visual analytics is to illuminate the way people understand data and then turn it into an algorithmic discipline which benefits from both the power of automated processing techniques and the capabilities of humans in discerning and analyzing visual patterns.

Visualization research has been successful in turning raw data into meaningful visual presentations yet the general perspective of the field does not differentiate between small and simple datasets with large and complex ones. This paradigm is changing as visualization experts face with unforeseen challenges unique to the big data era. For example, as the size of the dataset grows, the responsiveness of traditional visualization systems drops until it is no longer interactive. In addition to the scalability issue, visualizing and understanding complex datasets with hundreds of features is very challenging. These issues have opened new lines of research which often try to change the underlying visualization approach in order to overcome these limitations.

Fekete [11] provides a nice summary of the challenges faced by current visual analytics tools and the paradigm shifts required to overcome these issues. He argues that as data sizes are getting larger, query latency is posing a serious problem. If the system does not provide an answer to a particular question within a few seconds, the analyst may forget her question and not benefit from the answer. He suggests that by shifting from conventional *accurate but slow* analytic tools toward *inaccurate but fast* paradigms, we can overcome the query latency issue when we are dealing with large and complex datasets. It is interesting to see this mindset has started to gain momentum for example in [12] where authors use partial but fast querying techniques to analyze very large databases. Fekete argues that another issue in current analytic systems is the lack of feedback and steering. When a user sends a query, the system starts producing a report. This process cannot be interrupted by the user. She should wait until a query–response "episode" is finished and then start asking a new question. We need to be able to steer the system toward our desired answer as it is analyzing the data. For example, we should be able to play with the parameters of our question or navigate through the data space and ask for finer and more accurate answers for a subspace of a large dataset. Interactivity is another important aspect of visualization systems. For example when a user tries to rotate a 3D object, the operation should happen instantaneously, usually within a 100 ms. This poses a great technical challenge in front of current visualization tools which their frame rate usually drops considerably fast even with modest data sizes of tens of thousands [8]. To summarize these issues we should expect new visual analytic tools which provide responsive multi-modal exploration mechanisms to support sensemaking, provide novel steering abilities to navigate large and high dimensional data, focus on small query latency even in cost of inaccurate answers, and provide non–episodic interactions where a user can modify her query while it is being processed.

2.2 Dealing with Size: Screen–Aware Tools

While data sizes are growing without any foreseeable limit, our cognitive abilities are fairly limited. We probably can only perceive a few million features or even less [11]. As it is us who are the actual bottlenecks in understanding visualizations of large data, a new class of solutions are emerging which focus on the output instead of input. These screen–aware (or output–sensitive [7]) tools use various data abstractions to reduce the size of presented information and avoid analyzing portions of data that are out of the scope of screen. They then use interactive and exploratory mechanisms to help the user navigate through the visualization and understand the data better. These tools are based on the assumption that we do not care for fine details in a big data visualization. A data analyst who looks at a visualization of millions of points is often only interested in the general shape of the visualization; the exact location of a single pixel is usually not important to her. On the other hand, she would prefer to be to able to interact (e.g. zoom, pan) with this visualization in a fluid manner in order to form a better mental model of the overall characteristics of the visualization.

One class of screen–aware solutions are called on-demand processing [7]. They only draw those things that would be visible. For example if the visual representation of a data point is smaller than a pixel or outside of the scope of screen, there is no need to process it. One of the most common techniques used in this class is semantic zoom [15]. In contrast to geometric zoom which redraws all pixels upon zooming, the semantic zoom provides more detail when zooming in and hides some of the detail when zooming out. This can result in tremendous conservation in processing and communication load and therefore it has been used extensively in visualization systems, such as online maps, etc. Semantic zooming is usually used in conjunction with multi–resolution data structures. The basic idea of a multi–resolution data structure is to pre–compute the visible data for each zoom level. As an example, this technique has been used in GigaPan and TimeMachine [21] to present massive high resolution images and videos in an interactive setup which allows zooming on any desirable part of the video while keeping the communication and processing under a manageable limit. Another example of multi-resolution data structures is presented in [19]. Here, the data structure is more complicated and has many dimensions but the fundamental idea is to aggregate over different features and pre–compute these values for several desirable zoom levels. This can then be used to interactively visualize multi–dimensional datasets with over billions of data points.

Another class of data abstraction solutions go further than only showing visible things and instead focus on only showing the important things. In one popular set of techniques, it is the computer/algorithm itself that decides on what is considered important. These techniques are usually pursued as clustering, sampling, aggregation, filtering, . . . where the algorithm either combines several data points or selects a smaller subset of them and only processes those smaller representations. An excellent example in this class is presented in [12]. Here, when the user sends a query to visualize some aspects of the data, the algorithm will randomly select a small sample of data points and then visualizes only those points. It also presents some confidence intervals around each visualized object in order to help the analyst in understanding the error range of the incomplete visualization. With more time, the system grabs more data points and increases the accuracy of its visualization (also decreasing the confidence intervals). This system provides a very promising approach to visualization of large datasets by using both aggregation (in the form of queries) and sampling while in the same time it provides an inaccurate but responsive experience.

While using computer algorithms in choosing the important aspects of data results in highly scalable visualization systems, it is not obvious whether the algorithms will always choose the correct abstraction. This is why another class of solutions insert the user in the loop and ask her to provide feedback on what is important and what should be visualized. The most common type of these techniques is query–based visualization [7]. Here, the user creates a query or search term and reduces the amount of data to a smaller subset which is then used for the final visualization. For example, Beyer et al. [6] present a query–based system for visualizing neurons in a terabyte scale dataset. The user selects regions and neurons of interest and then the system presents neighboring neurons and their relationship in an interactive setting. Another technique used for finding the correct abstraction is steering. Here, the user guides the visualization system in a two–way mechanism — the system provides an initial visualization and then the user refines it by steering the system toward her regions of interest and then the process repeats. An excellent example in this area is presented in [24] where the system uses a dimension reduction algorithm to present a large and high–dimensional dataset but instead of keeping users as passive observers, it actively engages them: the system gradually shows more points in the projected visual space while the user can steer the system towards her desirable regions. This allows the system to only focus on projecting data points in that region, therefore avoiding unnecessary calculations.

2.3 Dealing with Complexity: Human–Assisted Navigation

High–dimensional datasets are inherently hard to visualize (think of a 4-D cube) yet current big data trend is not only expanding in data size, but also in data complexity. Most high–dimensional visualization systems focus on some sort of dimension reduction. One class of these techniques are human–assisted methods which benefit from human feedback in their dimension reduction process [23]. These methods are often heavily interactive as it seams interacting with a visualization can somehow compensate for our inability to perceive high–dimensional space. Human–assisted dimension reduction usually starts with a projection algorithm that has some parametric values. The role of the human is to fiddle with these parameters until the final projection is more suited to her needs. This approach adds an extra layer of sophistication to the visualization system and extends its capabilities in generating meaningful projections of the complex data. It also has the added benefit of engaging the operator in the visualization process. This can both increase the awareness of the analyst plus through her feedbacks, the system can save valuable computational resources. One of the early examples of human–assisted methods in visualizing high–dimensional datasets is Grand Tour [23]. In a Grand Tour, the analyst can choose any arbitrary nonorthogonal projection of the data. This can reveal features that may remain hidden in the conventional orthogonal projections used in some other approaches such as parallel coordinate plots. Another early example of human–assisted methods is presented in [20]. This system has been used to visualize documents in a multi–dimensional setting. Each dimension is represented as a point in the visualization plane and documents would attract/repel to these points based on their similarity to each dimension. Also, by moving these feature points, the user can see how each document reacts. This helps in clustering documents into similar groups in their complex environment.

Steering is one of the recent techniques in human–assisted approaches. Williams and Munzner [24] introduce a navigation mechanism in which the operator steers the system toward the desired subspace of the original dataset. The projection algorithm is then focused on this area, avoiding unnecessary computations on the rest of the dataset. Also, by actively engaging the user in the process of complexity reduction, the operator builds a better mental model of the data. Ingram et al. [16] provide a different mechanism for engaging the user. Here, the system provides a collection of different dimension reduction algorithms and provides tools for tuning their parameters. The analyst can combine these algorithms together until she finds a desirable low–dimensional representation of the data. This is especially beneficial when the user is not an expert in machine learning and dimension reduction techniques. The authors also extensively use the idea of navigation and landmarks. Different levels of global and local navigation improve the exploration ability of the visualization tool while landmarks help the user to find interesting projections of the data. In a similar fashion, Gratzl et al. [14] introduce a tool for exploring rank–based data. Here, the projection algorithm is a simple weighted linear combination of dimensions, but the user has much more power on selecting each weight and the overall combination rules. The tool is also highly interactive, making it easy to create new hypotheses and then testing them through a simple drag and drop process.

2.4 Next Steps in Visual Analytics

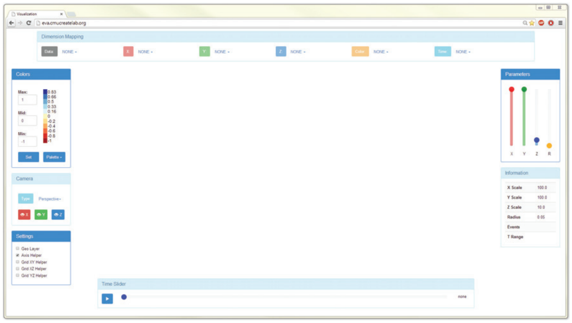
The solutions discussed here are reshaping the conventional visualization paradigm. They put priority over speed and responsiveness even if it results in reduced accuracy, presenting a subset of data or presenting an abstract and compressed version of it. These solutions are also often screen–aware, which means their computational complexity is usually dependent on the screen size rather that data size. This makes them great candidates for emerging visual analytic tools that are capable of scaling with growing data sizes. Future visual analytic systems should also offer non–episodic interaction with the data. In this type of interaction, the user can constantly fiddle with the parameters of the query while the system instantly demonstrates new visualizations. This means that when the system receives a new input from the user, it would not wait until it completes the previous data analysis action. Instead, it adjusts its results to the new query. This interactive query building is essential in forming and improving our hypotheses about the data and as Fisher et al. [12] show in a case study, this can be highly beneficial for data analysts. Non–episodic interaction can be useful because in knowledge discovery we often need to ask many questions and perform multiple iterations on our hypotheses before we can form the right questions. A data analyst seldom asks only one question. She should form many assumptions and refine those assumptions through consecutive visualizations until she can find the answers she is looking for. The ability to change query parameters on the fly should be accompanied by fast response times from the system. Our memory is very limited, specially when dealing with vast quantities of visual information. Short query latency and intuitive navigation mechanism can help us go back and forth between several visualizations and look at the data from multiple perspectives, therefore increasing our chance for finding meaningful patterns.

3. EVA PROTOTYPE

Explorable Visual Analytics (EVA [5]) is a visualization system prototype. It has been developed to address the challenges arising in dealing with large and complex datasets. The main philosophy behind designing EVA is to improve hypothesis generation, both in quality and quantity. EVA tries to provide easy to use and intuitive navigation mechanisms. Through them, the user can easily navigate in a large space of data objects. It also helps the analyst to look at the multi–dimensional data from multiple perspectives, hence giving her a better chance for finding interesting phenomena in the data. In general, the interactive nature of EVA is critical in sense making and creating a mental model of the data. Also, EVA is designed to be responsive as it is beneficial to minimize the time between generating a question and testing it. There is an important period between when an analyst forms a question in her mind until she can see the relevant visualization to test that hypothesis. If it takes too long (e.g. even more than 10 seconds), the analyst may lose her train of thought. This is mainly due to our limited working memory. EVA minimizes this delay period and therefore lets the analyst to instantaneously test her new ideas. This is in turn helpful in generating more questions. In conclusion, EVA provides a simple navigation mechanism for studying a large and complex dataset through visual inquiries. It also has short processing time in order to avoid any delay between receiving a query from the user and visualizing it. EVA is also designed to provide a high resolution visualization, as richness of details is an important factor in doing knowledge discovery. All of these aspects helps the user to start with a relatively small set of assumptions, test them, generate new questions, refine them, and gradually build a better model of the data, which then results in finding new and meaningful patterns.

Based on the knowledge discovery framework presented in Section 2.1, EVA is composed of three major conceptual sections. In the *data representation* section, EVA provides a 5 dimensional visual space consisting of spatial coordinates (X, Y, Z), color and visibility period (named as Time). Each data point can be assigned to an instance of this visual space. In the *hypothesis space*, the user can use a simple one-to-one mapping function from data space to visual space. It is also possible to scale data values to better fit them in the visual space. In the *experimental space*, EVA provides various tools for interacting with and manipulating the visualization in order to do a visual search and find interesting patterns. These mechanisms include tools and techniques such as zoom, pan, rotation, choosing color palette, scaling, camera features, external visual aids such as Google Maps and also some textual helpers such as an information panel.

EVA is a web–based tool developed at CMU's CREATE Lab [1]. It is a part of Explorables [2] collaborative which consists of various projects aiming at interactive visual representations of large datasets. EVA is accessible from http://eva.cmucreatelab.org. It is written in JavaScript and HTML. It uses a selection of color palettes presented in Color Brewer [9]. It also uses the WebGL–based Three.js [4] library for its graphical engine. Choosing web–based technologies has been helpful in sharing EVA with other experts and incorporating their suggestions during development phase. EVA fully utilizes the GPU and RAM in order to visualize large datasets without sacrificing its response time. Currently, it can handle data sizes of up to a few million points consisting of tens of dimensions. Figure 2 shows a screenshot of EVA in a browser.



Max: 1
Mid: 0
Min: -1

☐ Geo Layer
☑ Axis Helper
☐ Grid XY Helper
☐ Grid XZ Helper
☐ Grid YZ Helper

X Scale	100.0
Y Scale	100.0
Z Scale	10.0
Radius	0.05
Events	
T Range	

Figure 2: EVA's main screen.

Choosing the right dataset for EVA has been based on several factors. First, we wanted a dataset large enough to be beyond the processing capacity of usual visualization tools, yet not too large to complicate the development of our first prototype. As current tools are usually limited to visualizing a few tens of thousands of objects, we chose a limit around a few millions of points for our dataset. The second factor in choosing a dataset is its complexity. A dataset with a few dimensions (say 4) can be visualized completely using spatial dimensions and color. On the other hand, manually selecting and navigating through hundreds or more dimensions is tedious and very complicated. Therefore, we limited the datasets dimension cardinality to tens of dimensions. It is also important to chose a meaningful dataset acquired from real world measurements. This can lead to relevant and useful knowledge discovery. Also, the analyst can benefit from her expertise in the contextual information accompanying that dataset. Finally, the data should have some meaningful representation in the spatial space, otherwise a purely visual exploration may not be as beneficial.

Based on these characteristics, we chose United States Census Longitudinal Employer–Household Dynamics (LEHD [3]) dataset. This dataset provides information on employers and employees across country. This information includes categories such as employees earning, age, ethnicity, education level, etc[1]. It is aggregated over census blocks which are small geographical regions usually equivalent to a city block. Also, the data is produced yearly, therefore providing enough details both on the spatial and temporal levels. This dataset is being used by a wide span of scientists and analysts from economists to urban researchers. As such, it can be used with a rich set of contextual knowledge from various fields and therefore it can be a good candidate for doing meaningful knowledge discoveries. Currently, the visualization tools dedicated to LEHD are limited and they often work on aggregations of the original data, hence they do not visualize it with fine details. The LEHD dataset in its entirety is very large (more than a 100GB). Therefore we have limited our work to the state of Pennsylvania[2]. This

[1] Details of LEHD data structure is available at http://lehd.ces.census.gov/data/lodes/LODES7/LODESTechDoc7.0.pdf

[2] Particularly to the residence–based workforce information subsection of LEHD. For Pennsylvania, this dataset is around 300MB.

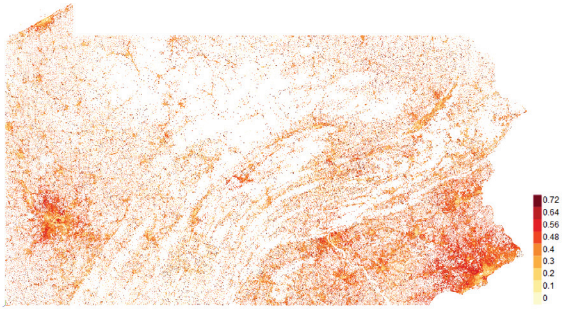


Figure 3: Earnings more than $3333 per month for Pennsylvania. Each dot represents the center point of the corresponding census block. Red areas show regions with a higher percentage of residents in high–end income range. The color palette on the right shows the minimum percentage of employees with the aforementioned income level in each census block.

subsection of LEHD has around 2.8M data entries and 44 dimensions. Next, we will go through some examples of using EVA on LEHD for understanding the data better and then doing discoveries as we interact with the data.

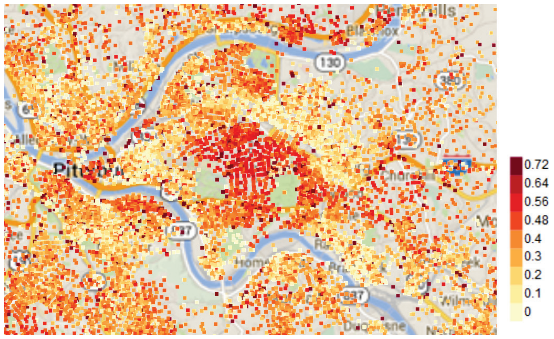


Figure 4: Earnings more than $3333 per month for Pittsburgh.

The first example is a simple visualization of income (Figure 3). Each dot represents one instance of data. The longitude dimension of each data instance is assigned to the visual dimension X (the horizontal orientation of the figure). The latitude dimension is assigned to the visual dimension Y (the vertical orientation of the figure)[3]. The visual dimension of color represents the ratio between the number of jobs with an income of $3333 or more per month with the total number of jobs. Therefore a pixel with bright red color shows a relatively wealthy neighborhood while a pixel with yellow color shows a poorer area. In general, there are 2.8 million pixels in the visualization. From this visual representation, it is easy to locate the major population poles of the state, such as Philadelphia on the bottom right corner or Pittsburgh on the left side. It is also possible to distinguish the major geological features of the area such as the distinctive Appalachian Mountains in the middle of the map. The other important observation is the non–uniform distribution of wealth throughout the state. Most of high–end income earning neighborhoods are concentrated in the suburbs of Philadelphia and Pittsburgh while the regions in the middle are usually less populated and often have a lower amount of income. Figure 4 shows a zoomed in version of Figure 3, focusing on Pittsburgh. This picture also includes a Google Map helper in the background. This layer can be helpful in distinguishing the exact location of each census block. Based on this map, the main wealthy neighborhoods are seen in the middle of the picture, where the University of Pittsburgh and Carnegie Mellon University are located.

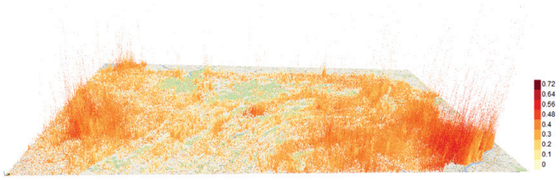
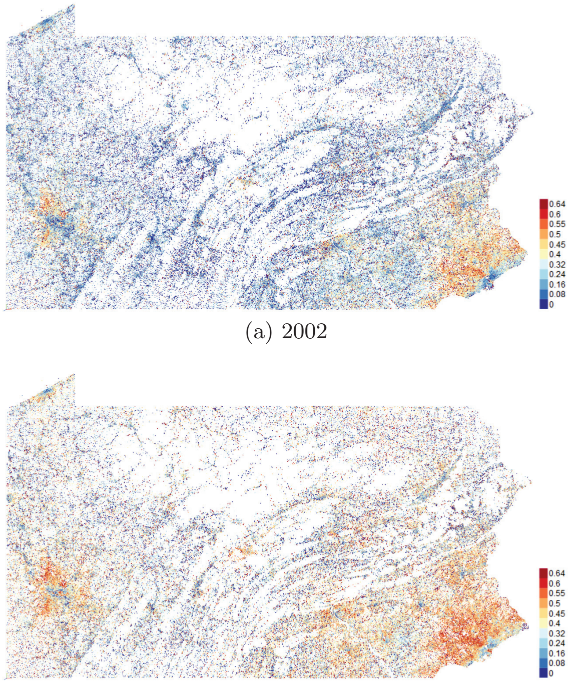


Figure 5: Earnings more than $3333 per month (as color) combined with total number of jobs (as elevation).

In Figure 5, we have utilized all the 3 spatial dimensions. Here, besides assigning longitude and latitude to X and Y, we have assigned total number of jobs in each location to dimension Z. By rotating the visualization, the user can look at the high–income levels (as color) and total number of jobs (as elevation) at the same time. Through this representation, it is again easy to find the major population hubs. Also, it is more evident that there is a more complex relationship between income level and number of jobs. For example by looking at Philadelphia at the bottom right corner, we can see areas of high income (red) and low income (yellow) with almost the same number of jobs adjacent to each other. Another interesting example is State College, home of Pennsylvania State University, located at the center of the map. This small city has a relatively low number of jobs, but the color of those jobs shows a high–income region, representative of its higher education employment sector. It should be noted that most of the visual objects in a point cloud are obscuring each other, therefore it is essential to have interactive capabilities. Through rotation, zooming and panning, the user has a much better chance of understanding the general outline of the visual space.

The last visual dimension available in EVA is Time. By assigning a data dimension to time, we can create an animation and control it through the bottom slider. Figure 6 shows the high–end income range percentages over a course of 10 years. As it is evident from comparing Figure 6(a) to Figure 6(b), the percentage of people with higher incomes is increasing over the decade. This can be due to the inflation in income or a real increase in the overall earnings. The time slider plays an important role in revealing this pattern as the user should go back and forth in time multiple times to better perceive the gradual change in earnings. Again, the interactive nature of visualization is vital in the knowledge discovery step. The same data is represented in a different view in Figure 7. Here, instead of the usual assignment of years to Time dimension, we have assigned it to Z. This results in a series of planes dissecting the data accord-

[3] The latitude and longitude measures represent the central location of the corresponding census block.



(a) 2002

(b) 2011

Figure 6: Earnings more than $3333 per month in years 2002 and 2011.

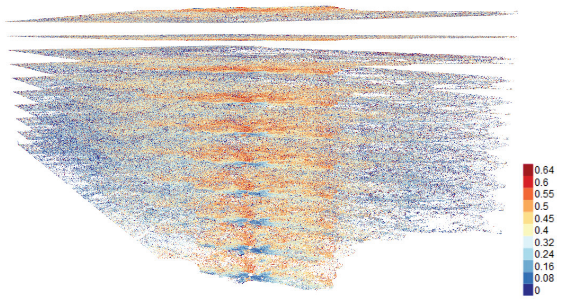
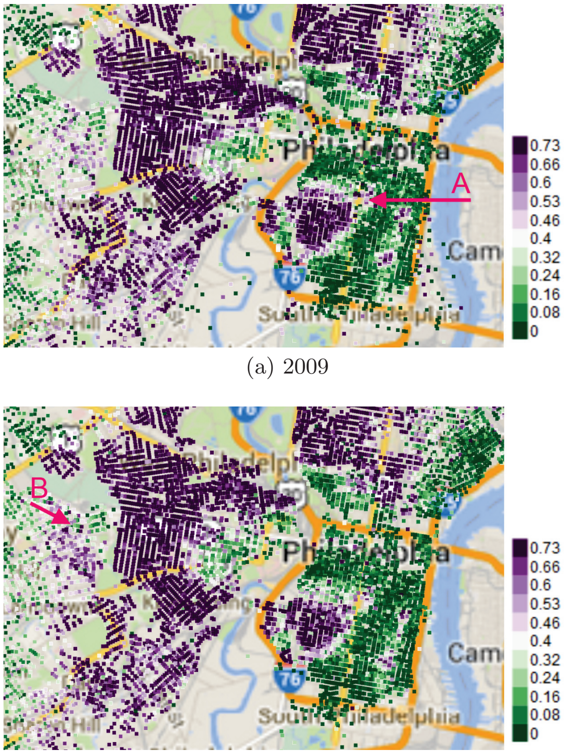


Figure 7: Earnings more than $3333 per month. The year dimension from the data is assigned to the Z dimension in the visual space.

ing to their year. This is useful for looking at the general trend. For example, the region in the front of the picture in Figure 7 is Philadelphia. We can see the lower layer (corresponding to year 2002) has more blue dots (corresponding to poor neighborhoods). As we go up in the layers we are going forward in time and we can see the shrinking of blue regions and the growth of higher–income neighborhoods.

Figure 8 looks at the distribution of races in the city of Philadelphia over the course of three years (from 2009 to 2011). The green regions represent neighborhoods with a majority of Whites while purple regions show neighborhoods with a majority of workforce from African American community. The first observation is the segregation between these two communities. Neighborhoods are mostly dominated by only one race while in between there are some small border neighborhoods that accommodate a more balanced mixture of both races. The other observation is the relatively fast shifts in the population proportions of some border neighborhoods within a course of a few years. For example, the region marked as **A** in Figure 8(a) shows an area that is mostly composed of African Americans in 2009. But as we go forward in time to year 2011 (Figure 8(b)), this area becomes a more mixed race neighborhood. The opposite phenomena is happening in region **B** where it is changing from a mixed community to a more single–race neighborhood. During some informal discussion with a Philadelphia resident, he hypothesized that this population shift may be related to a new wave of African immigrants settling in the west side of the city.



(a) 2009

(b) 2011

Figure 8: Distribution of employees based on their race. Purple areas represent neighborhoods with a majority of African American workforce while the green areas represent neighborhoods with a majority of Whites. (a) shows this distribution in year 2009 and (b) is for year 2011.

The next example shows an accidental discovery. Here, the exploration was not driven by a hypothesis. Instead, it was the exploratory nature of the tool that led to an unexpected visualization. This later resulted in formation of new hypotheses. When working with geolocated data such as LEHD, it is common to visualize the data on a map. Figure 9 shows a visualization of LEHD data in an effort to view it outside of a geo–spatial representation. Here, each dot corresponds to one census block (i.e. neighborhood) on

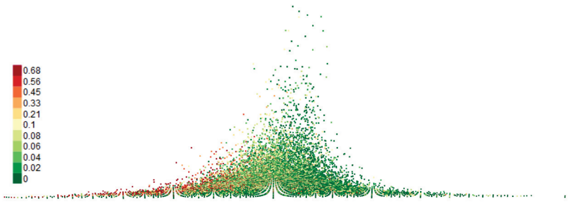


Figure 9: The relationship between race, gender, and total number of jobs. The dots on the right–hand side represent neighborhoods where a majority of workforce are men. The dots on the left–hand side are areas where the majority of working people are women. The elevation shows the relative total number of jobs. The color shows the percentage of African Americans in that neighborhood (red shows higher percentage of African Americans in that census block).

the map. The number of jobs for males has been assigned to the X dimension and the number of jobs for females has been assigned to Y dimension. Furthermore, the total number of jobs in each neighborhood has been assigned to the Z dimension. Viewing the final visualization from a perpendicular angel, we come up with Figure 9 where a dot on the right–hand side represents a neighborhood with a higher percentage of workforce being male, while a dot on the left–hand side shows a region with a higher percentage of females in the workforce. The elevation shows the total number of jobs. As it can be expected, most of the neighborhoods are located in the middle, with an almost 50–50 percent distribution of jobs between men and women. But the unexpected feature of this visualization is the one–sided distribution of red dots. Here, we have assigned number of jobs for African Americans to the Color dimension. Therefore the red dots show neighborhoods with a majority of workforce from African American community. Seeing that most of these dots are on the female side of the graph we can hypothesize that either there is a high unemployment rate among African American men or that they are working in areas with a majority of workforce from other races, hence their presence is not visible. In either case, the exploratory nature of EVA plus the ability of going through many visualizations in a short amount of time was crucial in creating this visualization and therefore forming new hypotheses about the nature of the data. It can be imagined that even randomly going through several different projections of the data can reveal some interesting patterns that are not evident in the first place, due to the lack of initial hypotheses in the mind of the analyst.

4. DISCUSSION

We can summarize EVA's contributions in three aspects: high resolution, explorability, responsiveness. High resolution is the ability of EVA to show as many data points as possible on a screen without aggregating them into overall summaries. The aggregation technique is used in many tools to improve their ability in working with larger datasets, but it also reduces the clarity of final picture and hides the fine details of the data. Knowledge discovery can be very dependable on the amount of detail a user can see. In the explorability aspect, EVA provides usual interactive techniques (e.g. zoom, pan, etc.) plus easy navigation between multiple projections of data through its dimension assignment tool. Our initial experiments showed that the ability of viewing data from multiple perspectives is crucial in understanding the data and finding the "wow" moments where the analyst observes some unexpected pattern. These moments usually lead to deeper investigations, new hypothesis generation, and sometimes to new discoveries. Finally, the responsiveness aspect of EVA fully utilizes its other features. Knowledge discovery is a memory intensive process. The analyst should form a series of assumptions and questions in her mind, and then create a series of visualizations, looking at one characteristic of the data in each step. It is important to remember all of these steps and their possible interpretations. If there is a long waiting period between each two step, the user can easily forget her previous observations and hence the general knowledge discovery process will be interrupted. EVA is designed from the ground up to address this issue by fully utilizing local computing resources available in order to make fast and smooth transitions from one visualization to the other. This is a fundamental feature in data exploration, specially when data size and complexity grows.

It is worth noting that EVA should be used in conjunction with a statistical tool. The main purpose of EVA is to facilitate hypothesis generation. It will also show visual representations of the data so the analyst can perform an initial test for each hypothesis, but coming up with a final accurate and reliable answer is the job of a statistical tool. Another important note about EVA is the role of experts in shaping it. From its inception, EVA has benefited from many experts. The choice of data, its visual characteristics (such as color palettes used), . . . has been formed through many joint sessions with analysts from various backgrounds. Their realtime feedback while working with their own data on EVA has also been tremendously helpful in recognizing EVA's capabilities as well as its limitations. This collaboration would remain an ongoing part of EVA during the future expansions.

We are going to expand EVA in two major aspects: scaling and navigation in the action space. Currently, EVA downloads the full dataset into the local memory. In this way it can fully utilize clients local resources such as GPU and RAM. But this approach is limited to moderate data sizes of a few million points. Larger datasets take a long amount of time to download and they often cannot be fitted to local memory. Therefore, in the future EVA should support a client/server architecture which actively limits data transmission based on the screen resolution and user needs. This screen–aware method would not be accurate and complete, but can be scaled for large datasets. Another addition to EVA is a history function. When users explore a dataset they generate many different visualizations and sometimes they need to compare several views together in order to form a better mental model. A history function can help them navigate in their action space. This can also augment users working memory and improve the quality of their knowledge discovery.

5. ACKNOWLEDGMENTS

This project has been funded by Google.

6. REFERENCES

[1] CREATE Lab. http://www.cmucreatelab.org/, 2014.

[2] CREATE's Explorables. http://explorables.cmucreatelab.org/, 2014.

[3] Longitudinal Employer-Household Dynamics. http://lehd.ces.census.gov/, 2014.

[4] three.js, JavaScript 3D Library. http://threejs.org/, 2014.

[5] S. Amirpour Amraii. Explorable Visual Analytics. http://eva.cmucreatelab.org/, 2014.

[6] J. Beyer, A. Al-Awami, N. Kasthuri, J. W. Lichtman, H. Pfister, and M. Hadwiger. ConnectomeExplorer: query-guided visual analysis of large volumetric neuroscience data. *IEEE transactions on visualization and computer graphics*, 19(12):2868–2877, Dec. 2013.

[7] J. Beyer, M. Hadwiger, and H. Pfister. A survey of GPU-based large-scale volume visualization. In *Eurographics Conference on Visualization (EuroVis)*, page to appear, Swansea, UK, 2014.

[8] M. Bostock, V. Ogievetsky, and J. Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, Dec. 2011.

[9] C. Brewer and M. Harrower. Color brewer 2.0. http://www.colorbrewer2.org, 2012.

[10] J. S. Bruner. The course of cognitive growth. *American Psychologist*, 19(1):1–15, 1964.

[11] J.-D. Fekete. Visual analytics infrastructures: From data management to exploration. *Computer*, 46(7):22–29, July 2013.

[12] D. Fisher, I. Popov, S. Drucker, and m. schraefel. Trust me, i'm partially right: Incremental visualization lets analysts explore large datasets faster. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 1673–1682, New York, NY, USA, 2012. ACM.

[13] B. J. Fry. *Computational information design*. Thesis, Massachusetts Institute of Technology, 2004. Thesis (Ph. D.)–Massachusetts Institute of Technology, School of Architecture and Planning, Program in Media Arts and Sciences, 2004.

[14] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. LineUp: Visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2277–2286, Dec. 2013.

[15] I. Herman, G. Melancon, and M. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, Jan. 2000.

[16] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller. DimStiller: Workflows for dimensional analysis and reduction. In *2010 IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 3–10, Oct. 2010.

[17] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. *Visual analytics: Definition, process, and challenges*. Springer, 2008.

[18] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann. *Mastering The Information Age-Solving Problems with Visual Analytics*. Florian Mansmann, 2010.

[19] Z. Liu, B. Jiang, and J. Heer. imMens: Real-time visual querying of big data. *Computer Graphics Forum*, 32(3pt4):421–430, June 2013.

[20] K. A. Olsen, R. R. Korfhage, K. M. Sochats, M. B. Spring, and J. G. Williams. Visualization of a document collection: The vibe system. *Information Processing & Management*, 29(1):69–81, Jan. 1993.

[21] R. Sargent, C. Bartley, P. Dille, J. Keller, I. Nourbakhsh, and R. LeGrand. Timelapse GigaPan: Capturing, sharing, and exploring timelapse gigapixel imagery. *Fine International Conference on Gigapixel Imaging for Science*, Nov. 2010.

[22] C. D. Schunn and D. Klahr. A 4-space model of scientific discovery. In *Proceedings of the seventeenth annual conference of the Cognitive Science Society*, page 106–111, 1995.

[23] M. Theus. High-dimensional data visualization. In *Handbook of Data Visualization*, Springer Handbooks Comp.Statistics, pages 151–178. Springer Berlin Heidelberg, Jan. 2008.

[24] M. Williams and T. Munzner. Steerable, progressive multidimensional scaling. In *IEEE Symposium on Information Visualization, 2004. INFOVIS 2004*, pages 57–64, 2004.

EigenSense: Saving User Effort with Active Metric Learning

Eli T Brown
Tufts University
Department of Computer Science
161 College Ave.
Medford, MA, USA
ebrown@cs.tufts.edu

Remco Chang
Tufts University
Department of Computer Science
161 College Ave.
Medford, MA, USA
remco@cs.tufts.edu

ABSTRACT

Research in interactive machine learning has shown the effectiveness of live, human interaction with machine learning algorithms in many applications. Metric learning is a common type of algorithm employed in this context, using feedback from users to learn a distance metric over the data that encapsulates their own understanding. Less progress has been made on helping users decide which data to examine for potential feedback. Systems may make suggestions for grouping items, or may propose constraints to the user, generally by focusing on fixing areas of uncertainty in the model. For this work-in-progress, we propose an active learning approach, aimed at an interactive machine learning context, that tries to minimize user effort by directly estimating the impact on the model of potential inputs, and querying users accordingly.

With EigenSense, we use eigenvector sensitivity in the pairwise distance matrix induced by a distance metric over the data to estimate how much a given user input might affect the metric. We evaluate the technique by comparing the output points it proposes for user consideration against what an oracle would like to choose as inputs.

Categories and Subject Descriptors

I.5.5 [**Pattern Recognition**]: Implementation—*Interactive Systems*; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Human Factors

Keywords

Active Learning, Metric Learning, Interactive Machine Learning

1. INTRODUCTION

The field of interactive machine learning has demonstrated the effectiveness of using human interaction to improve machine learning results, and simultaneously using machine learning algorithms to improve user experiences. Example systems help people group or model their data without having to understand machine learning techniques [14].

One concept in the machine learning apparatus underlying many examples is metric learning. Understanding the similarity between objects is a powerful way to model them for grouping or labeling. In metric learning, a distance function over the data is learned from side information about the data, often in the form of constraints for which objects are similar. For an interactive context, the algorithm must update incrementally by making improvements iteratively with increasing user feedback.

User feedback is expensive, since human throughput at reviewing data is far lower than a computer's throughput at analysis. In order to maximize utility of user efforts, active learning researchers develop techniques to query users for feedback in ways that will help the machine learner. A common approach is to query users about points chosen so that the user feedback will resolve uncertainty in the model.

Emphasizing the interactive learning perspective, our approach keeps the user in control, providing suggestions only when the user initiates a direction of inquiry. We target our active learning method toward predicting the impact of any given user input. Using our method, a user can judiciously spend the effort of developing feedback on data that will affect the underlying model as much as possible.

In this work-in-progress, we first introduce EigenScore, a measure that leverages "eigenvector sensitivity" to predict how much a potential user input will change an underlying metric learning model. We then propose EigenSense, which uses EigenScores to guide a user toward making the most productive feedback while minimizing his or her effort (in terms of data points examined). Finally we provide two types of evidence of the efficacy of this algorithm. First, we compare EigenScores to the ground-truth of what they estimate: the amount that particular constraints would change the metric learning model. Second, we show with simulations that the few points selected for user review by EigenSense often include the best possible choices as evaluated by an oracle.

2. MOTIVATION: EIGENVECTOR SENSITIVITY TO FIND CRITICAL POINTS

The eigenvectors of a matrix have been used to represent its underlying structure for applications in many domains including connectivity graph analysis [33], face recognition [43], and clustering [47]. The eigenvectors of symmetric matrices A for which entry A_{ij} represents some measure of distance between objects i and j is of particular relevance. For example, the PageRank [33] algorithm uses an $n \times n$ pairwise matrix to represent the transition probabilities between pairs of the n webpages. Here entry A_{ij} corresponds to the probability of landing on node (page) j during a length-one random walk, having started at node i. Raising that matrix to the power k gives a matrix of the probabilities of landing on node j having started a *length-k* random walk at node i. Increasing powers of A^k will show the asymptotic behavior of flow through the graph. Conveniently, following from the definition and orthogonality of eigenvectors, the dominant eigenvector approximates this quantity. For a real, symmetric matrix A, suppose we have the eigenvalues $\lambda_1,\ldots,\lambda_n$ sorted in decreasing order, and their corresponding eigenvectors $\mathbf{v_1},\ldots,\mathbf{v_n}$. Because the eigenvectors are orthogonal, any vector $\mathbf{x} \in \mathbb{R}^n$ can be written as a linear combination of the eigenvectors, with coefficients α_i. We can observe the asymptotic behavior:

$$\begin{aligned}
x &= \alpha_1 \mathbf{v_1} + \alpha_2 \mathbf{v_2} + \ldots + \alpha_n \mathbf{v_n} \\
Ax &= \alpha_1 A\mathbf{v_1} + \alpha_2 A\mathbf{v_2} + \ldots + \alpha_n A\mathbf{v_n} \\
A^k x &= \alpha_1 \lambda_1^k \mathbf{v_1} + \alpha_2 \lambda_2^k \mathbf{v_2} + \ldots + \alpha_n \lambda_n^k \mathbf{v_n} \\
&= \alpha_1 \lambda_1^k \left(\mathbf{v_1} + \frac{\alpha_2}{\alpha_1} \left(\frac{\lambda_2}{\lambda_1} \right)^k \mathbf{v_2} + \ldots + \frac{\alpha_n}{\alpha_1} \left(\frac{\lambda_n}{\lambda_1} \right)^k \mathbf{v_n} \right)
\end{aligned}$$

Note that because the dominant eigenvalue, $\lambda_1 \geq \lambda_i, i = 2 \ldots n$ in the final sum, the $\mathbf{v_1}$ term dominates.

When studying population dynamics, biologists take advantage of this fact with a matrix L, called a "Leslie" matrix, where each element L_{ij} represents an organism's survival prospects to age i from age j [1]. To see the equilibrium point of a population, biologists study the dominant eigenvector of this matrix [22].

Extending this technique to see how the population can be affected by environmental factors, biologists adjust survival rates at different times in the lifecycle by editing the matrix, and reconsider the new dominant eigenvalue. This sensitivity of the eigenvalue to change in particular matrix entries is the eigenvalue sensitivity [22].

Motivated by biologists' successes with eigenvalue sensitivity in Leslie matrices, we consider the behavior of the dominant eigenvector of our related $n \times n$ pairwise distance matrices, and we adapt the concept of sensitivity to the context of active metric learning. We will use the dominant eigenvector of a pairwise distance matrix as a standin for its overall structure, and calculate the sensitivy of that eigenvector with respect to changes in entries of that matrix. Since each entry corresponds to a pair of data points, we will use this approach to estimate how individual user inputs, i.e. user constraints that certain pairs of points have small distances between them, will affect the structure of the distance matrix and the underlying distance metric.

3. RELATED WORK

This work builds on previous efforts in both machine learning and human-computer interaction. We begin with an overview of related work in interactive machine learning and then discuss metric and active machine learning.

3.1 Interactive Machine Learning

Interactive machine learning researchers strive to use human interaction with machine learning to improve machine learning results and improve user experiences, leveraging computers' raw analytical power and humans' reasoning skills to achieve results greater than either alone [39]. Several methods coupling machine learning techniques with visualization to cluster or classify data have been proposed [6, 10, 17]. Systems have been built for grading [11], network alarm triage [5], building social network groups [4], ranking search results with user context [2], managing overeating [16], and searching for images [3].

Another vein of this research, from visual analytics, focuses on data analysis tasks, and on effectively leveraging user interaction to refine an underlying model, generally by adjustments of layouts or clusterings [14, 20, 23, 13].

All of this work integrates human reasoning with machine learning without asking the human to understand the machine learning. However, these systems do not offer strong active learning support to help the user give the most efficient feedback. The EigenSense approach aims to provide this support by guiding the user toward the most important data to review.

3.2 Metric Learning

Many of the examples above of interactive machine learning use metric learning algorithms at their core. This powerful approach has been the subject of much research since 2003 [8, 12, 19, 25, 34, 38, 42, 45, 46, 48, 51, 52] and has proven applicable in many real-world application domains including information retrieval, face verification and image recognition [18, 26, 29].

Most of these methods assume that the machine learner is given additional information beyond the data itself, most often as pairwise constraints between data points, i.e. that certain pairs should or should not be close together. With that information, metric learning techniques learn a distance function optimized to produce relatively small distances between points that belong close together, and large distances between those that belong far apart [52].

It is generally assumed that a domain expert can easily provide pairs of similar data points and pairs of dissimilar data points, but that assumption implies a perfectly accurate user who is motivated and available to review all of the data. This work-in-progress begins to address this gap by introducing a technique for paring down what points an expert would actually have to review in an interactive machine learning context by trying to guide a user toward constraints that will be most impactful to the distance metric.

3.3 Active Learning

Active learning is a form of semi-supervised machine learning in which the learner iteratively queries the user for additional information while building its model. The key idea behind active learning is that an algorithm can achieve greater accuracy or performance with fewer training labels if it is allowed to choose the most helpful labels [36].

A common approach is to select the data points that are most uncertain to classify. Different measures of the uncertainty are based on the disagreement in the class labels predicted by an ensemble of classification models [1, 32, 37], by distance to the decision boundary [15, 35, 41], by the uncertainty of an unlabeled example's projection using the Fisher information matrix [31, 53], or with Bayesian analysis that takes into account the model distribution [24, 30, 40, 54].

Active learning and metric learning come together in several recent works, where authors determine what the user should see based on uncertainty of labels and coverage of the dataset [21], or the median points in groups with the same label [44]. Yang and Jin select pairs of points for feedback based on the uncertainty of deciding their closeness [50].

In a sub-category of active learning algorithms called active clustering, the end goal is a clustering instead of classification, and the common approach is to gather constraints by iteratively querying the user about pairs of points. Points are chosen by uncertainty [28], or by most informative example pairs [9]. One work by Xu, et al., is especially related to ours. The authors learn a two-class spectral clustering with active learning by examining the eigenvectors of the pairwise distance matrix to find points on the boundary of being put in either cluster [49].

Generally, active learning methods are based on querying the user for one unit of feedback at a time. In our approach the user plays an active role in deciding what feedback to provide: no sug-

[1] The matrix represents different age groups' rates of survival and reproduction by setting the first row to the fertility rate of each age group, and the lower off-diagonal entries to organism survival probabilities from one age group to the next.

gestions are given without an initial seed point of interest from the user, and then, several suggestions are provided for the user to peruse.

4. APPLICATION CONTEXT

The EigenSense method is best understood within a real interactive learning context. In prior work, Brown et al. created Dis-Function [14], a system that shows an analyst high-dimensional real-valued data in a 2D projection, and learns a distance function iteratively though user feedback. Feedback is provided by dragging together points that should be closer together. The method is effective with appropriate user feedback, but the user is given no information about what points would be helpful to the metric learning backend.

For illustration, we have integrated EigenSense into Dis-Function. When a user clicks a point of interest, EigenSense responds by showing several points that may be of interest relative to the first. Figure 1 presents a screenshot of this modified Dis-Function, specifically the data projection portion. The data have been arranged using multidimensional scaling (MDS), and colored based on a spectral clustering. The user has clicked the point marked by a red X. In response, EigenSense adds colored squares showing which points may be of interest relative to that X. Darker colors indicate a higher eigenvector sensitivity score, or EigenScore (see Section 5.1).

These predictions of which points could provide the strongest update to the model are intended to guide the user towards giving the machine learner fruitful feedback, and taking best advantage of precious expert user time.

5. THE EIGENSENSE METHOD

In our interactive machine learning context, we have presented a user with data and need useful side information to improve our learned model. More specifically, in the context provided by Section 4, we assume a user examines a visualization of data and notices points of interest, perhaps outliers, cluster exemplars, or points aligned with personal expertise. We aim to answer, given one point of interest selected by the user, which are other points that the user should examine. We chose this interaction paradigm for two reasons: first, the user guides the process as opposed to simply being used for point comparisons, and second, having an initial point sharply reduces the computational complexity (see Section 5.2). In deciding which points to suggest for user examination, our ideal is to uncover the point that would make the strongest update to the model with the user's feedback, leveraging expertise efficiently to minimize effort.

In this section, we introduce a technique using eigenvector sensitivity on a pairwise distance matrix to provide these predictions of strong model updates. First we associate a score (called the EigenScore) with any pair of data points. The EigenScore of a pair is designed to predict the strength of a model update corresponding to user feedback about that pair. We then present the EigenSense algorithm, which uses EigenScores to recommend top candidate points to the user.

5.1 Calculating EigenScores

The EigenScore between two points represents our prediction for how strongly a change in distance between them would affect the underlying structure of the pairwise distance matrix. Specifically, it is a measure of the sensitivity of the dominant eigenvector of that matrix to changes in its elements, which correspond to pairs of data points.

Given a distance function and a data set with N points, we calculate the pairwise distance matrix

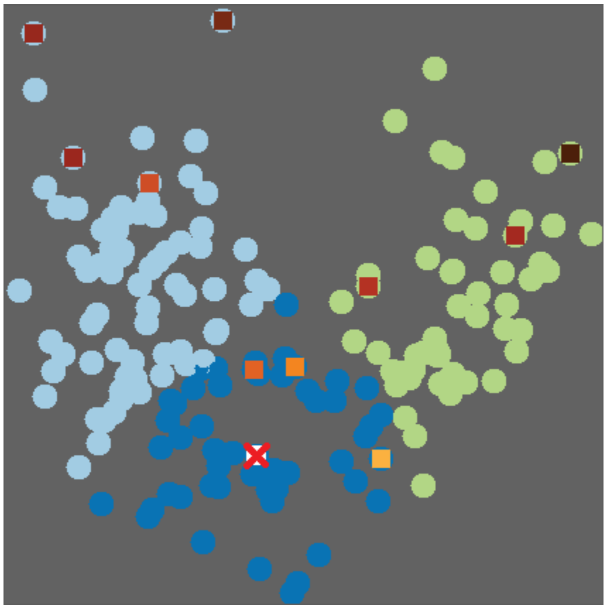


Figure 1: EigenSense demonstrated on an interactive scatterplot of projected data – all data points are laid out with multidimensional scaling (MDS) and colored by a spectral clustering. The point with a red X is the one the user clicked, asking what other data should be considered in relation to that point. The colored squares show the EigenSense response, with darker colors indicating higher EigenScores (see Section 5.1). Only the top five percent of scores from each cluster are highlighted, helping the user target the most fruitful data to examine.

$$D \in \mathbb{R}^{N\times N} \text{ where } D_{ij} = distance(x_i, x_j)$$

Note that no specific type of distance function is required. To model how that matrix changes with specific x_i and x_j assumed to be perfectly close together, i.e. because the user specified so with feedback, we construct a new distance matrix D' which is identical to D, except that we set $D_{ij} = D_{ji} = 0$. These entries now reflect that x_i and x_j should be close to one another.

We next compute the dominant eigenvector for D, called v_1, and for D', which we indicate with v_1'. We compute the cosine similarity between v_1 and v_1'. Note that we desire a dissimilarity metric, showing how much v_1' is different from v_1, so we define

$$EigenScore(x_i, x_j) = 1 - CosineSimilarity(v_1, v_1')$$

Algorithm 1 summarizes this process.

Note that computing the function $eig(D)$ to return the dominant eigenvector is computationally expensive if implemented using factorization methods [27]. Techniques such as SVD [27] and the Cholesky decomposition [27] return all eigenvectors of the matrix D. However, because computing the EigenScore requires only the dominant eigenvector (and because we are restricted to a real-valued symmetric matrix), we can dramatically improve performance by using the Lanczos method [27], which returns only the dominant eigenvector and which we denote $eigs(D,1)$ as in MATLAB.

5.2 Using EigenScores to make EigenSense

Algorithm 1: EigenScore

```
Input: Data points $x_i, x_j$, distance matrix $D$
Output: $ES_{ij} \in [0,1]$
Calculate $v_1 = eigs(D,1)$ [dominant eigenvector]
Let $D' = D$
Set $D'_{ij} = D'_{ji} = 0$
Calculate $v'_1 = eigs(D',1)$
Set $ES_{ij} = 1 - CosineSimilarity(v_1, v'_1)$
return $ES_{ij}$
```

The EigenScore algorithm maps a pair of points to a scalar value representing potential impact on the distance metric, and thus implicitly provides a ranking over pairs of points. This section addresses how to use this ranking with the goal of reducing user effort.

Calculating EigenScores over all pairs of points is prohibitively expensive for an interactive context. However, recall that in our usage context, the user has selected one point of interest and we must suggest options for a second point to pair with the first for a potential user constraint. Evaluating possibilities for just the choices of a second point requires only $(N-1)$ evaluations of EigenScore. We further limit the number of suggestions the user sees to some proportion $k \in (0,1]$ of the total data to save the user from examining every point. Rather than returning a fully ranked list of the top $k*(N-1)$ of the $(N-1)$ total points, we want to choose a *diverse* set of points for consideration. Our rationale is that we expect high EigenScores to correspond to pairs of points where user constraints would cause big updates to the model, but these may not be good updates. For example, outliers in the dataset will often contribute to high EigenScores, but should not necessarily be used in constraints.

To create the desired set of suggestions, we first cluster the data (using the current learned distance function), then sort the points in each cluster c by their EigenScore and return the top $k*|c|$ points within each cluster. This process is detailed in Algorithm 2.

The performance of our implementation is critical to demonstrating the feasibility of this technique for interactive systems. Our prototype system provides EigenSense recommendations on demand as response to interaction with a visualization. The current implementation connects to MATLAB from C# via a COM interface to take advantage of the Lanczos algorithm for quickly calculating the dominant eigenvector. Still, as an example of performance capability, on a laptop with an Intel i5 480M processor, for a dataset of about 200 points with about 20 dimensions, an EigenSense response takes about one second.

Algorithm 2: EigenSense

```
Input: Initial point $x_i$, distance matrix $D$, set of clusters $C$,
       threshold $k$
Output: $S$, a set of model-critical points
foreach cluster $c \in C$ do
    foreach point $x_j \in c$ do
        Compute $ES_{ij} = EigenScore(x_i, x_j, D)$
    Let $S_c$ be the set of $k \times |c|$ points with the highest $ES_{ij}$
Let $S = \bigcup S_c$
return $S$
```

6. EXPERIMENTS AND RESULTS

We validate the accuracy and effectiveness of our proposed method through two experiments on test datasets from the UCI Machine Learning Repository[7]. First, we compare EigenScores against actual values of the quantity they estimate and see that they could be an effective low-cost estimator of model change. Second, we evaluate the accuracy of EigenSense by considering the quality of the sets of points it offers to users compared against the ground-truth best points. We show that guided by EigenSense, a user could pick high-quality inputs while reviewing small amounts of data.

6.1 Experiment 1: Compare To Ground Truth

In this experiment we evaluate the EigenScores by comparing them directly to the value they are attempting to estimate. Recall that in our interactive metric learning context, EigenScores are an estimate of how much the distance matrix, as a stand-in for the distance metric itself, would be changed by constraining a given pair of points. The ground truth is prohibitively expensive to calculate for an interactive system, but can be prepared offline.

For three datasets, starting from scratch with no constraints, we used our prototype system (with interactive metric learning based on Brown et al. [14]) to calculate for each possible pair of points the actual change in distance function resulting from constraining the pair. The graphs in Figure 2 show the comparison of these values to the EigenScores. We use weighted Euclidean distance functions, thus the initial distance function is parameterized by the vector $\Theta_{init} = (1/M, ..., 1/M)$ of length M. In the graphs, the horizontal axis is the change in the distance metric from applying the constraint and the vertical axis is the EigenScore:

$$1 - \text{CosineSimilarity}(\Theta_{init}, \Theta_{post\_constraint})$$

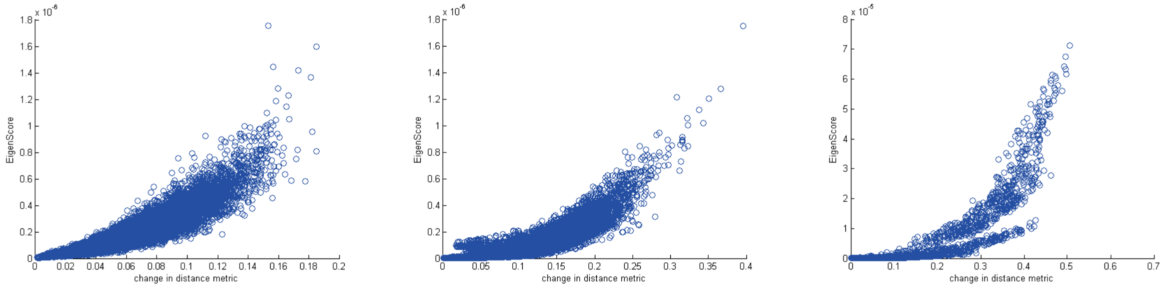
Although the correlations between EigenScores and actual distance metric change are not obvious linear relationships, it is apparent from visual inspection that the quantities are related. This first pass evaluation shows the promise of EigenScores as an estimate of distance metric change, which implies that it could be an inexpensive way to predict model change for interactive machine learning.

6.2 Experiment 2: Evaluate Suggestion Quality

The goal of this experiment is to determine the quality of EigenSense recommendations by comparing them to the choices an oracle would make. Given an oracle that can rank all user feedback options in terms of which yield the best distance functions, we look to see how the EigenSense recommendations rank in that list.

We simulate choices of a point of interest x_i by the user, and then compute both the oracle's ranking of all possible constraint pairs with x_i, and the set of EigenSense options that would be presented to the user. Specifically, the oracle takes advantage of the labels for our test datasets to calculate, for all pairs of constraints that include x_i, the accuracy (with k-NN) of the distance metric resulting from an update with the given constraint. That is, given one point x_i, the oracle applies the system's metric learning algorithm with each constraint pair $(x_i, x_j)\ \forall x_j$, and evaluates each resulting distance function at classifying the data with k-NN. The accuracy scores of these evaluations provide a ranking over the constraint pairs. We compare the EigenSense options against the oracle ranking by finding the EigenSense recommendation with minimum oracle rank.

Figure 3 shows the results of our experiment. Each graph line corresponds to a different dataset, and each plotted point represents an average over ten simulations, each of which simulated ten user inputs. Simulated users picked a first point randomly then some



(a) Wine (b) Ion (c) Parkinsons

Figure 2: Experiment 1 – In this comparison between EigenScores and the quantity they estimate, each point in each graph represents a pair of data points from the appropriate dataset. The horizontal axis shows the actual amount the underlying distance function changes when a given pair of points is constrained together. The vertical axis shows the EigenScore for that pair of points.

(not necessarily optimal) EigenSense recommendation for the second. In total, each plotted point represents 100 uses of EigenSense. The horizontal axis is the *k* parameter of EigenSense (see Algorithm 2 and Section 5.2), which determines how many points will be shown to the user. Because the vertical axis shows the best oracle ranking of the EigenSense points, lower scores are better. It is no surprise that with larger values of *k*, where the user is being shown more points, the opportunity for the best-ranked points to be included is higher. Using a low value of *k* means showing the user few points and saving effort, whereas using a high value means showing more points but having a better chance to show the absolute best ones. The results of this experiment suggest that, depending on the dataset, a user could give strong feedback to a metric learner while only reviewing less than ten percent of the data, or in some cases, substantially less.

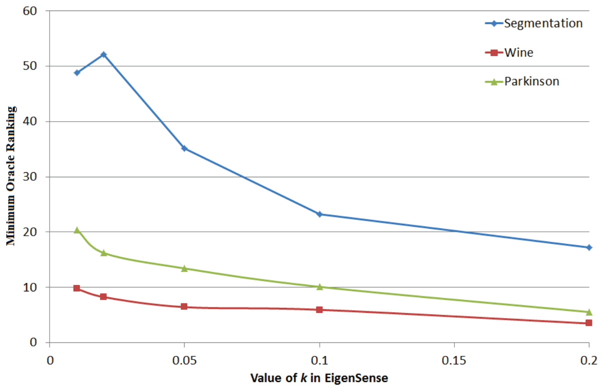


Figure 3: Experiment 2 – The horizontal axis shows values of the *k* parameter to EigenSense, i.e. how much data is shown to the user. The vertical axis shows the minimum (best) rank of the EigenSense recommendations in the oracle's ordering of all possible point pairs. Note that, as expected, as more data is shown to the user (*k* increases), there is more chance of the best possible options being revealed (rank decreases). Even with a small amount of data revealed, the EigenSense suggestions provide strong options.

7. FUTURE WORK

Although we have collected the presented evidence of EigenSense's effectiveness, there are opportunities for improving the algorithm itself. For example, there are several variations on how to generate pairwise distance or similarity matrices. Further, the performance of the implementation could be improved by using a library implementation of the Lanczos method for calculating the dominant eigenvector, instead of using MATLAB via COM calls.

The performance improvement is critical for the main thrust of future work, which is to complete the evaluation of the technique by testing it with human subjects. In particular, participants in a user study will use the tool to cluster some images with known classes. We can then evaluate their comfort with the tool, confidence in the recommendations, and progressive accuracy of the distance metrics learned from their inputs to see if they do better with or without EigenSense.

8. CONCLUSION

This paper contributes to the study of interactive metric learning by applying active learning to reduce the workload of the human actor. We introduced the concept of EigenScores based on eigenvector sensitivity of distance matrices, and then applied these to create the EigenSense algorithm, which identifies and recommends points for user consideration given an initial exploratory direction. We presented evidence of the effectiveness of the algorithm by demonstrating its correlation with ground-truth values of the quantity it estimates, and then by showing the frequency with which EigenSense presents the best possible option to users. Our results indicate that EigenSense could help save human workload by vastly reducing the number of data points to be considered while maintaining near-optimal metric learning results.

9. REFERENCES

[1] N. Abe and H. Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML)*, pages 1–9, 1998.

[2] R. Agrawal, R. Rantzau, and E. Terzi. Context-sensitive ranking. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, pages 383–394. ACM, 2006.

[3] S. Amershi, J. Fogarty, A. Kapoor, and D. S. Tan. Effective end-user interaction with machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1529–1532, 2011.

[4] S. Amershi, J. Fogarty, and D. Weld. Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 21–30. ACM, 2012.

[5] S. Amershi, B. Lee, A. Kapoor, R. Mahajan, and B. Christian. Human-guided machine learning for fast and accurate network alarm triage. In *Proceedings of the International Joint Conference on Artifical Intelligence (IJCAI)*, pages 2564–2569, 2011.

[6] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. Interactive visual clustering of large collections of trajectories. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 3–10. IEEE, 2009.

[7] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[8] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6(6):937–965, 2005.

[9] S. Basu, A. Banerjee, and R. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the IEEE International Conference on Data Mining*, pages 333–344, 2004.

[10] S. Basu, S. M. Drucker, and H. Lu. Assisting Users with Clustering Tasks by Combining Metric Learning and Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 394–400, 2010.

[11] S. Basu, C. Jacobs, and L. Vanderwende. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1:391–402, 2013.

[12] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 81–88, 2004.

[13] J. Broekens, T. Cocx, and W. Kosters. Object-centered interactive multi-dimensional scaling: Ask the expert. In *Proceedings of the Eighteenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC)*, pages 59–66, 2006.

[14] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 83–92. IEEE, 2012.

[15] C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 111–118, 2000.

[16] E. Carroll, M. Czerwinski, A. Roseway, A. Kapoor, P. Johns, K. Rowan, and M. Schraefel. Food and mood: Just-in-time support for emotional eating. In *Proceedings or the Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 252–257, Sept 2013.

[17] J. Choo, H. Lee, J. Kihm, and H. Park. iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 27–34. IEEE, 2010.

[18] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 539–546, 2005.

[19] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *Proceedings of Twenty-Fourth International Conference on Machine Learning (ICML)*, pages 209–216, 2007.

[20] M. Desjardins, J. MacGlashan, and J. Ferraioli. Interactive visual clustering. In *Proceedings of the Twelfth International Conference on Intelligent User Interfaces*, pages 361–364. ACM, 2007.

[21] S. Ebert, M. Fritz, and B. Schiele. Active metric learning for object recognition. In A. Pinz, T. Pock, H. Bischof, and F. Leberl, editors, *Pattern Recognition*, volume 7476 of *Lecture Notes in Computer Science*, pages 327–336. Springer Berlin Heidelberg, 2012.

[22] S. P. Ellner and J. Guckenheimer. *Dynamic models in biology*. Princeton University Press, 2011.

[23] 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)*, pages 473–482. ACM, 2012.

[24] Y. Freund, H. Seung, Sebastian, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning Journal*, pages 133–168, 1997.

[25] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 19*, pages 513–520, 2004.

[26] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? Metric learning approaches for face identification. In *Proceedings of the International Conference on Computer Vision*, pages 498–505, 2009.

[27] M. Heath. *Scientific Computing*. The McGraw-Hill Companies, Incorporated, 2001.

[28] T. Hofmann and J. Buhmann. Active data clustering. In *Advances in Neural Information Processing Systems 12*, pages 528–534, 1997.

[29] S. Hoi, W. Liu, M. Lyu, and W. Ma. Learning distance metrics with contextual constraints for image retrieval. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2072–2078, 2006.

[30] R. Jin and L. Si. A bayesian approach toward active learning for collaborative filtering. In *Uncertainty in Artificial Intelligence*, pages 278–285, 2004.

[31] D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, pages 590–604, 1992.

[32] P. Melville and R. J. Mooney. Diverse ensembles for active learning. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML)*, pages 74–83, 2004.

[33] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. *Stanford InfoLab*, 1999.

[34] R. Rosales and G. Fung. Learning sparse metrics via linear programming. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 367–373, 2006.

[35] N. Roy and A. Mccallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 441–448, 2001.

[36] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 2010.

[37] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 287–294. ACM, 1992.

[38] C. Shen, J. Kim, L. Wang, and A. Hengel. Positive semidefinite metric learning with boosting. In *Advances in Neural Information Processing Systems 22*, pages 1651–1659, 2009.

[39] S. Stumpf, V. Rajaram, L. Li, W.-K. Wong, M. Burnett, T. Dietterich, E. Sullivan, and J. Herlocker. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies*, 67(8):639–662, 2009.

[40] S. Tong and D. Koller. Active learning for parameter estimation in bayesian networks. In *In Advances in Neural Information Processing Systems 14*, pages 647–653, 2001.

[41] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Journal Of Machine Learning Research*, pages 999–1006, 2001.

[42] L. Torresani and K. C. Lee. Large margin component analysis. In *Advances in Neural Information Processing Systems 20*, pages 1385–1392, 2007.

[43] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591. IEEE, 1991.

[44] F. Wang, J. Sun, T. Li, and N. Anerousis. Two heads better than one: Metric+active learning and its applications for it service classification. In *Ninth IEEE International Conference on Data Mining (ICDM)*, pages 1022–1027, Dec 2009.

[45] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 19*, pages 10:207–244, 2006.

[46] K. Weinberger and L. Saul. Fast solvers and efficient implementations for distance metric learning. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML)*, pages 1160–1167, 2008.

[47] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 975–982. IEEE, 1999.

[48] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 505–512, 2002.

[49] Q. Xu, M. desJardins, and K. Wagstaff. Active constrained clustering by examining spectral eigenvectors. In *Discovery Science*, pages 294–307, 2005.

[50] L. Yang and R. Jin. Bayesian active distance metric learning. *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007.

[51] Y. Ying, K. Huang, and C. Campbell. Sparse metric learning via smooth optimization. In *Advances in Neural Information Processing Systems 22*, pages 2214–2222, 2009.

[52] Y. Ying and P. Li. Distance metric learning with eigenvalue optimization. In *Journal of Machine Learning Research*, pages 1–26, 2012.

[53] T. Zhang and F. Oles. A probability analysis on the value of unlabeled data for classification problems. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 1191–1198, 2000.

[54] Y. Zhang, W. Xu, and J. Callan. Exploration and exploitation in adaptive filtering based on bayesian active learning. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, pages 896–903, 2003.

CrowdMGR: Interactive Visual Analytics to Interpret Crowdsourced Data

Abon Chaudhuri[*]
Intel, Hillsboro, USA
abon.chaudhuri@intel.com

Mahashweta Das[†]
HP Labs, Palo Alto, USA
mahashweta.das@hp.com

ABSTRACT

Crowdsourcing is popularly defined as a paradigm that utilizes human processing power to solve problems that computers cannot yet solve. While recent research has been dedicated to improve the problem-solving potential of crowdsourcing activities, not much has been done to help a user quickly extract the valuable knowledge from crowdsourced solutions to a problem, without having to spend a lot of time examining all content in details. Online knowledge-sharing forums (Y! Answers, Quora, and StackOverflow), review aggregation platforms (Amazon, Yelp, and IMDB), etc. are all instances of crowdsourcing sites which users visit to find out solutions to problems. In this paper, we build a system CrowdMGR that performs visual analytics to help users manage and interpret crowdsourced data, and find relevant nuggets of information. Given a user query (i.e., a problem), CrowdMGR returns the solution, referred to as the SolutionGraph, to the problem as an interactive canvas of linked visualizations. The SolutionGraph allows a user to systematically explore, visualize and extract the knowledge in the crowdsourced data. It not only summarizes content directly linked to a user's query, but also enables her to explore related topics within the temporal and topical scope of the query and discover answers to questions which she did not even ask. In the demonstration, participants are invited to manage and interpret crowdsourced data in StackOverflow and Computer Science Stack Exchange, question and answer site for students, researchers and practitioners of computer science.

1. INTRODUCTION

Crowdsourcing is the practice of soliciting services, ideas, solutions, or content from an undefined, generally large group of people in the form of an open call. It is also popularly defined as *a paradigm that utilizes human processing power to solve problems that computers cannot yet solve* [7]. Crowdsourcing has received a lot of attention lately from researchers for its potential in solving problems, often unsolvable by computers, by tapping in to the collective intelligence of the crowd. Efforts have been dedicated to designing the optimal task and workflow, recruiting people by studying behavioral and cognitive biases, incentivizing the crowd, processing crowdsourced data to sift value, etc. However, not much has been done to help a user quickly extract the valuable knowledge from crowdsourced solutions to a problem, without having to spend a lot of time examining them in details. Online knowledge-sharing forums (Y! Answers, Quora, and StackOverflow), review aggregation platforms (Amazon, Yelp, and IMDB), etc. are all instances of crowdsourcing sites which users visit to find out solutions to problems. For example, a user may visit Stack Overflow[1] to find the answer to the problem *Is Java "pass-by-reference" or "pass-by-value"?*. Stack Overflow has 47 solutions to the problem and it would not be possible for the user to find the pertinent answer without examining the detailed textual information, often conflicting, at her disposal. Similarly, a user may visit Yelp[2] to find the answer to the problem *Is "B Patisserie" a healthy bakery to visit in the San Francisco neighborhood?*. Yelp has over 500 solutions to the problem (i.e., reviews for the bakery) that the user needs to go through in order to make her decision. Note that, the crowdsourced data in Stack Overflow concerns facts and information while that in Yelp is more about opinion and judgment. However, the task of eliciting the "solution" for the "problem" from the crowdsourced data remains the same across both the applications.

In this paper, we develop a framework that addresses this need. Our system, called CrowdMGR[3] performs analytics to help users *manage* and *interpret* crowdsourced data, and find *relevant* nuggets of information. Given a user query (i.e., a problem), CrowdMGR returns the solution, referred to as the SolutionGraph[4], to the problem as an interactive canvas of linked visualizations. The purpose of SolutionGraph is to enable a user to quickly access the knowledge in the crowdsourced data, in addition to the information that current crowdsourcing sites showcase. Over the past decade, researchers have developed techniques to summarize user-generated content in review sites, internet forums, blogs, etc. [3][5][6]. SolutionGraph not only summarizes content directly linked to a user's query, it also enables her to explore related topics within the *topical* and *temporal* scope of the query and discover answers to questions which she did not even ask. We present our SolutionGraph as an intuitively intelligible visual form, that incorporates graph drawing methods and geometric techniques for high dimensional data visualization, in order to communicate complex analytical information effectively. Its interactive exploration feature allows a user to seamlessly navigate from the high-level overview to the desired levels of granularity and back.

[*] Authors are listed alphabetically

[†] Majoriy of work done while at University of Texas, Arlington

[1] http://stackoverflow.com/

[2] http://www.yelp.com/

[3] Abbreviated from Crowd Manager

[4] Name draws inspiration from Knowledge Graph

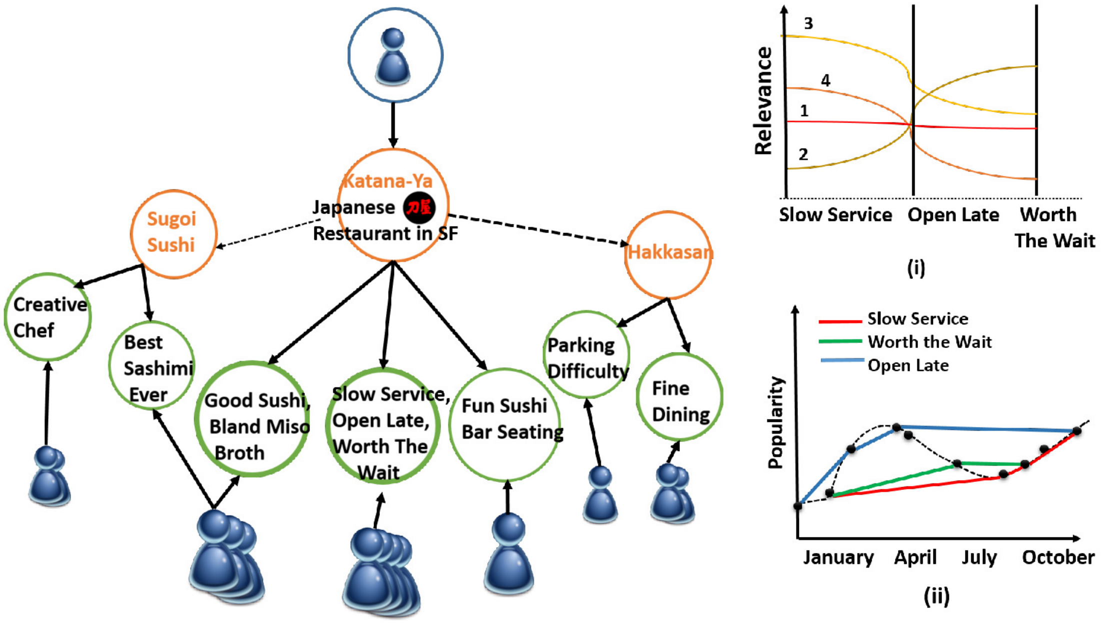


Figure 1: Example SolutionGraph for a query about a Japanese Restaurant in San Francisco; (i) Topical Analysis and (ii) Temporal Analysis of the Answer Node {Slow Service, Open Late, Worth the Wait}

Let us explain our system CrowdMGR and SolutionGraph with a simple illustrative example, presented in Figure 1. Suppose, a user wants to find out *if she will like to visit a particular Japanese restaurant "Katana-Ya" in San Francisco.* Given her query, the SolutionGraph in Figure 1 not only returns the reviews for the restaurant, but also returns two other related restaurants - one Japanese, "Sugoi Sasha" and one Asian Fusion, "Hakkasan" - in the neighborhood and the reviews for them. Since the number of reviews for a restaurant is huge, we aggregate the reviews based on content similarity. If the user is interested in eating Sashimi at a Japanese restaurant in San Francisco at that time, the graph helps her discover a restaurant that meets her preferences better than the one she is querying. Note that, there exists a user who has reviewed both the query restaurant under consideration and the related Japanese restaurant, the former for Sushi and the latter for Sashimi. If the user is interested in eating only at the particular restaurant she is querying, the graph helps her quickly access the broad summary of the feedback it has received. SolutionGraph also allows the user to interact with the system and obtain a detailed insight of the temporal and topical trends of each aggregate answer, as shown in Figure 1-(i) and Figure 1-(ii). Topical analysis of the answer node {Slow Service, Open Late, Worth the Wait} in Figure 1-(i) helps the user access the relevance of the keywords in each of the reviews, that are aggregated. If the user is interested in reading a review about the keyword 'Slow Service', she may read Review 3 in details. Moreover, since the plot suggests that Review 3 and Review 4 are similar in their content, she can readily filter out Review 4. Figure 1-(i) reveals the temporal trend of the answer node {Slow Service, Open Late, Worth the Wait}. The user may note that the keyword 'Open Late' has been frequently mentioned in the reviews in the summer months, while the frequency of the keyword 'Slow Service' has steadily increased over time.

The two main technical challenges in achieving the objective of our system is: (i) how to select the solution nodes for the user query, i.e., problem node in the SolutionGraph; and (ii) how to discover the problem nodes related to the user query in the SolutionGraph. Both the goals are wedded to the definition of *relevance* measure that decides what CrowdMGR intends to show to a user. In this study, our goal is to not advocate one particular measure over another. Rather, we focus on defining the problem framework and demonstrate the utility of our system for managing and interpreting crowdsourced data. Online sites today usually sort user reviews, answers, etc. by decreasing order of popularity (i.e., how many people found the review useful), recency in activity, etc. Ghose et.al [2] has designed review ranking strategies that orders reviews based on their expected helpfulness and expected effect on sale. In this work, we transform each solution to a multi-dimensional weighted feature vector of keywords and employ K Means Clustering that associates similar feature vectors and dissociates dissimilar vectors, where the extent of association (or, dissociation) is measured by the Euclidean distance between the vectors. Several online sites employ natural language processing techniques and machine learning approaches to identify and return list of content related to user query. In this work, we represent each problem as a boolean vector of keywords and employ Jaccard similarity coefficient to identify the related problems. We use force-directed graph drawing algorithm to visualize the graph. Interactive visual analysis of the SolutionGraph to cater to a user's cognitive needs and aid further explorations possess additional challenges. We use parallel-coordinate plots to visually capture and present the topical diversity among similar answers and a 3D point-based plot enhanced by novel visual cues to visually highlight the temporal trend of topics in the SolutionGraph. Note that, CrowdMGR is a real-time system and hence the task of building the SolutionGraph for a user query incurs computational challenges too.

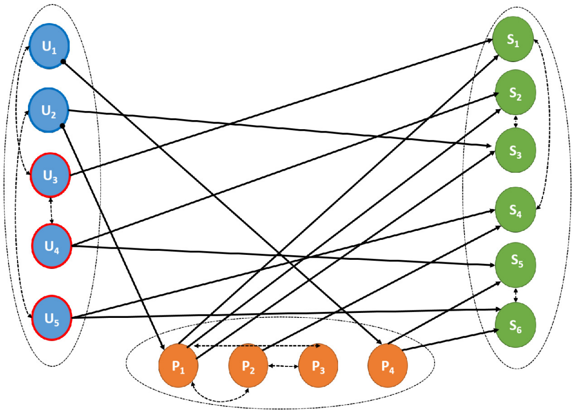


Figure 2: CrowdMGR **Data Model**

2. CrowdMGR DESIGN

The focus of our work is to provide a framework for organizing crowdsourced data in order to help a user access relevant content effectively and efficiently. We first introduce our data model, then discuss our mining problem and algorithmic solution, and finally present our analytics and interactivity features.

2.1 Data Model

A crowdsourcing site, as shown in Figure 2, contains heterogeneous information and can be modeled as a directed tri-partite graph $\mathtt{G}$:

- Nodes: Users ($\mathtt{U}$), Problems ($\mathtt{P}$), and Solutions ($\mathtt{S}$) co-exist in the graph. Note that, there are two kinds of users: Problem giver ($\mathtt{U_P}$) and Solution giver ($\mathtt{U_S}$).
- Inter-relational edges: Edges between user nodes, problem nodes and solution nodes can be derived from the explicit interactions in the crowdsourced data. For a user $u \in \mathtt{U}$, there exists an edge from u to a problem $p \in \mathtt{P}$ or to a solution $s \in \mathtt{S}$, depending on $u \in \mathtt{U_P}$ or $u \in \mathtt{U_S}$. There also exists an edge from a node $p \in \mathtt{P}$ to a node $s \in \mathtt{S}$ if s is a solution for the problem p.
- Intra-relational edges: The set of nodes in the partite $\mathtt{P}$ share edges based on content similarities. The set of nodes in the partite $\mathtt{U}$ share edges based on social network ties, demographic profile information overlap, etc. The set of nodes in the partite $\mathtt{S}$ share edges based on semantic relatedness.
- Node weight: User nodes are weighted by their qualification score, problem and solution nodes are weighted by the aggregated count of votes (up, down) they have received. Instead of scalars, the weights can be vectors, e.g., weighted vector of relevance score of keywords in the solution nodes.

For example in Figure 2, $\mathtt{U} = \{u_1, u_2, u_3, u_4, u_5\}$ where $u_1, u_2 \in \mathtt{U_P}$ and $u_3, u_4, u_5 \in \mathtt{U_S}$; $\mathtt{P} = \{p_1, p_2, p_3, p_4\}$; $\mathtt{S} = \{s_1, s_2, s_3, s_4, s_5, s_6\}$. s_1, s_2, s_3 are solutions to user u_1's problem p_1 provided by users u_2, u_3, u_4 respectively.

2.2 Problem Overview

Given crowdsourced data as a tripartite graph $\mathtt{G}$, a user u_i ($u_i \in \mathtt{U_P}, \mathtt{U_P} \subseteq \mathtt{U}$) and her query p_j ($p_j \in \mathtt{P}$), CrowdMGR identifies the subgraph $\mathtt{G'}$ from $\mathtt{G}$ that contains:

- Set $\mathtt{P'}$ of k_p nodes ($\mathtt{P'} \subseteq \mathtt{P}$) related to p_j by measure $\mathcal{A}$
- Set $\mathtt{S'}$ of k_s nodes ($\mathtt{S'} \subseteq \mathtt{S}$) having directed edges from $\{p_j \cup \mathtt{P'}\}$ and aggregated by measure $\mathcal{B}$
- Set $\mathtt{U'}$ of nodes ($\mathtt{U'} \subseteq \mathtt{U}, u_i \in \mathtt{U_P}, \mathtt{U'} - u_i \subseteq \mathtt{U_S}$) having directed edges from $\{p_j \cup \mathtt{P'}\}$

The subgraph $\mathtt{G'}$ is the SolutionGraph for the query. It can also be classified as a semantic graph [1] consisting of heterogeneous nodes and links that carry semantic information in them.

Measure $\mathcal{A}$: The objective of measure $\mathcal{A}$ is to select the top-k_p of neighboring problems in $\mathtt{P}$ that are related to user query p_j. In our system, we consider the popular Jaccard similarity coefficient. We use the keyword extraction toolkit Alchemy API[5] to extract keywords from the problems in $\mathtt{P}$. Thus, each problem is represented as a boolean vector of size n_p, where n_p is the the total number of distinct keywords extracted from $\mathtt{P}$. The Jaccard measure help us determine the top-k_p related problem nodes. Thus, we only leverage the intra-relational edge information between the problem nodes. We may employ the intra-relational edge information in all three partites for this purpose, e.g., people who viewed this restaurant also viewed feature in Yelp.

Measure $\mathcal{B}$: The objective of measure $\mathcal{B}$ is to determine the k_s solution nodes in $\mathtt{S}$ that are to presented in $\mathtt{G'}$ as solution nodes to user query p_j. If the number of solutions for a problem is not large ($\leq n_s$, $\mathtt{G'}$ may just comprise of the solution nodes. However, the number of answers per question is usually large, and answers often receive multiple comments, e.g., in Stack Overflow. Hence, we aggregate the set of all solutions for a problem to determine k_s nodes. In this work, we consider Euclidean distance and employ K Means clustering to group similar vectors together.

2.3 Visualization

Our system presents the analytics report of the crowdsourced data in a visually engaging way:

SolutionGraph: The graph is presented as a node-link style visualization. It is generated using Kamada-Kawai layout algorithm [4]. As discussed in Section 2, there are three types of entities in the graph: user nodes, problem nodes, and solution nodes. Since this is a semantic graph, we use different colors and shapes for representing the different entities and links. Some design choices regarding encoding information in the graph have been made very carefully to facilitate analytics (to be explained in Section 2.4). The visualization of SolutionGraph delivers information on demand to avoid clutter. Before generating the graph, the user can control its size by tuning two parameters: maximum number of related problems and maximum number of solutions for each problem. However, it is also possible to generate a content-rich graph and then control the amount of information to show by applying filters on graph based on node type, degree, popularity and so on. For example, the user may want to see only the highly voted answers for each problem for further analysis. The graph view is associated with two other linked visualizations.

Topical Analysis: The solution nodes are the key entities of interest in the graph. They contain weighted vector of relevance score of keywords extracted from the solutions and

[5] http://http://www.alchemyapi.com/api/keyword-extraction/

broadly summarizes the content in the nodes. Recall that each solution node in the SolutionGraph is an aggregated group of similar answers in the crowdsourced data obtained by K Means clustering. The individual answers belonging to a solution node, though similar, are not identical. To help the user access answer(s) based on keywords, we employ parallel coordinates (PC) plot - a technique known for visualizing high dimensional data - as shown in as shown in Figure 1-(i). Each vertical axis in the PC plot denotes a keyword. The relevance is denoted as a point on the axis. Hence, a feature vector of keywords is represented by a line connecting the points on each axis. PC plot can effectively highlight how similar two answers are even when they belong to the same solution node. To accommodate large number of axes (keywords), we use zoomable PC plot which focuses on a few keywords at a time.

Temporal Analysis: In crowdsourcing sites, the answers to a question are usually posted and voted over a long period of time. A topically relevant answer may actually have lost relevance over time. For example, if the restaurant in Figure 1 was being praised for 'Good Sushi' 3 years back but could not maintain its reputation, a temporal analysis of the keyword 'Good Sushi' should reflect that. To capture these temporal characteristics, we present all the individual answers aggregated in a solution node on a 2D scatterplot enhanced with visual cues, as shown in Figure 1-(ii). The answers are temporally ordered along the horizontal axis, the y-axis captures the popularity of each answer (number of upvotes - number of downvotes). A selectable list of keywords (a subset containing the frequent ones) is presented alongside the plot. As the user selects a keyword, a spline curve connects the answers that contains that keyword. This overlaid curve on top of the scatterplot clearly highlights many facts, e.g., if that keyword has appeared consistently over time, if there is a correlation between the popularity of an answer and existence of that keyword, etc.

2.4 Interaction and Analytics

CrowdMGR allows a user to perform analytics by easy and effective interaction with the system in order to help her seamlessly navigate from the high-level overview to the desired levels of granularity. Our system favors analytics in two ways:

By driving user interaction: Each of the three visualizations in Section 2.3 above has information encoded in such a way that it can channel the user's attention to the meaningful components of the SolutionGraph. For example, the sizes of the user nodes in the graph are determined by the user's reputation (measured by how much the community trusts the user, how actively the user participates, etc.) in the crowdsourcing site. Hence, while looking at the creator of a post (problem node or a solution node), the demo participant can get some idea about the creator's credibility which may help her choose or skip a node. Again, the size of a solution node is proportional to the number of individual answers it is aggregating, thereby conveying the solution highlights and content distribution to the user effortlessly.

By responding to user interaction: As the user views the visual analytics result returned by our system, she is presented with opportunities to drive the analytic process forward. For example, as the user clicks on a solution node on the SolutionGraph, the topical and temporal analysis plots are populated for further analysis. Again, clicking on a curve in the topical analysis plot brings out the actual text of the answer with keyword highlighted. Thus, our interaction framework enables a user to navigate through various levels of detail, otherwise unmanageable. At any point of time, the user can restart the analysis, or step back without having to click through a series of browser back buttons, or having to scroll a long way up.

3. DEMONSTRATION

The CrowdMGR system can work on any crowdsourcing site that provides data as descried in Section 2.1. For the purpose of the demo, we use publicly available Stack Overflow and Stack Exchange data[6]. As of August 2012, the Stack Exchange dump for Computer Science has 10,529 registered users; 4,926 questions of which 2,487 have accepted answers; 7,122 answers; 25,042 comments; and 60,035 votes. The Stack Exchange dump for Programmers has 96,744 registered users; 29,025 questions of which 17,451 have accepted answers; 116,491 answers; 282,421 comments; and 1,391,975 votes. The Stack Overflow dump is even bigger having over 1.3 million registered users and over 4 million questions.

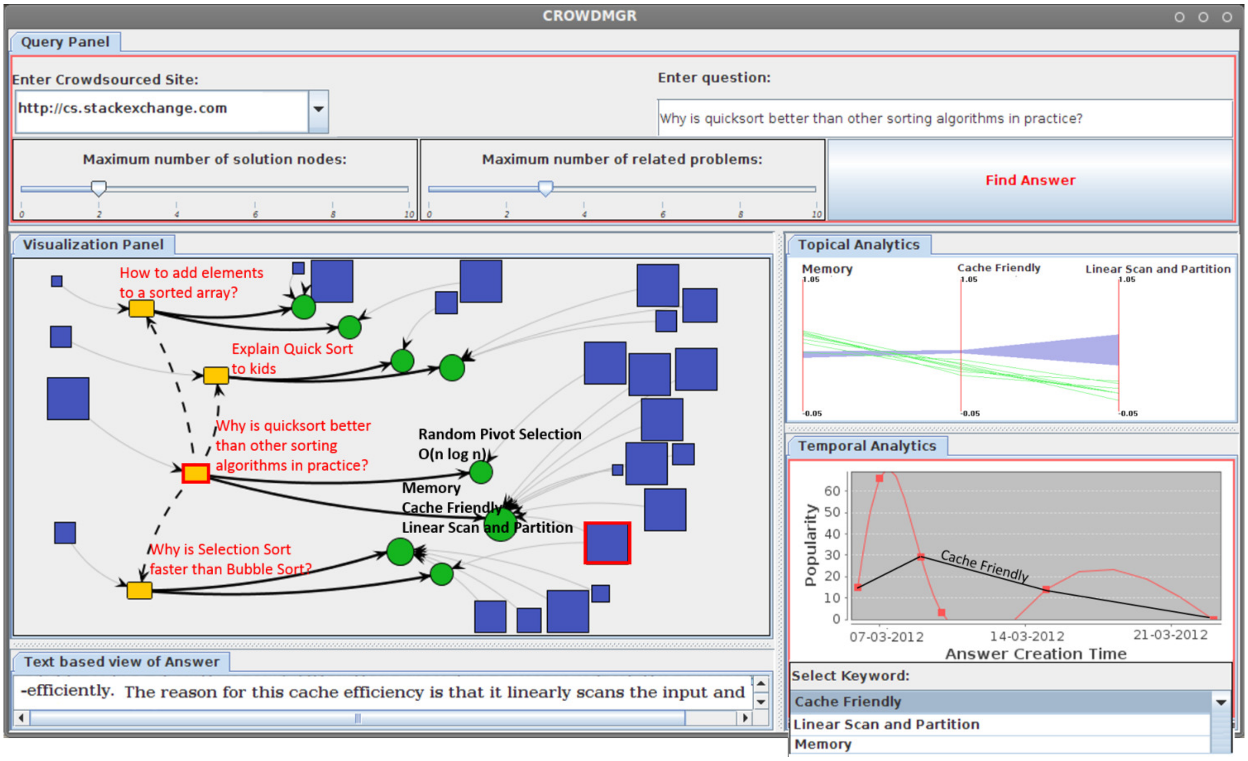
3.1 Demo

Our demo allows the audience to use a standalone application as shown in Figure 3 and specify search query in the scope of the crowdsourcing site under consideration. Example queries include: *Why is quicksort better than other sorting algorithms in practice?*, *What are the text editors for large files?*, and so on. If the query entered by the user is not present in the crowdsourcing site, we identify the question that is most similar to the user query and proceed with it. The audience can specify other query settings such as: maximum number of solution nodes(i.e., k_s) and maximum number of related problems (i.e., k_p) they want to see in the SolutionGraph. They can select any one of the solution nodes and observe the topical and temporal trends of the keywords in it. They can drill down deeper to view the actual textual solution too. Such exploration will give the audience a deeper appreciation of our systemŠs utility to aid users extract the valuable knowledge from crowdsourced data quickly, and it's superiority over content displayed in existing crowdsourcing sites.

3.2 Use Case Study

Let us illustrate our demo with a detailed use case study. Suppose, a user selects the crowdsourcing site `http://cs.stackexchange.com/` and submits the query *Why is quicksort better than other sorting algorithms in practice?*. The maximum number of solution nodes and the maximum number of related problem nodes she submits as input are 2 and 3 respectively. On clicking Find Answer button, The SolutionGraph is generated in the Visualization panel. The problem nodes are in orange (with the user query node having a red border); the solution nodes are in green; and the user nodes are in blue. The size of the user node is proportional to the user's reputation score in the site. The size of the solution node is proportional to the number of individual answers aggregated in it. Note that in the Solution Graph, there exists a user (the node having a red border) who has submitted answer to the user query as well as to

[6] http://http://stackexchange.com/



CROWDMGR
Query Panel
Find Answer
Visualization Panel
How to add elements to a sorted array?
Explain Quick Sort to kids
Why is quicksort better than other sorting algorithms in practice?
Random Pivot Selection O(n log n)
Memory
Cache Friendly
Linear Scan and Partition
Why is Selection Sort faster than Bubble Sort?
Topical Analytics
Memory
Cache Friendly
Linear Scan and Partition
1.05
-0.05
Temporal Analytics
Popularity
60
50
40
30
20
10
0
Cache Friendly
07-03-2012
14-03-2012
21-03-2012
Answer Creation Time
Text based view of Answer

Enter Crowdsourced Site: http://cs.stackexchange.com

Enter question: Why is quicksort better than other sorting algorithms in practice?

Maximum number of solution nodes: 0 2 4 6 8 10

Maximum number of related problems: 0 2 4 6 8 10

-efficiently. The reason for this cache efficiency is that it linearly scans the input and

Select Keyword:
Cache Friendly
Linear Scan and Partition
Memory

Figure 3: User Interface of CrowdMGR

a related problem, *Why is Selection Sort faster than Bubble Sort?*. Suppose, the user wants to explore the most popular solution node for the query that have 8 answers aggregated, i.e, the solution node mentioning 'Memory', 'Linear Scan and Partition', and 'Cache Friendly'. On clicking that node, the Topical Analytics and Temporal Analytics plots are displayed. On selecting one of the green curves in the Topical Analytics plot, the text of the actual answer is shown in the Text Based View of the Answer panel (bottom left). Also, the user can select a keyword from the drop down option in Temporal Analytics panel to observe the popularity of a keyword, overlaid on top of the popularity of all solutions over time.

4. CONCLUSION

Given a user query, i.e., a problem, CrowdMGR generates a SolutionGraph that helps user manage and interpret crowdsourced data and extract valuable nuggets, i.e., solutions, from it. It enables a user to conduct temporal and topical analysis of the solutions returned for the problem by the system, as well as discover answers to questions which she did not even an ask. Our demo allows users to generate and interactively explore interesting SolutionGraph-s for questions in Stack Overflow and Computer Science Stack Exchange.

Our work is a preliminary look at a very novel problem of research in crowdsourcing and there appear to be many exciting directions of future research. Our immediate goal is to improve the efficiency and effectiveness of our system by employing sophisticated techniques in order to conduct big data analytics and identify nodes to be displayed in the SolutionGraph. Since user-generated content is always on the rise, we plan to handle updates and insertions of new users, problems, and answers in our system. We also intend to investigate the applicability of our framework to other forms of crowdsourced data involving images and videos, as well as other novel applications, e.g., how SolutionGraph can improve the quality of recommendation, etc.

5. REFERENCES

[1] T. Coffman, S. Greenblatt, and S. Marcus. Graph-based technologies for intelligence analysis. *Communications ACM*, 47(3):45–47, 2004.

[2] A. Ghose and P. G. Ipeirotis. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *ICEC*, pages 303–310, 2007.

[3] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD*, pages 168–177, 2004.

[4] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31(1):7–15, 1989.

[5] L.-W. Ku, Y.-T. Liang, and H.-H. Chen. Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 100–107, 2006.

[6] Y. Liu, S. Li, Y. Cao, C.-Y. Lin, D. Han, and Y. Yu. Understanding and summarizing answers in community-based question answering services. In *COLING*, pages 497–504, 2008.

[7] L. Von Ahn. *Human Computation*. PhD thesis, 2005.

Formalising the subjective interestingness of a linear projection of a data set: two examples

[This is a ‘work-in-progress’ paper]

Tijl De Bie
University of Bristol, Intelligent Systems Laboratory
Bristol, United Kingdom
tijl.debie@gmail.com

ABSTRACT

The generic framework for formalising the subjective interestingness of patterns presented in [2] has already been applied to a number of data mining problems, including itemset (tile) mining [3, 8, 9], multi-relational pattern mining [18, 19, 20], clustering [10], and bi-clustering [12, 11]. Also, it has been pointed out without providing detail that also Principal Component Analysis (PCA) [7] can be derived from this framework [2]. This short note describes work-in-progress aiming to show in greater detail how this can be done. It also shows how the framework leads to a robust variant of PCA when used to formalise the subjective interestingness of a data projection for a user who expects outliers to be present in the data.

Categories and Subject Descriptors

H.4 [**Information systems applications**]: Data mining

General Terms

Theory, Algorithms

Keywords

Principal Component Analysis, Robust PCA, subjective interestingness

1. INTRODUCTION

This short note gives two examples of how the framework from [2], can be used to formalise the subjective interestingness of a linear projection of a data set. Thus it illustrates how the framework can lead to different approaches for linear dimensionality reduction depending on the prior beliefs of the user, illustrating the importance of this initial interaction with the user.

We consider two types of prior beliefs in particular. The first one of these leads to an algorithm identical to Principal Component Analysis (PCA). The second one, which is suited for users who feel they have no accurate belief about the spread of the data but only about the order of magnitude of that spread, can be thought of as a robust (outlier insensitive) alternative to PCA that appears to be novel.

This note sweeps all details under the carpet, and leaves a number of important questions unanswered. These details and questions will be resolved in a later publication. The hope is that this short note further demonstrates the usefulness of the framework from [2] across the breadth of exploratory data mining research. It helps in elucidating when a certain pattern is interesting to a given user, depending on the beliefs of that user.

In this particular study, it shows that PCA is not the best approach for users who anticipate the presence of outliers. While this will come as no surprise to many, this is a formal and rigorous demonstration of why that is the case, and additionally offers an alternative method that is appropriate when outliers are expected by a user.

2. SUBJECTIVE INTERESTINGNESS IN A NUTSHELL

2.1 Notation

Scalars are denoted with standard face, vectors with bold face lower case, and matrices with bold face upper case letters. The i'th data point is denoted as $\mathbf{x}_i \in \mathbb{R}^d$ with d the dimensionality of the data space. The matrix containing all data points transposed $\mathbf{x}_i'$ ($i = 1, \ldots, n$) as its rows is denoted as $\mathbf{X} \in \mathbb{R}^{n \times d}$.

2.2 Projection patterns

In the general framework of [2], we formalised patterns as any property the data satisfies. In this paper, the particular kind of pattern considered can formalised as a constraint on the data of the form:

$$\mathbf{X}\mathbf{w} = \mathbf{p},$$

where $\mathbf{w} \in \mathbb{R}^d$, referred to as a weight vector (also known as the loadings), has unit norm and parameterises the pattern. The vector $\mathbf{p} \in \mathbb{R}^n$ specifies the value of the projections of the data points onto the weight vector $\mathbf{w}$. The fact that the projections of all data points onto a given weight vector $\mathbf{w}$ are equal to specific values is clearly a property a data set may or may not have, and revealing it to a user provides clear information to that user restricting the set of possible values the data set can have.

Although ideally any possible $\mathbf{w} \in \mathbb{R}^d$ can be considered, in practice only a finite though large number of them can be considered due to the lack of finite code for the set of real numbers. Similarly, the values of $\mathbf{p}$ cannot be specified to an infinite accuracy. This short note brushes over these issues, which can be dealt with rigorously by assuming they are specified up to a certain accuracy. A rigorous treatment of these issues is deferred to a later publication.

2.3 The subjective interestingness of projection patterns

In [2], the interestingness of a pattern (defined generically as any constraint on the value of the data) is formalised as the trade-off between the description length of the pattern, and its subjective information content. More specifically, the *subjective interestingness* of a pattern is formalised as its subjective information content divided by its description length. Here we very briefly summarize this framework, and start outlining how it can be applied to the kind of patterns of interest in this paper, namely projection patterns.

It is reasonable to consider the *description length* as constant, independent of $\mathbf{w}$ and $\mathbf{p}$. Indeed, this amounts to assuming that each possible $\mathbf{w}$ requires the same description length, and that $\mathbf{p}$ is shown with constant absolute precision. The latter is the case when e.g. the projections are visualized on a computer screen or printed on paper. If the values of $\mathbf{p}$ are normalised before visualizing, then the description length is not exactly constant as also the normalising factor needs to be specified, which requires a variable length code if the normalisation factor is unbounded. However, in practice this should always account for a very small part of the description length of the pattern.

The *subjective information content* is minus the logarithm of the probability that the pattern is present, where the probability is computed with respect to the so-called background distribution, which represents the belief state of the user about the data.

The belief state can be modelled assuming a certain set of prior beliefs (expressed as constraints on the expected values of certain test statistics given the background distribution). Among all distributions satisfying these constraints, the *background distribution* is the one with maximum entropy.

Each time a pattern is revealed to the user, the user's background distribution changes. More specifically, it is conditioned on the presence of the pattern just revealed.

3. INTERESTING PROJECTIONS WHEN NO OUTLIERS ARE EXPECTED

3.1 Prior beliefs and the background distribution

A user not expecting any outliers will be able to express an expectation about the value of the average two-norm squared of the data points:

$$\mathbb{E}_{\mathbf{X}\sim P}\left\{\frac{1}{n}\sum_{i}^{n}\mathbf{x}_i'\mathbf{x}_i\right\} = \sigma^2.$$

To determine the value for σ user involvement appears to be inevitable at first sight. However, below it will become clear that the ordering of projection patterns according to interestingness is in fact independent of the value of σ, so in practice the exact value will not need to be known.

It is well known (and easy to derive) that the distribution of maximum entropy given this prior belief constraint on the scatter matrix of the data points is a product distribution of multi-variate normal distributions with mean $\mathbf{0}$ and covariance matrix $\sigma\mathbf{I}$. I.e. the density function for each of the data points $\mathbf{x}$ is:

$$p(\mathbf{x}) = \frac{1}{\sqrt{2\pi}^d}\exp\left(-\frac{\mathbf{x}'\mathbf{x}}{2\sigma^2}\right).$$

Thus, the product of n such distributions, one for each of the data points, is the background distribution formalising a user's prior belief state about the data set, when that user does not anticipate the presence of outliers.

3.2 The subjective interestingness of a projection pattern

It is well-known that the probability distribution of an orthogonal transformation of a normal random variable is again a normal random variable, with the same mean and with a covariance matrix that is transformed accordingly. In the current context, with $\mathbf{W}$ an orthogonal matrix (i.e. $\mathbf{W}'\mathbf{W} = \mathbf{W}\mathbf{W}' = \mathbf{I}$), and with $\mathbf{z} = \mathbf{W}'\mathbf{x}$, it holds that:

$$\begin{aligned} p(\mathbf{z}) &= \frac{1}{\sqrt{2\pi}^d}\exp\left(-\frac{\mathbf{z}'\mathbf{z}}{2\sigma^2}\right), \\ &= \prod_{k=1}^{d}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{z_k^2}{2\sigma^2}\right). \end{aligned}$$

I.e., the distribution of $\mathbf{z}$ is a product distribution with a factor for each of the components of $\mathbf{z}$. Thus, the marginal distribution for the first component, z_1, is given by:

$$p(z_1) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{z_1^2}{2\sigma^2}\right).$$

Referring to the first column of $\mathbf{W}$ as $\mathbf{w}$ (and note that $\mathbf{w}'\mathbf{w} = 1$ follows from $\mathbf{W}'\mathbf{W} = \mathbf{I}$), this means that the projections $\mathbf{Xw} = \mathbf{p}$ of all data points follow this normal distribution, and thus the subjective information content of a projection pattern specified by this equality is equal to:

$$\begin{aligned} &\text{SubjectiveInformationContent}\left(\mathbf{Xw} = \mathbf{p}\right) \\ = &-\log\left(p(\mathbf{Xw} = \mathbf{p})\right), \\ = &\frac{n}{2}\log(2\pi) + \frac{1}{2\sigma^2}\mathbf{w}'\mathbf{X}'\mathbf{Xw}. \end{aligned}$$

As the descriptional complexity is constant, this is proportional to the subjective interestingness.

3.3 The maximiser of the interestingness is the maximiser of the variance

PCA's goal is to maximise $\mathbf{w}'\mathbf{X}'\mathbf{X}'\mathbf{w}$ subject to the constraint $\mathbf{w}'\mathbf{w} = 1$, which is clearly equivalent with maximising this subjective interestingness. PCA can thus be regarded as finding the projection pattern with maximal subjective interestingness for the user not expecting any outliers.

3.4 Subsequent iterations

After revealing the first projection pattern, the background distribution is conditioned on the fact that $\mathbf{Xw} = \mathbf{p}$. The updated background distribution is then a product distribution of multivariate standard normal distributions on the subspace orthogonal to $\mathbf{w}$. The result of that is that the subjective information of patterns in subsequent iterations is computed as for the first pattern after deflating the data: considering only the component of the data points orthogonal to $\mathbf{w}$. This is precisely the way PCA works.

4. INTERESTING PROJECTIONS WHEN OUTLIERS ARE EXPECTED

With a slightly different prior belief that assumes the presence of outliers (leading to a heavy-tailed background distribution), a method that can be thought of as a robust version of PCA is obtained.

4.1 Prior beliefs and the background distribution

As prior beliefs, now the following is used:

$$\mathbb{E}_{\mathbf{X}\sim P}\left\{\frac{1}{n}\sum_i^n \log\left(1+\frac{1}{\rho}\mathbf{x}_i'\mathbf{x}_i\right)\right\} \quad = \quad c.$$

This kind of prior belief specifies an expectation on a measure of the spread of the data, which amplifies contributions from points with small norm relative to the data points with large norm through a log transformation. Thus, using such a prior belief rather than say a prior belief on the second moment considers outliers in the data relatively more probable. The smaller the value of ρ, the less important the constant term in the argument of the logarithm will be, the more logarithmic this statistic will therefore vary with the norm of $\mathbf{x}_i$, and thus the more tolerant this model will be to outliers. Informally: rather than determining an expectation on the spread of the data, for small values of ρ it determines an expectation on the *order of magnitude* of the spread of the data.

For convenience in the following derivations, let us introduce the function

$$\kappa(\nu) \quad = \quad \psi\left(\frac{\nu+d}{2}\right) - \psi\left(\frac{\nu}{2}\right),$$

where ψ represents the digamma function. In the sequel the value of $\kappa^{-1}(c)$ will need to be used, denoted as ν for brevity. Then, the initial background distribution can be derived by relying on [22], where it is shown that the maximum entropy distribution subject to the specified prior information is the product of independent multivariate standard t-distributions with density function p defined as:

$$p(\mathbf{x}) \quad = \quad \frac{\Gamma\left(\frac{\nu+d}{2}\right)}{\sqrt{(\pi\rho)^d}\Gamma\left(\frac{\nu}{2}\right)} \cdot \frac{1}{\left(1+\frac{1}{\rho}\mathbf{x}'\mathbf{x}\right)^{\frac{\nu+d}{2}}},$$

with one factor in this product distribution for each data point. Here Γ represents the gamma function.

Note that for $\rho, \nu \to \infty, \frac{\rho}{\nu} \to \sigma^2$ this tends to the multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix $\sigma^2\mathbf{I}$. For $\rho = \nu = 1$ this is a multivariate standard Cauchy distribution, which is so heavy-tailed that its mean is undefined and its second moment is infinitely large. Thus, this type of prior beliefs can clearly model the expectation of outliers to varying degrees.

4.2 The subjective interestingness of a projection pattern

To compute the subjective information content, note that the density function for the transformed variable $\mathbf{z} = \mathbf{W}'\mathbf{x}$ with $\mathbf{W}$ an orthogonal matrix is given as:

$$p(\mathbf{z}) \quad = \quad \frac{\Gamma\left(\frac{\nu+d}{2}\right)}{\sqrt{(\pi\rho)^d}\Gamma\left(\frac{\nu}{2}\right)} \cdot \frac{1}{\left(1+\frac{1}{\rho}\mathbf{z}'\mathbf{z}\right)^{\frac{\nu+d}{2}}}.$$

Now, the density function for the marginal distribution of a t-distribution with given covariance matrix is again a t-distribution density with the same number of degrees of freedom, obtained by simply selecting the relevant part of the covariance matrix [13, 15]. With $\mathbf{w}$ denoting the first column of $\mathbf{W}$, this means that the density function for $z_1 = \mathbf{w}'\mathbf{x}$, the first component of $\mathbf{z}$, is:

$$p(z_1) \quad = \quad \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\rho}\Gamma\left(\frac{\nu}{2}\right)} \cdot \frac{1}{\left(1+\frac{1}{\rho}z_1^2\right)^{\frac{\nu+1}{2}}}.$$

Written in terms of $\mathbf{x}$, this is:

$$p(\mathbf{x}'\mathbf{w}) \quad = \quad \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\rho}\Gamma\left(\frac{\nu}{2}\right)} \cdot \frac{1}{\left(1+\frac{1}{\rho}\mathbf{w}'\mathbf{x}\mathbf{x}'\mathbf{w}\right)^{\frac{\nu+1}{2}}}.$$

Thus, the subjective information content of a pattern stating that $\mathbf{X}\mathbf{w} = \mathbf{p}$ is:

$$\begin{aligned}&\text{SubjectiveInformationContent}\,(\mathbf{X}\mathbf{w} = \mathbf{p})\\ =\;& \frac{\nu+1}{2}\sum_{i=1}^n \log\left(1+\frac{1}{\rho}(\mathbf{x}_i'\mathbf{w})^2\right) + \text{a constant.}\end{aligned}$$

Again, as the description length is constant, this is proportional to the subjective interestingness.

4.3 Maximising the interestingness using a robust version of PCA

Taking into account that $\mathbf{w}'\mathbf{w} = 1$ (as required in the patterns considered and as imposed by the orthogonality of $\mathbf{W}$), maximising the subjective interestingness is thus equivalent to solving the following problem:

$$\begin{aligned}\max_{\mathbf{w}} \quad & \sum_{i=1}^n \log\left(\rho + (\mathbf{x}_i'\mathbf{w})^2\right),\\ \text{s.t.} \quad & \mathbf{w}'\mathbf{w} = 1.\end{aligned}$$

The method of Lagrange multipliers leads to the following optimality condition for the subjective information content:

$$\left(\sum_{i=1}^n \frac{\mathbf{x}_i\mathbf{x}_i'}{\rho + (\mathbf{x}_i'\mathbf{w})^2}\right)\mathbf{w} \quad = \quad \lambda\mathbf{w}.$$

Note that the matrix on the left hand side is propotional to essentially a weighted empirical covariance matrix for the data, where points contribute more if they have a smaller value for $(\mathbf{x}_i'\mathbf{w})^2$: the weight for $\mathbf{x}_i\mathbf{x}_i'$ is $\frac{1}{\rho+(\mathbf{x}_i'\mathbf{w})^2}$.

Although this optimisation problem is not convex and the optimality conditions do not admit a closed form solution in terms of e.g. an eigenvalue problem, a modified version of the power method for solving eigenvalue problems empirically appears to be a good heuristic approach. The algorithm goes as follows:

1. Solve the eigenvalue problem $\left(\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i'\right)\mathbf{w} = \lambda\mathbf{w}$ for the dominant eigenvector,[1] further denoted $\mathbf{w}^{(0)}$. This vector is normalised to unit norm.

[1] This amounts to solving the problem for $\rho \to \infty$, which is essentially equivalent to PCA. This is no coincidence as for $\rho, \nu \to \infty, \frac{\rho}{\nu} \to \sigma^2$ the background distribution is an isotropic multivariate Gaussian distribution, as noted above.

2. Iterate from $k = 1$ until convergence or maximum number of iterations reached:
   (a) $\mathbf{v}^{(k)} = \left(\sum_{i=1}^{n} \frac{\mathbf{x}_i\mathbf{x}_i'}{\rho+(\mathbf{x}_i'\mathbf{w}^{(k-1)})^2}\right)\mathbf{w}^{(k-1)}$.
   (b) $\mathbf{w}^{(k)} = \frac{\mathbf{v}^{(k)}}{\|\mathbf{v}^{(k)}\|}$.

Clearly this is not guaranteed to converge to the global optimum, but in practice it appears to perform well. Whether it always converges to a local optimum is left as an open question in this note.

The effect of the parameter ρ is as follows. For a smaller value of ρ, the tail of the background distribution can be heavier, as then the nonlinearity of the logarithm in the prior belief constraint will affect data points of smaller magnitude. The effect of this is that outliers (for which $(\mathbf{x}_i'\mathbf{w})^2$ may be very large) will not weigh in as strongly as they would in PCA, as the contribution of $\mathbf{x}_i\mathbf{x}_i'$ to what can be thought of as a reweighted covariance matrix is reduced, and relatively more so than for data points for which $(\mathbf{x}_i'\mathbf{w})^2$ is small as compared to ρ (for which the reduction is roughly constant). Informally speaking, ρ is a soft threshold on the squared distance along $\mathbf{w}$ beyond which data points will no longer be able to bias the solution in their own direction.

Interestingly, just like in PCA where the value of σ has no effect on which pattern is most interesting, here the value of ν and thus of c has no effect on which projection is the most interesting one. (Though σ and c do affect the value of the interestingness in both cases.) This significantly reduces the demands on the user in specifying their prior beliefs.

4.4 Subsequent iterations

A property of the multivariate t-distribution is that the conditional distribution conditioned on the value of any of the dimensions is again a multivariate t-distribution, though with a different number of degrees of freedom and a different covariance matrix [15]. Thus, after revealing the values of the projections $\mathbf{p}$, the updated background distribution is again a multivariate t-distribution for the parts of the data points orthogonal to $\mathbf{w}$ from the first pattern. The next pattern can be found essentially by projecting the data points onto the orthogonal complement of $\mathbf{w}$ and repeating the same procedure.

5. EXPERIMENT

To illustrate the robustness of the PCA alternative derived in the previous section, consider a dataset consisting of 1000 data points sampled from a Gaussian distribution with mean $\mathbf{0}$ and with covariance matrix $\begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$, to which a further 100 'outliers' are added, sampled from a Gaussian distribution with mean $\mathbf{0}$ and with covariance matrix $\begin{pmatrix} 16 & 12 \\ 12 & 13 \end{pmatrix}$.

The weight vector resulting from standard PCA is shown with a full red line in Fig. 1. The black dash-dotted lines show the weight vectors retrieved by the robust PCA method described, with values for ρ equal to $1, 10$, and 100. The largest value of these resulted in the line closest to the PCA result. The green dashed line shows the weight vector that would have been found using standard PCA had there been no outliers at all (i.e. computed just on the first 1000 data points).

The left figure shows the resulting weight vectors on top of a scatter plot of all data points, clearly showing that the PCA result is determined primarily by the outliers. The right figure shows the same resulting weight vectors on top of a scatter plot of only the first 1000 data points (excluding the outliers). Clearly, the robust PCA version is much less strongly affected by the outliers and primarily determined by the dominant variance direction in the bulk of the data points excluding the outliers.

6. DISCUSSION AND FURTHER WORK

This note shows how PCA can be derived as an instantiation of the framework from [2] for deriving subjective interestingness of exploratory data mining patterns. Additionally, it shows how prior beliefs reflecting the expectation that outliers may be present in the data lead to an alternative to PCA that is less sensitive to such outliers.

Robust PCA is an important research topic that has been studied for decades, see e.g. [1, 14, 6, 21] for a few recent references. Often the problem is tackled as an instance of projection pursuit (and also our algorithm could be viewed as such) [4, 5], by making use of a robust estimator of the covariance matrix [17, 16], or by making additional assumptions about the nature of the interesting aspects of the data and the corrupting noise process. The algorithm derived in this note appears to be most strongly related to the algorithm from [14], but further study into connections between the two is required.

In further work we will enhance the rigour of the derivations, attempt to establish the convergence of the algorithm for the robust version of PCA, and investigate the utility of other alternatives to PCA that are useful for other relevant kinds of prior belief states. E.g. it is relatively straightforward to add assumptions on anisotropy of the data to the prior beliefs in both the derivation of PCA and of the robust version of PCA, as well as assumptions about the expected average of the data points not being the origin. However also altogether different kinds of prior beliefs could be of interest.

Acknowledgements

This work is supported by the ERC Consolidator Grant FORSIED.

7. REFERENCES

[1] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[2] T. De Bie. An information-theoretic framework for data mining. In *Proc. of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2011.

[3] T. De Bie. Maximum entropy models and subjective interestingness: an application to tiles in binary databases. *Data Mining and Knowledge Discovery*, 23(3):407–446, 2011.

[4] Jerome H Friedman and John W Tukey. A projection pursuit algorithm for exploratory data analysis. 1973.

[5] Peter J Huber. Projection pursuit. *The annals of Statistics*, pages 435–475, 1985.

[6] Mia Hubert, Peter J Rousseeuw, and Karlien Vanden Branden. ROBPCA: a new approach to robust principal component analysis. *Technometrics*, 47(1):64–79, 2005.

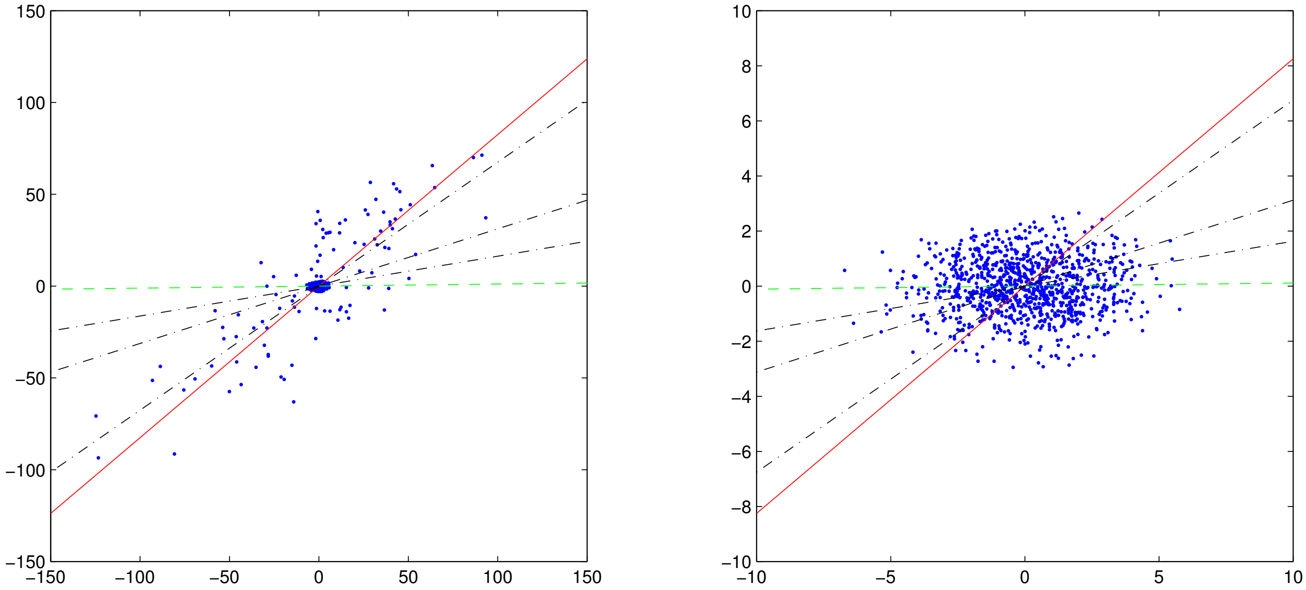


Figure 1: The left plot shows a scatter plot of all data points including outliers, with weight vectors of standard PCA (continuous red line), as well as the robust PCA with values for $\rho = 1, 10, 100$ (black dash-dotted line) and standard PCA on the data points excluding the 100 outliers (green dashed line). The right plots shows the same results but now without visualising the outliers in the scatter plot.

[7] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[8] K.-N. Kontonasios and T. De Bie. An information-theoretic approach to finding informative noisy tiles in binary databases. In *Proc. of the 2010 SIAM International Conference on Data Mining (SDM)*, 2010.

[9] K.-N. Kontonasios and T. De Bie. Formalizing complex prior information to quantify subjective interestingness of frequent pattern sets. In *Proc. of the 11th International Symposium on Intelligent Data Analysis (IDA)*, 2012.

[10] K.-N. Kontonasios and T. De Bie. Subjectively interesting alternative clusterings. *Machine Learning*, 2013.

[11] K.-N. Kontonasios, J. Vreeken, and T. De Bie. Maximum entropy modelling for assessing results on real-valued data. In *Proc. of the IEEE International Conference on Data Mining (ICDM)*, 2011.

[12] K.-N. Kontonasios, J. Vreeken, and T. De Bie. Maximum entropy models for iteratively identifying subjectively interesting structure in real-valued data. In *Proc. of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery from Databases (ECML-PKDD)*, 2013.

[13] S. Kotz and S. Nadarajah. *Multivariate t distributions and their applications*. Cambridge University Press, 2004.

[14] Yongmin Li, L-Q Xu, Jason Morphett, and Richard Jacobs. An integrated algorithm of incremental and robust pca. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I–245. IEEE, 2003.

[15] Michael Roth. On the multivariate t distribution. Technical Report LiTH-ISY-R-3059, Department of Electrical Engineering, Linköping universitet, April 2013.

[16] Peter J Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

[17] Peter J Rousseeuw and Bert C Van Zomeren. Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, 85(411):633–639, 1990.

[18] E. Spyropoulou and T. De Bie. Interesting multi-relational patterns. In *Proc. of the IEEE International Conference on Data Mining (ICDM)*, 2011.

[19] E. Spyropoulou, T. De Bie, and M. Boley. Interesting pattern mining in multi-relational data. *Data Mining and Knowledge Discovery*, 2013.

[20] E. Spyropoulou, T. De Bie, and M. Boley. Mining interesting patterns in multi-relational data with n-ary relationships. In *Discovery Science (DS)*, 2013.

[21] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.

[22] K. Zografos. On maximum entropy characterization of pearson's type II and VII multivariate distributions. *Journal of Multivariate Analysis*, 71(1):67–75, 1999.

Toward Usable Interactive Analytics: Coupling Cognition and Computation

Alex Endert
Pacific NW National Lab
Richland, WA USA
alex.endert@pnnl.gov

Chris North
Virginia Tech
Blacksburg, VA USA
north@cs.vt.edu

Remco Chang
Tufts University
Medford, MA USA
remco@cs.tufts.edu

Michelle Zhou
IBM Research
Almaden, CA USA
mzhou@us.ibm.com

ABSTRACT

Interactive analytics provide users a myriad of computational means to aid in extracting meaningful information from large and complex datasets. Much prior work focuses either on advancing the capabilities of machine-centric approaches by the data mining and machine learning communities, or human-driven methods by the visualization and CHI communities. However, these methods do not yet support a true human-machine symbiotic relationship where users and machines work together collaboratively and adapt to each other to advance an interactive analytic process. In this paper we discuss some of the inherent issues, outlining what we believe are the steps toward usable interactive analytics that will ultimately increase the effectiveness for both humans and computers to produce insights.

1. INTRODUCTION

To tackle the onset of big data, visual analytics seeks to marry the human-intuition of visualization with the analytical horsepower of mathematical models. Yet, a critical open question is how humans will interact with, steer, and train these complex mathematical models.

The visual analytics community has worked to provide visual representations of data, as approximated by complex models and analytics [34]. User interaction is critical to the success of such visual data exploration, as it allows users to engage in a process of testing assertions, assumptions, and hypotheses about the information given one's prior knowledge about the world. This cognitive process can be generally referred to as sensemaking. Visual analytics emphasizes sensemaking of large, complex datasets through interactively exploring visualizations generated via a combination of analytic models. Thus, a central focus is understanding how to leverage human cognition in concert with powerful computation through usable visual metaphors.

Initially, the principles of direct manipulation were applied to such models in a simplistic fashion by using control panels to directly manipulate model parameters. Direct manipulation specifies the following three properties for interaction design for information visualization: (1) continuous representation of the object of interest, (2) physical actions or labeled button presses instead of complex syntax, and (3) rapid incremental reversible operations whose impact on the object of interest is immediately visible [31]. Typically, these principles are applied in the form of a control panel, containing visual widgets such as sliders, buttons, or query fields, coupled to the parameters of a visual representation in the main view. For the purpose of interactive machine learning, these interfaces provide feedback in an expressive and formal way (e.g., standard training and labeling tasks).

However, for users and their analytic tasks, these interactions may present significant usability issues by forcing the user out of their cognitive flow or zone [11,22], and may place fundamental limitations on sensemaking activity due to lack of recognition of the depth of interactions which humans apply in their cognitive processes. Exploiting humans merely as data labelers or parameter tuners mis-uses human expertise and skills, forcing humans to adapt to formal algorithmic methods and apriori parameter specifications, when their strengths are in incremental informal reasoning. More importantly, it misses a major opportunity for the potential benefits of coupling cognition and computation.

We contend that a new methodology to couple the cognitive and computational components of such systems is needed. We suggest *Semantic Interaction* as a potential solution concept, which attempts to bridge these components by binding the user interactions used for visual sensemaking with the training of machine learning techniques [17]. Semantic interaction interfaces produce this coupling by leveraging the visual metaphor as the mapping function, and the visual encoding as the interactive affordance by which users perform their visual data exploration. In this paper we discuss the concept of semantic interaction as a method for systematically learning characteristics about a user and his or her reasoning process, adapting the underlying analytic model, and increasing the usability of incorporating the human in the loop.

2. SEMANTIC INTERACTION

Semantic interaction is an approach to user interaction for visual analytics in which the user's analytical reasoning is inferred and in turn used to steer the underlying models implicitly. The goal of this approach is to enable co-reasoning between the human and the analytic models (coupling cognition and computation) without requiring the user to directly control the models and parameters. This co-reasoning occurs through mutual interaction with a visual medium of communication – the visualization or visual metaphor.

The approach of semantic interaction is to overload the metaphor through which the insights are obtained (i.e., the visualization of information created by computational models) and the interaction metaphor through which hypotheses and assertions are communicated (i.e., interaction occurs within the visual metaphor). Semantic interaction enables users to directly manipulate data within visualizations, from which tacit knowledge about the user is captured, and the underlying analytic models are steered. The analytic models can be incrementally adapted based on the user's incremental sensemaking process and domain expertise explicated via the user interactions with the system. The specifics of the system could include multiple visual metaphors used in concert.

That is, the parameters of the underlying analytic models are exposed through the visual constructs of the visualization. Based on common visual metaphors (such as the geographic, spatial metaphor where proximity approximates similarity), tacit knowledge of the user's reasoning can be inferred through inverting these analytic models. As a result, users are shielded from the underlying complexities, and able to interact with their data through a bi-directional visual medium. The interactions users perform within the visualizations to augment the visual encodings within the metaphor enable the inference of their analytic reasoning, which are systematically applied to the underlying models. The visual metaphor helps define the mapping between the model parameters and the visualization, and the visual encoding provides the visual interactive affordance by which users can interact. Thus, the process of visual data exploration and models steering occur on the same set and sequence of interactions.

The semantic interaction pipeline (shown in Figure 1) takes an approach of directly binding model steering techniques to the interactive affordances created by the visualization. For example, a distance function used to determine the relative similarity between two data points (visually depicted as distance in a spatial layout), can serve as the interactive affordance to allow users to explore that relationship. Therefore, the user interaction is directly in the visual metaphor, creating a bi-directional medium between the user and the analytic models. This method of user interaction is also similar to the "by example" method of interaction, as users can directly show their intention using the structure of the visualization. This adds to the role of visualization in the reasoning process, in that it is no longer intended to be solely a method for gaining insight, but also one for directly interacting with the information and the system. The bi-directionality afforded by semantic interaction comes via binding the parameter controls traditionally afforded by the GUI directly within the visual metaphor. It is through this binding that an inference can be made about the user's analytic reasoning from the user interaction with the visualization with regards to the parameters of the underlying mathematical model.

For example, a spatial layout is one specific visual metaphor where existing research on semantic interaction has been conducted, described in [14,15,16]. The spatial visual metaphor (i.e., a spatialization) is one where the bi-directionality afforded by semantic interaction has been demonstrated. A spatial metaphor lends itself well to common dimension reduction models to reduce the dimensionality of complex data to two dimensions. For example, relationships and similarities between high-dimensional data objects can be shown in two dimensions by leveraging dimension reduction models including: principal-component analysis, multi-dimensional scaling, force-directed layouts, etc. In general, these models attempt to approximate the distance between data objects in their true, high-dimensional representation using a smaller number of dimensions (e.g. two dimensions in the case of spatial visualization).

Prior work has applied semantic interaction methods to this visual metaphor. For example, inverting multi-dimensional scaling, principal-component analysis, and generative topographic mapping can enable bi-directional spatializations to afford semantic interaction [4,16]. The ability to understand the parameters of each of the models that can be exposed through the visual encoding (in this case, relative distance between data points) enabled this affordance. Further work has explored the tradeoffs between the various ways to map the user feedback of changing the relative distance between data objects to the underlying dimension reduction models [24,27].

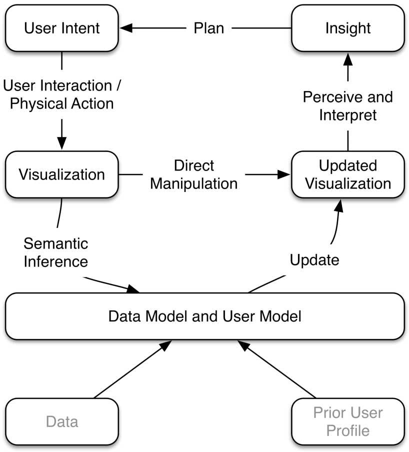


Figure 1 A generalizable model for coupling cognition and computation. Plans generate intents that are externalized by users via interactions and physical actions. Data and user models can be inferred from these actions, and used to update a visualization to continue the analytic process.

3. RESEARCH AGENDA

Based on the promising initial results of current research on semantic interaction for visual analytics, the sections below describe open areas of research to advance the field in usable interactive analytics. These sections describe current work in each topic, as well as illuminate open areas of research that can advance the goal of creating usable interactive analytics via semantic interaction. The areas of research can be depicted in a generalizable model for semantic interaction interfaces, shown in Figure 1.

3.1 Sensing and Capturing User Interaction

Semantic interaction interfaces are grounded in the concept of treating user interaction as data from which models about the user are created. This interaction data about the user can be captured from two categories of sources: virtual interactions, and physical actions.

Virtual interactions refer to those that a user performs within a user interface. These have been previously studied for the purposes of understanding the user. For example, Yi et al. presented an extensive categorization of user interactions available in popular exploratory visualization tools [35]. Further, Dou et al. have shown that through logging user interactions in a visualization of financial data, low-level analytical processes can be reconstructed [9,26]. Most importantly, these results indicate that a detectable connection exists between the low-level user interaction and the high-level analytic processes of users when it comes to visual data exploration. The advancement of understanding how processes and knowledge from users manifest in user interaction forms the *science of interaction* [29].

The physical actions or attributes that humans exhibit while analyzing data may also provide cues from which models can be generated and adapted. For example, research has shown that navigating large information spaces using physical navigation with large displays is significantly advantageous over virtual navigation with small displays [2]. These physical actions, or strategies for interacting, can also be analyzed to identify effectiveness of analytic strategies on such displays [12]. For example, the sensing of office chair rotation relative to a large display can provide an approximation of the user's primary focus of attention [13]. These, as well as other physiological measures, such as EEG, fNIRS, and fMRI, can increase the amount of information about a user that can be modeled, and ultimately re-cast into interactions with mixed-initiative analytics systems [1,28,32].

Open questions within this topic include:

- What additional visual metaphors and user interfaces can be sources of user interaction data to add breadth to the science of interaction?
- How can the directness of the virtual interactions (with respect to the interface and task) be coupled with the passiveness of the physical actions? What are the tradeoffs between the passive sensing of physical actions and the direct sensing of virtual interactions?

3.2 Inferring User Models

As visualization systems become more complex, so do the user's ability to express their reasoning process through these complex interfaces. These reasoning processes reflect a user's cognitive abilities [7] and personality traits [36], and are often influenced by the user's cognitive and mental state (such as emotion and cognitive load) [23,28].

The research goal of User Modeling is to reconstruct the relevant profile of a user by analyzing their interactions with a complex visualization tool. For example, Brown et al. demonstrated that a user's performance during a visual search task, as well as aspects of a user's personality profile, can be inferred and predicted in real-time [5]. Similarly, the physical motions of a user's mouse movement have been shown to be effective as biometrics to authentic a user's identity for security purposes [30].

Beyond analyzing a user's virtual interactions (mouse and keyboard interactions), other user-generated data has also been used to infer models of a user. For example, eye-tracking data has been shown to reflect a user's cognitive abilities and personality traits [33]. More broadly, Gou et al. developed a tool called *System U* that can automatically identify a user's full personality profile by examining as little as two hundred of the user's Twitter postings [21]. These user modeling techniques give rise to the possibility of mixed-initiative visual analytics systems in which the computer can understand and support the user's analysis needs in real time [34].

Open research areas include:

- What other forms of models can be inferred, steered, and created (e.g., task models, role-based models, etc.)?
- How can we detect artifacts of cognitive processes that may be less desired (e.g., forms of bias, cognitive depletion, etc.)?

3.3 Inferring Data Models

Semantic interaction interfaces can implicitly map to, train, and steer underlying data models. One method to do this is to enable users to manipulate the output of the model, and then computationally invert the model to learn optimized inputs that would produce the desired outputs.

For example, a data model might consist of a weighting of data features applied in a weighted dimensionality reduction algorithm. Instead of requiring users to directly manipulate the input weighting of features, semantic interaction enables users to manipulate the output visualization of the information, from which the weighting of features can be inferred. Prior work has shown how such user interactions (e.g., re-organizing data within a spatial layout) can map to the weighting of features, in tools such as OLI [4,16,27] and Dis-Function [4]. Term weights for text analytcs can be learned from users interactions with spatial organizations of documents, highlighting, annotations, reading patterns, eye gaze, etc. in ForceSPIRE [14,15] and StarSpire [3]. iCluster demonstates learning of a document clustering model through users' incremental cluster membership choices [10]. Apolo demonstrates learning network belief-propagation models through textual sensemaking interactions [6].

Open questions include:

- What additional data processing models can be steered or created?
- How can we consider models that function on different scales of data (i.e., from overview to detail, but also from detail to meaningful context)?

3.4 Adaptive Visualization

Techniques for User and Data Modeling would inform the visualization and the analytics system's high-level information about the user's analysis goals and needs. Responding to these inputs, the visualization system can adapt the information and representation presented to the user. Similarly, the analytics engine can also modify its behavior to achieve more efficient and accurate analysis results (see Section 3.5).

Adaptive user interfaces and visualization systems have been an important research topic in HCI. Interfaces such as SUPPLE have demonstrated that a system can learn a user's motor disabilities (such as Parkinson's) or the limitations of the device (such as smart phone and tablet), and automatically adapt the size and positioning of UI elements to generate a user interface that is optimal to the user and the device [20].

In adaptive visualization, researchers have examined the relationship between visual metaphors and the user's personality traits [23,33,36]. Moving beyond interface-level adaptations, systems have also adapted based on the amount of information presented to the user. In the context of games and training, these types of adaptations are often referred to as "dynamic difficulty" adjustments [25], but the same techniques have been more broadly applied to real-world scenarios such as assisting operators of unmanned vehicles and robots [1,8,32].

Open questions include:

- How do we ensure that, in mixed-initiative systems, both the system and the user have equal opportunities to provide feedback?

- How do we ensure the system responds in a way that amplifies the cognitive processes, and aids them, instead of deteriorating the performance of the person?

3.5 Adaptive Computation

In addition to user-level adaptation, analytics algorithms and systems can also benefit from having knowledge about the user's cognitive style and analysis processes.

As datasets get larger, it becomes increasingly difficult for visualizations and analytic systems to provide both interactivity and complete data analysis simultaneously. The old design adage of "Overview First, Details on Demand" limits the size of the data that a visual analytics system can support at an interactive rate. The "Big Data" challenge requires new computational techniques and paradigm shifts. In the case of visual analytics, one potentially rich and fruitful approach is to integrate User and Data Modeling into novel adaptive computational techniques.

"Approximate computing" can generate an overview of a large dataset in real-time. Approximate computing will, by definition, be less accurate than traditional statistical or machine learning techniques, but will deliver sufficient information for the user to perceive high-level patterns within the data in a fraction of the time. Some plausible factors for approximate computing include the consideration of human perception properties such as just noticeable difference (JND), or cognitive limitations based on attention, working memory capacity, or cognitive load [18].

In addition, "user-guided computation" that leverages knowledge of the user's analysis process and goals can lead to advancements in efficient, online algorithms that compute only the information needed by the user. As the user explores the data, these algorithms can incrementally increase (or decrease) in detail by incorporating more (or less) data. Such an analytic engine can maintain a small memory footprint while providing the user with rich information throughout the user's exploration process [19].

Open questions include:

- How do we perform model selection over a set of models has been created, selecting the one (or combination) that is most appropriate given the context of the analysis?

- What are other forms of models or computation that lend themselves to the semantic interaction methods outlined in this position statement?

4. CONCLUSION

Achieving effective coupling of cognition and computation for interactive analytics will require significant research attention towards usability and interaction issues. Clearly, we must go well beyond existing simple human-in-the-loop methods. We have outlined a research agenda that we believe will be critical to enabling insight in the big data era.

ACKNOWLEDGEMENTS

PNNL is managed for the US Department of Energy by Battelle under Contract DE-AC05-76RL01830.

REFERENCES

1. Afergan, D., Peck, E.M., Solovey, E.T., et al. Dynamic Difficulty Using Brain Metrics of Workload. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2014), 3797–3806.
2. Ball, R., North, C., and A. Bowman, D. Move to improve: promoting physical navigation to increase user performance with large displays. *ACM CHI*, ACM (2007).
3. Bradel, L. and North, C. StarSpire: Multi-scale Semantic Interaction. *IEEE VAST*, (2014).
4. Brown, E.T., Liu, J., Brodley, C.E., and Chang, R. Dis-function: Learning Distance Functions Interactively. *IEEE VAST*, (2012).
5. Brown, E.T., Ottley, A., Zhao, J., et al. Finding Waldo: Learning about Users from their Interactions. *Trans. Vis. Comput. Graph*, (2014).
6. Chau, D.H., Kittur, A., Hong, J.I., and Faloutsos, C. Apolo: Making Sense of Large Network Data by Combining Rich User Interaction and Machine Learning. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2011), 167–176.
7. Chen, C. Individual differences in a spatial-semantic virtual environment. *Journal of the American Society for Information Science 51*, 6 (2000), 529–542.
8. Donmez, B., Nehme, C., and Cummings, M.L. Modeling Workload Impact in Multiple Unmanned Vehicle Supervisory Control. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 40*, 6 (2010), 1180–1190.
9. Dou, W., Jeong, D.H., Stukes, F., Ribarsky, W., Lipford, H.R., and Chang, R. Recovering Reasoning Processes from User Interactions. *IEEE Computer Graphics and Applications 29*, (2009), 52–61.
10. Drucker, S.M., Fisher, D., and Basu, S. Helping users sort faster with adaptive machine learning recommendations. Springer-Verlag (2011), 187–203.
11. Elmqvist, N., Moere, A.V., Jetter, H.-C., Cernea, D., Reiterer, H., and Jankun-Kelly, T. Fluid interaction for information visualization. *Information Visualization 10*, (2011), 327–340.
12. Endert, A., Andrews, C., and North, C. Visual Encodings that Support Physical Navigation on Large Displays. *Graphics Interface*, (2011).
13. Endert, A., Fiaux, P., Chung, H., Stewart, M., Andrews, C., and North, C. ChairMouse: leveraging natural chair rotation for cursor navigation on large, high-resolution displays. ACM (2011), 571–580.
14. Endert, A., Fiaux, P., and North, C. Semantic Interaction for Visual Text Analytics. *ACM Human Factors in Computing (CHI)*, (2012).
15. Endert, A., Fiaux, P., and North, C. Semantic Interaction for Sensemaking: Inferring Analytical Reasoning for Model Steering. *Visualization and Computer Graphics, IEEE Transactions on 18*, (2012), 2879–2888.
16. Endert, A., Han, C., Maiti, D., House, L., Leman, S.C., and North, C. Observation-level Interaction with Statistical Models for Visual Analytics. *IEEE VAST*, (2011), 121–130.
17. Endert, A. Semantic Interaction for Visual Analytics: Inferring Analytical Reasoning for Model Steering. 2012. http://scholar.lib.vt.edu/theses/available/etd-07112012-123927/.
18. Fisher, D., Drucker, S.M., and König, A.C. Exploratory Visualization Involving Incremental, Approximate Database Queries and Uncertainty. *IEEE Computer Graphics and Applications 32*, 4 (2012), 55–62.
19. Fisher, D., Popov, I., Drucker, S., and schraefel, m. c. Trust Me, I'M Partially Right: Incremental Visualization Lets Analysts Explore Large Datasets Faster. *Proceedings of the*

SIGCHI Conference on Human Factors in Computing Systems, ACM (2012), 1673–1682.

20. Gajos, K.Z., Wobbrock, J.O., and Weld, D.S. Improving the Performance of Motor-impaired Users with Automatically-generated, Ability-based Interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2008), 1257–1266.
21. Gou, L., Zhou, M.X., and Yang, H. KnowMe and ShareMe: Understanding Automatically Discovered Personality Traits from Social Media and User Sharing Preferences. *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, ACM (2014), 955–964.
22. Green, T.M., Ribarsky, W., and Fisher, B. Building and applying a human cognition model for visual analytics. *Information Visualization 8*, (2009), 1–13.
23. Harrison, L., Skau, D., Franconeri, S., Lu, A., and Chang, R. Influencing Visual Judgment Through Affective Priming. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2013), 2949–2958.
24. Hu, X., Bradel, L., Maiti, D., House, L., North, C., and Leman, S. Semantics of Directly Manipulating Spatializations. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2052–2059.
25. Hunicke, R. and Chapman, V. AI for dynamic difficulty adjustment in games. .
26. Jeong, D.H., Dou, W., Lipford, H.R., Stukes, F., Chang, R., and Ribarsky, W. Evaluating the relationship between user interaction and financial visual analysis. *IEEE Symposium on Visual Analytics Science and Technology, 2008. VAST '08*, (2008), 83–90.
27. Leman, S.C., House, L., Maiti, D., Endert, A., and North, C. A Bi-directional Visualization Pipeline that Enables Visual to Parametric Interaction (V2PI). *PLOS One*, (2011).
28. Peck, E.M.M., Yuksel, B.F., Ottley, A., Jacob, R.J.K., and Chang, R. Using fNIRS Brain Sensing to Evaluate Information Visualization Interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2013), 473–482.
29. Pike, W.A., Stasko, J., Chang, R., and O'Connell, T.A. The science of interaction. *Information Visualization 8*, (2009), 263–274.
30. Pusara, M. and Brodley, C.E. User Re-authentication via Mouse Movements. *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, ACM (2004), 1–8.
31. Shneiderman, B. Direct Manipulation: A Step Beyond Programming Languages. *Computer 16*, 8 (1983), 57–69.
32. Solovey, E., Schermerhorn, P., Scheutz, M., Sassaroli, A., Fantini, S., and Jacob, R. Brainput: Enhancing Interactive Systems with Streaming Fnirs Brain Input. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2012), 2193–2202.
33. Steichen, B., Carenini, G., and Conati, C. User-adaptive Information Visualization: Using Eye Gaze Data to Infer Visualization Tasks and User Cognitive Abilities. *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, ACM (2013), 317–328.
34. Thomas, J.J. and Cook, K.A. Illuminating the path. 2005. http://nvac.pnl.gov/agenda.stm#book.
35. Yi, J.S., Kang, Y. ah, Stasko, J., and Jacko, J. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics 13*, (2007), 1224–1231.
36. Ziemkiewicz, C., Ottley, A., Crouser, R.J., et al. How Visualization Layout Relates to Locus of Control and Other Personality Factors. *IEEE Transactions on Visualization and Computer Graphics 19*, 7 (2013), 1109–1121.

Rapid Data Exploration and Visual Data Mining on Relational Data

Gartheeban Ganeshapillai, Joel Brooks, John Guttag
Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology
32 Vassar Street
Cambridge, MA 02139
{garthee, brooksjd, guttag}@mit.edu

ABSTRACT

Exploring and analyzing a large amount of data is becoming increasingly common, and human involvement in the process is often required. The advantage of visual data mining is that it combines the flexibility, creativity, and general knowledge of a human with brute computational power. In this paper, we describe a novel system, a visual data mining framework, called GNoT, that supports interactive knowledge discovery by interconnecting state of the art tools for visualization, relational database management, and machine learning. The system essentially provides the glue connecting these kinds of components, and thus will be able to "ride the wave" of improvements in each of these areas. We demonstrate the tool's utility with a case study on a real-world application.

1. INTRODUCTION

Data mining is the process of extracting useful information from large data sets. Historically, research on data mining has emphasized some combination of machine learning, statistics, and database systems. In recent years, however, data visualization has come to play an ever more important role as the field of visual data mining has grown. Studies suggest visual data mining can be faster and more intuitive than traditional data mining [9]. In this paper, we describe a novel system, called GNoT, that combines data visualization and data analysis tools. It supports a style of interactive discovery in which a user follows the iterative process depicted in Figure 1. This approach to data mining has been shown to be highly effective on moderately sized data sets [9], and we believe will become even more useful in the era of "big data."

GNoT is a visual data mining framework that supports this style of interactive knowledge discovery by interconnecting state of the art tools for visualization, relational database management, and machine learning. For example, the dramatic increase in the size of the data that are mined brings a renewed focus on the database management, and in our work we attempt to benefit from the latest advancements in database systems.

By using an existing relational database management system (RDBMS), GNoT simplifies the process of storing, and accessing the data. The process of filtering, projecting, and formatting data can be done efficiently using a conventional query language (SQL). Similarly, the visualization phase is well-served by the incorporation of a high-level versatile visualization library [4] and its extensions, and the data analytics phase by the incorporation of several external libraries well suited for this purpose. The system essentially provides the glue connecting these kinds of components, and should therefore be able to "ride the wave" of improvements in each of these areas.

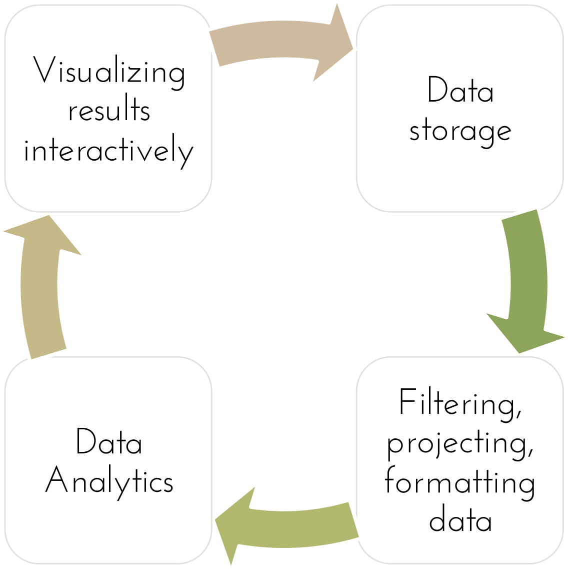

Visualizing results interactively
Data storage
Filtering, projecting, formatting data
Data Analytics

Figure 1: Pipeline of visual data exploration.

GNoT is written in Python (backend) and JavaScript (frontend). It is run as a web server, and visualizations are rendered in the clients' browsers. GNoT makes use of many JavaScript libraries that offer rich user-interactive dynamic visualizations. GNoT supports any PostgresSQL based RDBMS in the backend, and thus makes use of the recent developments on fast, parallel database systems based on a column store architecture such as Greenplum[1] and Vertica [10]. It can also interface to cloud based distributed database systems such as Redshift [15]. The platform is modular, comes with a large set of Visualization types readily available, and new types of visualization libraries can be easily added.

GNoT enables rapid visualization and visual data mining (Figure 2). With GNoT, using an existing module requires neither programming nor software engineering expertise, and extending a module typically involves integrating with a JavaScript library such as D3.

[1] http://www.gopivotal.com/

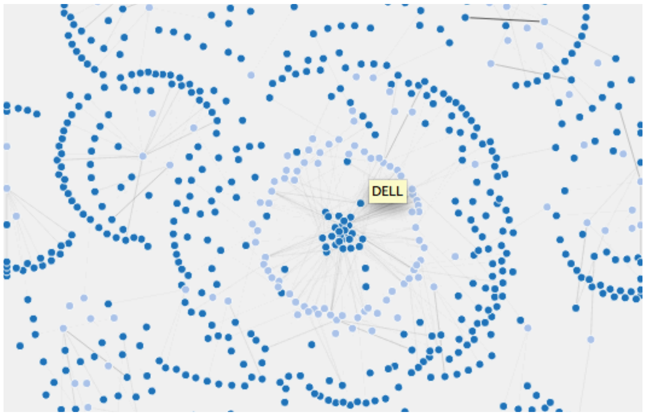


Figure 2: Using GNoT, we are able to quickly explore twitter data on financial news, and try to understand the patterns between the twitter users and the stocks covered by them. Query `|MODULE:explore_graph TABLE:finance.twitter_feed SOURCE:author_id TARGET:stock LIMIT:1000 FIELD: count(*)as n ORDERBY:n desc|` **generates the graph. First, we observe that a set of stocks are exclusively covered by a set of author_ids (twitter users). By clicking on a target node (light blue) we see the node's name (DELL). We can also observe that technology stocks form a cluster that is covered by a group of twitter users who rarely cover anything else.**

We make the following contributions in this work:

- A framework that elegantly interconnects tools from database systems, visualization tools, and machine learning libraries to offer a fluid experience in visual data mining. Our framework is modular, scalable, extensible, and makes use of state of the art tools to perform each of the subtasks.

- A fully functional implementation of the framework built using a column-store database, a set of core modules offering a large subset of the D3 visualization types, and a set of core machine learning modules. It is available at http://github.com/garthee/gnot.

- A detailed case study illustrating how the system can be used on a real problem. As part of this case study we introduce a set of modules offering analytics and visualization on geolocation data coupled with other types of data such as time series.

In the rest of the paper, we provide direct links to the demo pages (identified by ➥) whenever they are available.

The remainder of the paper is organized as follows. We first discuss related works. We then discuss the design rationale and the design of the framework. Then we go through a case study of using GNoT in a real-world data exploration application. We conclude with a short discussion and summary of the system.

2. RELATED WORK

We realize that "one size doesn't fit all". GNoT is neither an all-in-one framework nor is it optimized for a single specialized visualization task. Instead, it provides the glue connecting various specialized tools to perform rapid visualization on relational data. This enables us to easily provide the latest machine learning methods to the visual data mining process while making use of the advancements in RDBMS and visualization libraries. Whenever possible, GNoT offloads the underlying tasks to these specialized tools.

2.1 Languages and low-level tools

Many languages provide rich visualization capabilities [6]. For example, Matlab offers a large set of static plots and Matplotlib offers similar capability with Python. There are also similar tools such as Weka [8], GNU plot and ggplot that support visualization within the framework of a language. Because these tools are set within the framework of a programming language, they readily offer integration with machine learning algorithms built in those languages (e.g., Matlab, Weka and R).

There also exists a set of libraries specifically targeting visualization tasks including low-level graphics libraries such as Processing[2] and Raphael[3] [4]. While they offer great flexibility in how visualizations appear, they can be challenging to use for complex visualization tasks. Generally the visualizations generated by these tools lack dynamic controls and user interactions.

2.2 Database systems

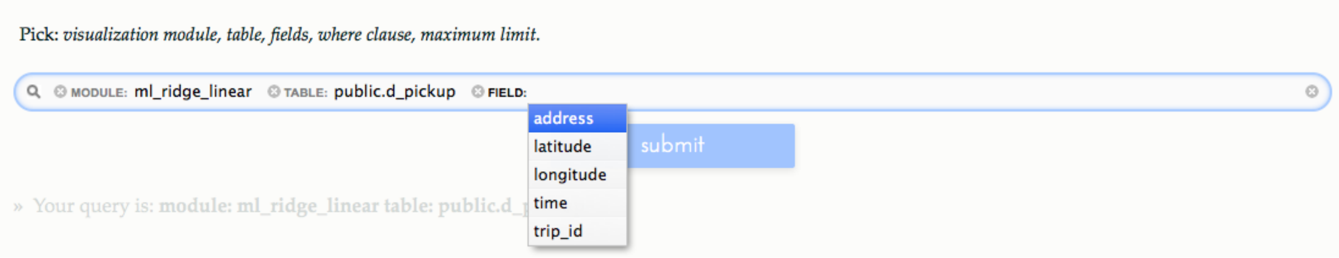
There have been significant advancements in relational database systems [14], e.g., column stores, distributed database systems, and map-reduce. Systems like Greenplum and Vertica are providing state of the art database techniques for relational data. Amazon's Redshift offers a distributed database system in the cloud [15]. However, these database systems do not offer any visualization capabilities. Further, they provide only limited support for complex analytics.

Tools such as VQE [7], Visage [12], Tioga-2 [1], and Snaptogether[11] were among the first to provide visualization environments that directly support interactive data exploration on relational data. However they offer only basic set of visualizations such as simple graphs, and are far behind contemporary graphics libraries such as D3 in keeping up with the latest advancements in visualization techniques. Further, they also do not offer any support for complex analytics. Recently, graphics libraries and visualization systems have taken their place.

2.3 Graphics Libraries

There have been many libraries developed in the last few years targeting specific graphical functionality. Flot, icharts, Exhibit, jQuery Visualize, Google Charts, and CartoDB are among those that offer specific types of in-browser visualizations. D3 stands out as a highly versatile generic browser-based visualization library [4]. It provides a close mapping between the data and the desired result, and a greater flexibility in achieving the latter. D3 also offers high-level capability by including a collection of helper modules that sit on top of the kernel library to offer rich visualizations with minimal effort. This capability has been further extended by other D3 based libraries such as Crossfilter[4], Rickshaw[5], and

[2] http://www.processing.org/
[3] http://raphaeljs.com/
[4] http://square.github.io/crossfilter/
[5] http://code.shutterstock.com/rickshaw/



Pick: *visualization module, table, fields, where clause, maximum limit.*

MODULE: ml_ridge_linear TABLE: public.d_pickup FIELD:

» Your query is: module: ml_ridge_linear table: public.d_p

Figure 3: Frontend UI: A query input box with autocomplete and hotkeys makes it easy for users to construct a query specifying table, field, and other options.

NVD3[6]. These libraries often accept the data in a delimited file (CSV or TSV) or as a JSON file.

Although GNoT is not tied to any particular front-end library, the current set of modules depend on D3 or its extensions for the front-end, and acts as middleman to feed the relational data and the results of the analytics performed on the relational data to these libraries in the format that they expect.

2.4 Visualization Systems

There are frameworks supporting the full range of the data mining pipeline. Ranging from IBM Many Eyes [16] to Improvise [17] and Polaris [13] (and its commercial implementation Tableau), they offer vertically integrated systems with a hierarchy of visualization components, tools for analytics, and data storage systems. Users of Improvise and Polaris can use existing components, subclass and extend an existing components, and add new components.

GNoT takes a similar approach, but trades tight integration for efficiency and rapid deployment. With GNoT, using an existing module requires neither programming nor software engineering expertise, and extending a module typically involves integrating a JavaScript library such as D3. Further, instead of providing a vertically integrated monolithic solution, GNoT interconnects state of the art tools for visualization, relational database management, and machine learning.

Recently, cloud based solutions for visual data mining are becoming popular, e.g., Google fusion tables, IBM's Many Eyes, Google Public Data Explorer, and Wolfram Alpha. While they differ in the offerings and the richness of the set of features, these systems typically allow the user to upload a dataset and build a set of visualizations from the data. However, they rarely offer any support to perform complex analytics on the data, and are ill suited for large datasets.

3. DESIGN RATIONALE

GNoT's primary objective is to offer rapid interactive exploration and complex analytics on large relational databases while allowing the user to benefit from the latest developments in visualization techniques, database systems, and machine learning methods for analytics.

To effectively support the stated objective, our framework must meet the following demands:

- **Ease of use:**
  In order to perform the exploration rapidly and interactively, analysts need to be able to create visualizations with relative ease. People working with data are accustomed to SQL queries and tabular data. A SQL like query with a front-end UI (Figure 3) that offers autocomplete to guide the user with input selections allows us to achieve this.

- **Easy entry and flexibility:**
  It should be easy for new users to execute simple task, while at the same time allowing the flexibility for more advance users to integrate modules needed for more advanced tasks. Instead of creating a new graphics language or protocol, we simply extend the SQL syntax. Also because the current set of modules make use of D3, they simply follow D3's protocols.

- **Technology reuse:**
  Technology reuse reduces the foot print of the framework, and allows the system to keep up to date with minimal effort. Our framework minimizes its footprint by bridging a RDBMS with front-end visualization libraries such as D3 and backend libraries such as Scikit to perform complex analytics. By reusing an ecosystem of related components, we offload a major fraction of the subtasks to specialized tools. Thus, we are able to keep up to date with the advancements in each of the subtasks.

- **Extensible and scalable:**
  In order to keep up to date, the system must be extensible. A modular approach that offloads most of the work to external libraries makes the system easily extensible. This approach allows us to replace one component by another to achieve greater functionality or scalability. For instance, in our framework a PostgresSQL based RDBMS is readily replaceable with Greenplum or Redshift. Similarly, the modular approach allows us to update the modules to use a different machine learning library with relative ease.

- **Performance:**
  Since the intent is to provide an interactive exploration tool, performance is critical. Therefore the system should have minimal overhead so that performance is governed by the speed of the individual components (e.g., the RDBMS or the machine learning component).

[6] http://nvd3.org/

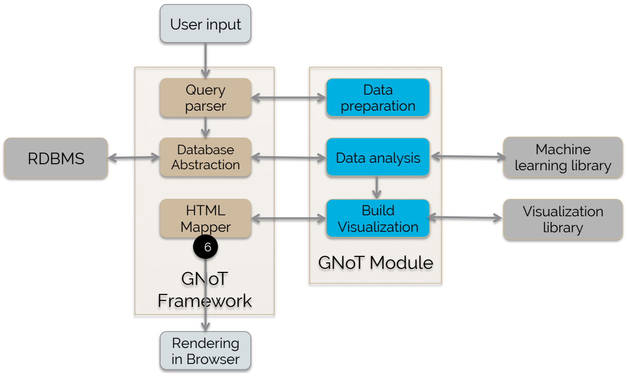


Figure 4: Architecture of GNoT

4. GNOT

Figure 4 shows the architecture of GNoT. The visualizations are implemented as a series of modules. The framework exposes the functionalities of each module to the user, and mediates interconnection among the user, the RDBMS, and the module. In addition, the framework performs various maintenance tasks such as caching, parsing and validating user inputs, and mapping HTML outputs.

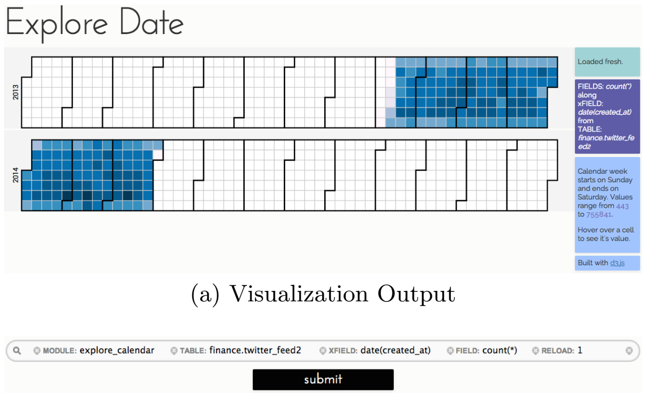
GNoT is a web server implemented on top of Werkzeug[7], and encompasses a query parser, database abstraction, and an output mapper. The database abstraction executes SQL queries generated by other modules in the system and returns the result. It also caches the query output in the file system so that queries are bypassed when the output exists in the file system. Users can overwrite the cache by opting for reload in the input query. The front end is a visual search UI built on top of Visualsearch[8]. The UI populates the options from a list of options specified by individual modules, and the table-field information from the database. The UI allows the user to quickly specify the inputs.

Visualizations are implemented by individual modules. Built-in modules cover the broad range of visualizations available in D3, and using them is as simple as selecting the visualization type in the query box. Figure 5 shows the use of such a module. Note that the input options make use of the SQL functions to format the fields.

Should users find the built-in modules insufficient, they can add additional modules. Adding a new module requires a backend file written in Python and a front end HTML/JS file to liaise with visualization libraries. The choice between these two approaches offers a tradeoff between simplicity and flexibility.

5. MACHINE LEARNING USING GNOT

A major thrust of this work is to allow users to interactively incorporate machine learning models into visual data mining tasks. GNoT has built in modules to support the three most commonly used machine learning tasks: classification (using an SVM), clustering (using k-means), and regression (using a Ridge regression). We use scikit-learn, a Python library to support the machine learning tasks in


Explore Date

Loaded fresh.

FIELDS: *count(*)* along xFIELD: *date(created_at)* from TABLE: *finance.twitter_feed2*

Calendar week starts on Sunday and ends on Saturday. Values range from 443 to 755841.

Hover over a cell to see it's value.

Built with d3.js

(a) Visualization Output

MODULE: explore_calendar TABLE: finance.twitter_feed2 XFIELD: date(created_at) FIELD: count(*) RELOAD: 1

(b) User input

Figure 5: To produce a date-based heat map (a) of the data using Explore_calendar is as easy as submitting the query `|MODULE:explore_calendar TABLE:finance.twitter_feed2 XFIELD:date(created_at )FIELD:count(*)|` **from the frontend UI (b).**

the built-in machine learning modules. However, the library can be switched for another with relative ease.

Let us walk through two of modules offering machine learning tasks: ML_SVM and ML_K-Means. We assume that the reader is familiar with using support vector machine (SVM) classifiers and the K-means clustering algorithm [5, 3].

5.1 ML_SVM

ML_SVM allows a user to apply a SVM classifier to the data. It learns a classifier separating the positives (+1) from negatives (-1) of a dependent variable using a set of features (independent variables) and validates its accuracy on a test set. ML_SVM allows user to easily construct a model, interactively tune the hyper parameters of the model, and visually analyze the fit of the model on the test set.

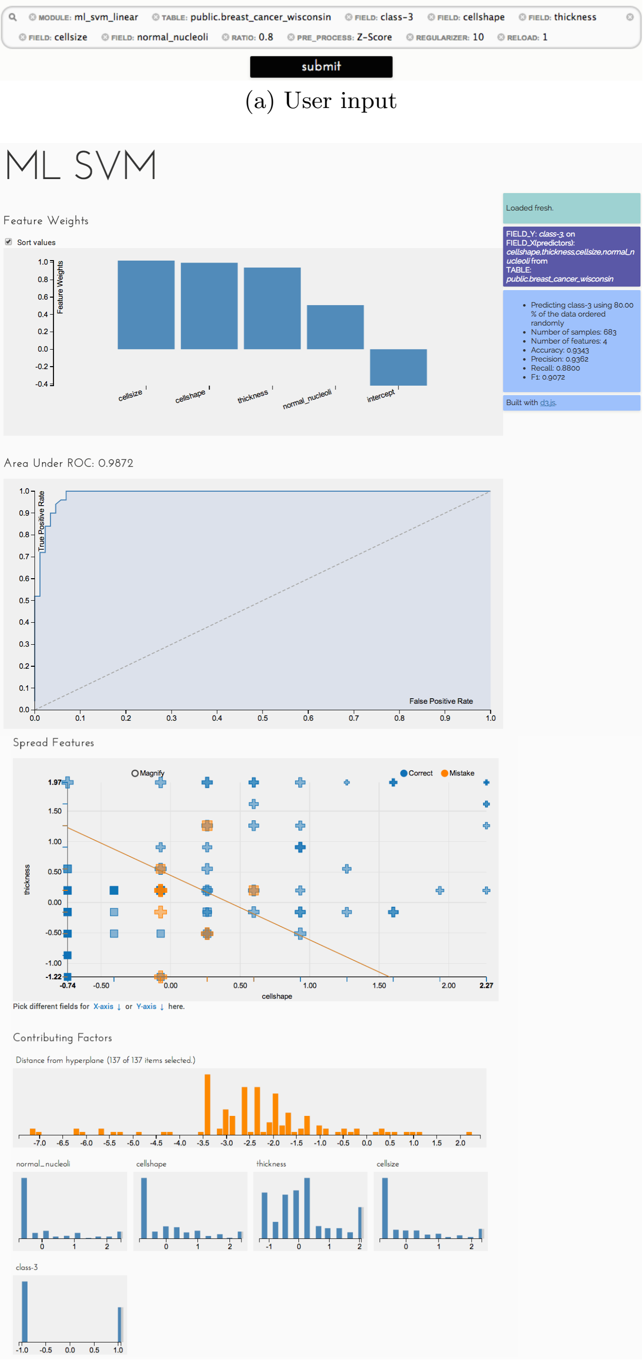
In the user's query, the first field is the dependent variable and the rest of the fields are the independent variables (i.e., the feature vector). The user also specifies the ratio of the data used for training the model and a regularization parameter for the SVM. The module also supports pre-processing and pre-transformation on the independent variables. For instance, the user can use builtin modules to normalize the independent variables by applying whitened PCA or a Z-score transformation, and then transform the data to a quadratic scale to better capture the distribution of the independent variables.

GNoT allows the user to visually investigate the characteristics of the data and the model produced by the SVM. Using the automatically produced visualization, the user can analyze the model fit on the test data by brushing and selecting a value range for each independent variable. The user can then visually examine the corresponding change in the distribution of the other independent variables and the model fit.

In ML_SVM ➥, we demonstrate the application of ML_SVM module of GNoT on the Wisconsin breast cancer dataset [2]. Figure 6 shows the resulting page. The side bar provides the summary of the results. Even with this basic model, we achieve an accuracy of 97% when predicting 20% of the samples by learning the model on the rest. The visualiza-

[7] http//http://werkzeug.pocoo.org/

[8] http://documentcloud.github.io/visualsearch/



MODULE: ml_svm_linear TABLE: public.breast_cancer_wisconsin FIELD: class-3 FIELD: cellshape FIELD: thickness FIELD: cellsize FIELD: normal_nucleoli RATIO: 0.8 PRE_PROCESS: Z-Score REGULARIZER: 10 RELOAD: 1

(a) User input

Loaded fresh.

FIELD_Y: *class-3*, on FIELD_X(predictors): *cellshape,thickness,cellsize,normal_nucleoli* from TABLE: *public.breast_cancer_wisconsin*

- Predicting class-3 using 80.00 % of the data ordered randomly
- Number of samples: 683
- Number of features: 4
- Accuracy: 0.9343
- Precision: 0.9362
- Recall: 0.8800
- F1: 0.9072

Built with d3.js.

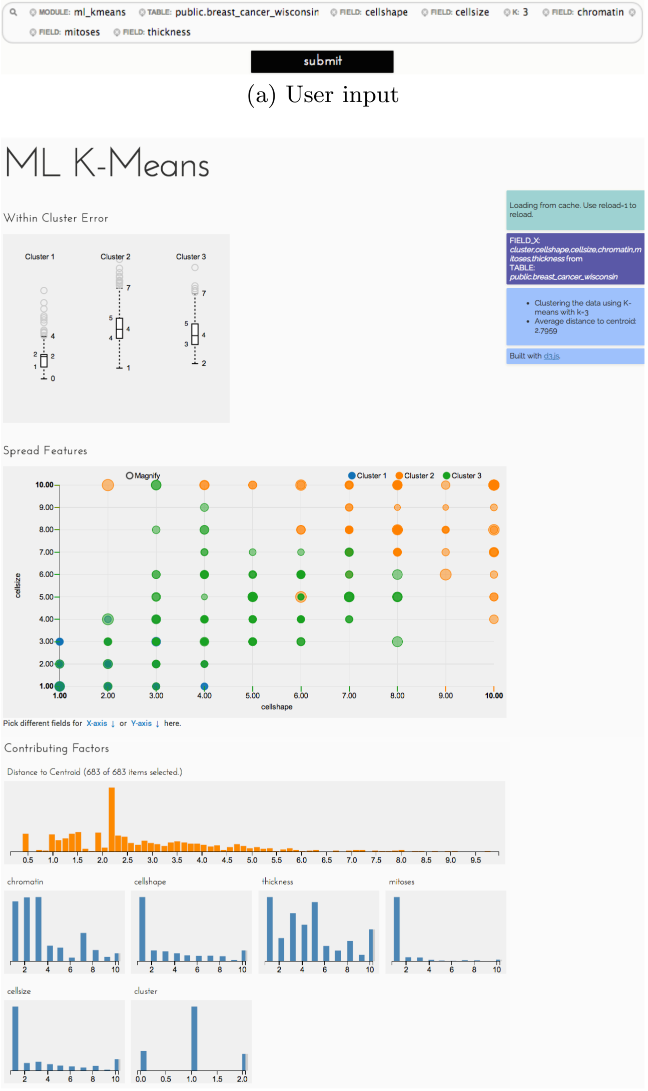
Area Under ROC: 0.9872

Distance from hyperplane (137 of 137 items selected.)

Pick different fields for X-axis ↓ or Y-axis ↓ here.

(b) Visualization Output

Figure 6: (a) **Query** `|MODULE:ml_svm_linear TABLE: public.breast_cancer_wisconsin FIELD:class-3 FIELD :cellshape FIELD:thickness FIELD:cellsize FIELD :normal_nucleoli RATIO:0.8 PRE_PROCESS:Z-Score REGULARIZER:10 |` **produces the visualization.** (b) **The visualization gives the summary of the results, feature weights of the model, and receiver operating characteristic curve (to show the accuracy of the model on the test data). It also allows the user to interactively examine the fit of the model on the test data by brushing and selecting a value range on each of the independent variable, and corresponding distributions on other variables.**



MODULE: ml_kmeans TABLE: public.breast_cancer_wisconsin FIELD: cellshape FIELD: cellsize K: 3 FIELD: chromatin FIELD: mitoses FIELD: thickness

(a) User input

Loading from cache. Use reload=1 to reload.

FIELD_X: *cluster,cellshape,cellsize,chromatin,mitoses,thickness* from TABLE: *public.breast_cancer_wisconsin*

- Clustering the data using K-means with k=3
- Average distance to centroid: 2.7959

Built with d3.js.

Distance to Centroid (683 of 683 items selected.)

Pick different fields for X-axis ↓ or Y-axis ↓ here.

(b) Visualization Output

Figure 7: (a) **Query** `|MODULE:ml_kmeans TABLE: public.breast_cancer_wisconsin FIELD:cellshape FIELD:cellsize FIELD:chromatin FIELD:mitoses FIELD :thickness K:3 |` **produces the visualization.** (b) **The visualization gives the summary of the model, cluster spreads, and the distribution of the clusters against any two fields. It also allows the user to interactively examine the fit of the model by brushing and selecting a value range on each of the fields, and corresponding distributions on other fields.**

tions allow the users to understand the characteristics of the model. The bar chart shows the weights of four features and the intercept (from the linear fit). The next graph shows the accuracy of the model on the test data with the receiver operating characteristic curve and the area under the curve. The scatter plot allows the user to visualize the spread of the samples from the test set on any two dimensions (the space of any two features) and the projection of the separating hyperplane on those two dimensions. The color and the shape of the samples indicate whether they were correctly classified, and the correct labels of the samples respectively. The visualizations on this page are connected to each other. The cross filter allows the user to brush and select a value range of an independent variable. This will update the distributions of the rest of the independent variables and the feature spread. Using cross filter, the user can see the contributing factors for large errors (by filtering by large positive distances from hyperplane): smaller cell sizes stand out as the most difficult to predict.

5.2 ML_K-Means

ML_K-Means allows a user to apply a K-means clustering algorithm to the data. The module also supports preprocessing and pre-transformation on the field values. The user can analyze the model fit on the data by brushing and selecting a value range for each of the fields. The user can also then visually examine the corresponding change in the distribution of the other fields and the model fit.

In ML_K-Means ➥, we demonstrate the application of ML_K-Means on Wisconsin breast cancer dataset [2]. Using the resulting page (Figure 7) we can try different field ranges and their effects in the resulting cluster distributions.

6. USING GNOT: A CASE STUDY

We built GNoT to help with exploring new datasets in solving real-world applications. We now describe an example application to demonstrate its usage and capabilities.

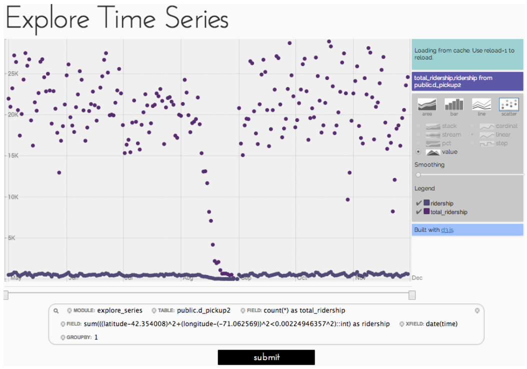
In early 2014, MIT Big Data Initiative at CSAIL together with the City of Boston hosted a Big Data Challenge[9] to gain new insights into how people use all modes of transportation to travel in and around the downtown Boston area. We use the Boston taxi dataset from this challenge.

6.1 Exploring Boston taxi data

Below, we outline how one might use GNoT to explore this dataset, generate hypotheses, and validate them. We use the demonstration site setup at `http://ddmg1.csail.mit.edu:4999` for following expositions. Links to demo pages are identified by ➥ whenever they are available.

- We start with a visualization of the raw data using query `|MODULE:explore_raw TABLE:public.d_pickup2 FIELD:* LIMIT:10|`. From the output of the Explore_Raw module as available at Pickup: Raw ➥, we understand that there are 5 fields: trip_id, time, address, longitude, and latitude.

- Next, we used the Explore_Calendar module to see the distribution of the data over the time span with query `|MODULE:explore_calendar TABLE:public.d_pickup2 XFIELD:date(time)|`. From the output Pickup: Calendar ➥ (Figure 5), we can see that the data spans



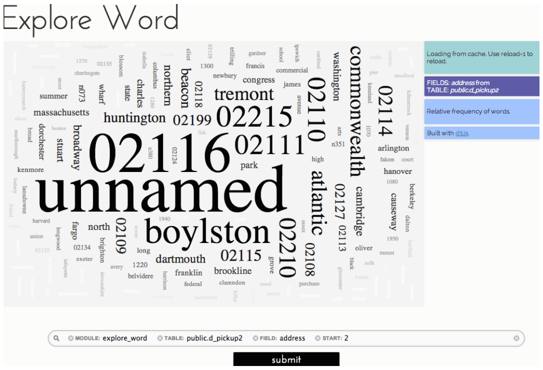
Loading from cache. Use reload=1 to reload.

total_ridership,ridership from public.d_pickup2

Built with d3.js.

MODULE: explore_series TABLE: public.d_pickup2 FIELD: count(*) as total_ridership FIELD: sum(((latitude-42.354008)^2+(longitude-(-71.062569))^2<0.00224946357^2)::int) as ridership XFIELD: date(time) GROUPBY: 1

Figure 8: Time series module in GNoT: Comparing the ridership around a location with that of the whole city using GNoT.



Loading from cache. Use reload=1 to reload.

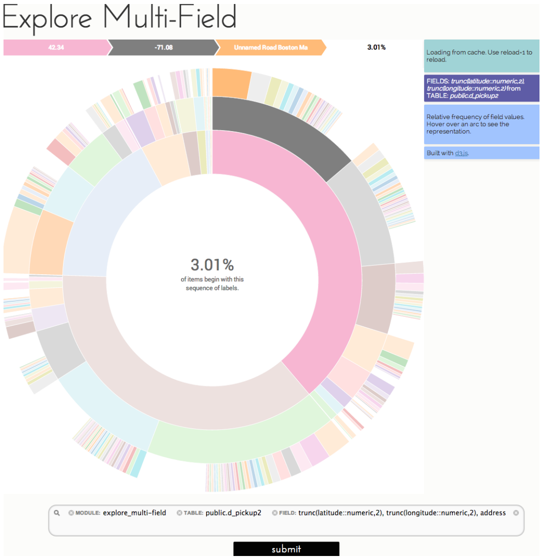
FIELDS: *address* from TABLE: *public.d_pickup2*

Relative frequency of words.

Built with d3.js.

MODULE: explore_word TABLE: public.d_pickup2 FIELD: address START: 2

Figure 9: Word module in GNoT: Exploring the popular words in the pickup addresses of taxi rides using GNoT.



Loading from cache. Use reload=1 to reload.

FIELDS: *trunc(latitude::numeric,2), trunc(longitude::numeric,2)* from TABLE: *public.d_pickup2*

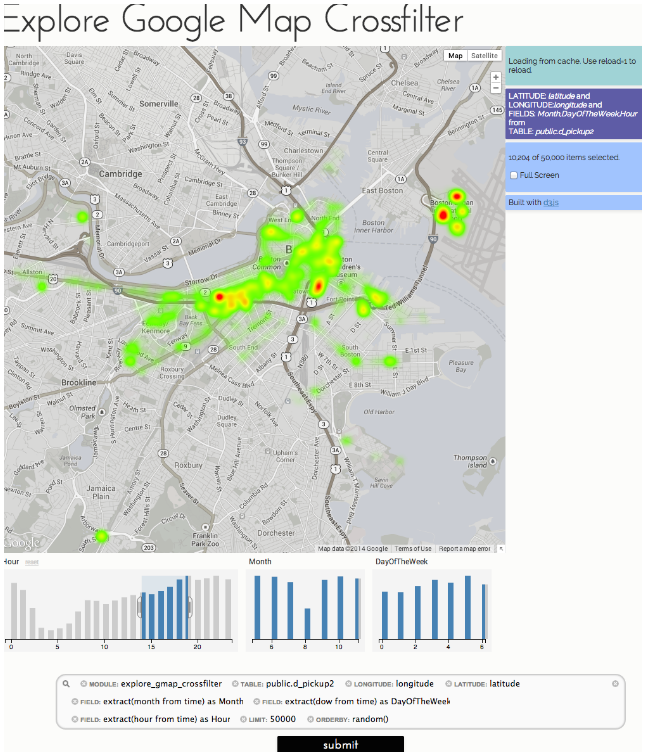
Relative frequency of field values. Hover over an arc to see the representation.

Built with d3.js.

MODULE: explore_multi-field TABLE: public.d_pickup2 FIELD: trunc(latitude::numeric,2), trunc(longitude::numeric,2), address

Figure 10: Multi-field module in GNoT: Visualizing the hierarchical split by various features.

[9] http://bigdatachallenge.csail.mit.edu



Loading from cache. Use reload=1 to reload.

LATITUDE: *latitude* and LONGITUDE:*longitude* and FIELDS: *Month,DayOfTheWeek,Hour* from TABLE: *public.d_pickup2*

10,204 of 50,000 items selected.

□ Full Screen

Built with d3.js

MODULE: explore_gmap_crossfilter TABLE: public.d_pickup2 LONGITUDE: longitude LATITUDE: latitude
FIELD: extract(month from time) as Month FIELD: extract(dow from time) as DayOfTheWeek
FIELD: extract(hour from time) as Hour LIMIT: 50000 ORDERBY: random()

Figure 11: Google maps with crossfilter in GNoT: Visualizing the factors contributing to the ridership with GNoT.

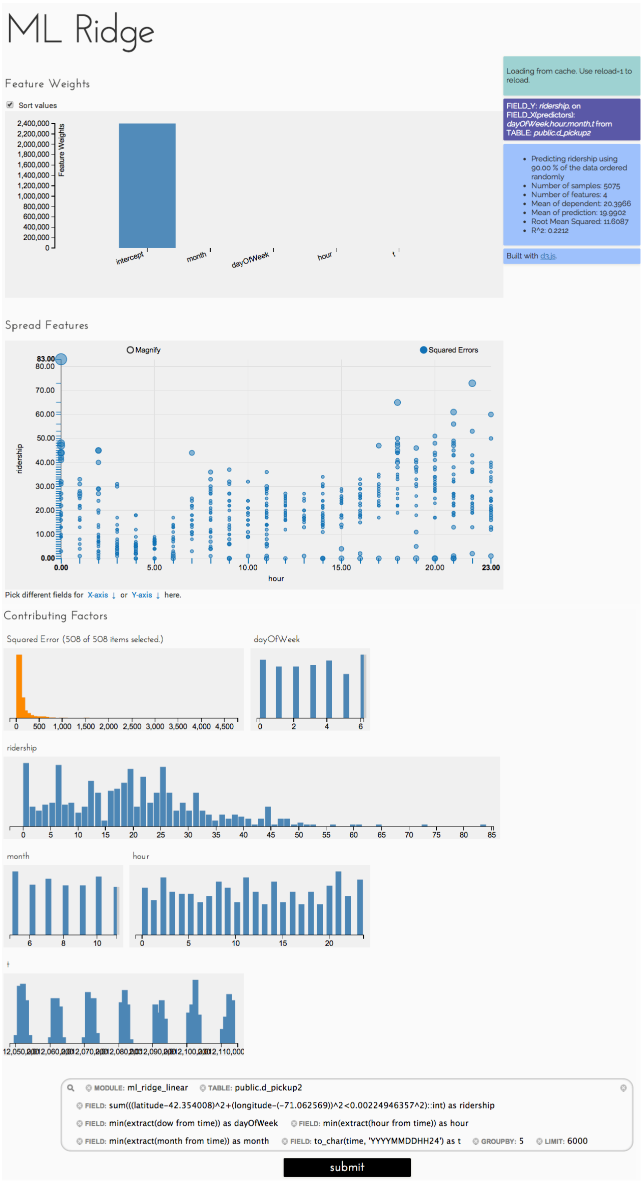
from May 1, 2012 to November 30, 2012, and that the data is missing for about two weeks during the latter part of August.

- We next used the Explore_Diff module to compare the ridership at specific locations to overall ridership across Boston with query `|MODULE:explore_diff TABLE:public.d_pickup2 XFIELD:date(time)FIELD:sum(((latitude-42.354008)^2+(longitude-(-71.062569))^2< 0.00224946357^2)::int)as ridership FIELD:count(*) as total_ridership|`.
  In the resulting page Pickup: Diff ➥, we see that the ratio fluctuates hugely.

  We used Explore_Series to get a more detailed view. It is as easy as changing the module option in the query box. In the resulting page Pickup: Series ➥, we have many types of visualizations at our disposal (Figure 8).

- We used the Explore_Word module to see the popular words in the addresses with query `|MODULE:explore_word TABLE:public.d_pickup2 FIELD:address|`. It shows the words sized according to the number of occurrences: Pickup: Word ➥. When we viewed this visualization, the word “Boston” was dominant, not surprising given the data set.

  We therefore asked it to omit “Boston” with query `|MODULE:explore_word TABLE:public.d_pickup2 FIELD:address START:1|`, and got the visualization Pickup: Word2 ➥ (Figure 9). (“Unnamed road” is a road within Logan Airport.)



Loading from cache. Use reload=1 to reload.

FIELD_Y: *ridership*, on FIELD_X(predictors): *dayOfWeek,hour,month,t* from TABLE: *public.d_pickup2*

- Predicting ridership using 90.00 % of the data ordered randomly
- Number of samples: 5075
- Number of features: 4
- Mean of dependent: 20.3966
- Mean of prediction: 19.9902
- Root Mean Squared: 11.6087
- R^2: 0.2212

Built with d3.js.

MODULE: ml_ridge_linear TABLE: public.d_pickup2
FIELD: sum(((latitude-42.354008)^2+(longitude-(-71.062569))^2<0.00224946357^2)::int) as ridership
FIELD: min(extract(dow from time)) as dayOfWeek FIELD: min(extract(hour from time)) as hour
FIELD: min(extract(month from time)) as month FIELD: to_char(time, 'YYYYMMDDHH24') as t GROUPBY: 5 LIMIT: 6000

Figure 12: ML_Ridge_Linear module in GNoT: Visualizing the fit of ridge regression predicting the number of pickups in an hour window.

- We used Explore_Multi-Field to understand which latitude, longitude, and specific address with high ridership in query `|MODULE:explore_multi-field TABLE: public.d_pickup2 FIELD:trunc(latitude::numeric,2) , trunc(longitude::numeric,2), address|`. Here, we truncated the coordinates to the second decimal point using SQL itself. The resulting visualization Pickup: Multi-field ➥ is seen in Figure 10. We can see latitude 42.34 has the highest with 34% of the ridership, out of which longitude -71.08 takes the highest ridership at 14%. Within this combination, “unnamed road Boston” (which is part of Boston Logan international airport) takes the highest fraction. Altogether, it represents 3% of the total taxi ridership.

- However, making sense of the latitudes and longitudes is easier when they are plotted on a map. We used Explore_Gmap to view them in Google maps with query `|MODULE:explore_gmap TABLE:public.d_pickup2 LATITUDE:trunc(latitude::numeric,3) LONGITUDE:trunc(longitude::numeric,3) FIELD:count(*)|` and got Pickup: Gmap ➥.

  We also visualized the changes in the ridership against time with query `|MODULE:explore_gmap TABLE:public .d_pickup2 LATITUDE:trunc(latitude::numeric,3) LONGITUDE:trunc(longitude::numeric,3)XFIELD:date (time)FIELD:count(*)|` and got Pickup: Gmap2 ➥.

- In order to gain a deeper insight into the ridership based on hour of the day, month of the year, and day of the week, we used Explore_Gmap_Crossfilter in query `|MODULE:explore_gmap_cross_filter TABLE:public.d_pickup2 LATITUDE:latitude LONGITUDE:longitude FIELD:extract(month from time) as Month FIELD:extract(dow from time)as Day FIELD:extract(hour from time)as Hour LIMIT:50000 ORDERBY:random()|` and got Pickup: Gmap Crossfilter ➥ (Figure 11). Using the brushing feature, we can learn that between 10 PM and 12 AM taxi rides peak around Boston University, Boylston Street/Prudential Tower, and Terminal E at Boston Logan international airport.

6.1.1 *Machining Learning Models*

First, we used k-means to perform unsupervised clustering of the coordinates with query `|MODULE:ml_kmeans TABLE: public.d_pickup2 FIELD:latitude, longitude LIMIT:1000 K:5|`. Arbitrarily, we chose to partition the coordinates into 5 clusters Pickup: K-means ➥. We observe the segments formed by dividing the city into 5 explainable regions. We could also interactively explore different number of segments by adjusting k.

What if we add the day of the week as the third dimension? Since the range of the day of the week is not similar to the range of latitude and longitude (i.e., they have different units), we apply Z-score normalization before applying K-means in the query `|MODULE:ml_kmeans TABLE:public. d_pickup2 FIELD:latitude, longitude, extract(dow from time)PRE_PROCESS:Z-Score LIMIT:1000 K:5|`. Here, we see a totally different pattern emerging Pickup: K-means2 ➥.

Now, let's look at a problem of predicting the number of pickups within 250 meters of a location (e.g., (-71.057114, 42.343365)) within a given time window. Before building a complex model, we can use GNoT to visually explore the data in order to test the viability of the task and identify useful feature combination.

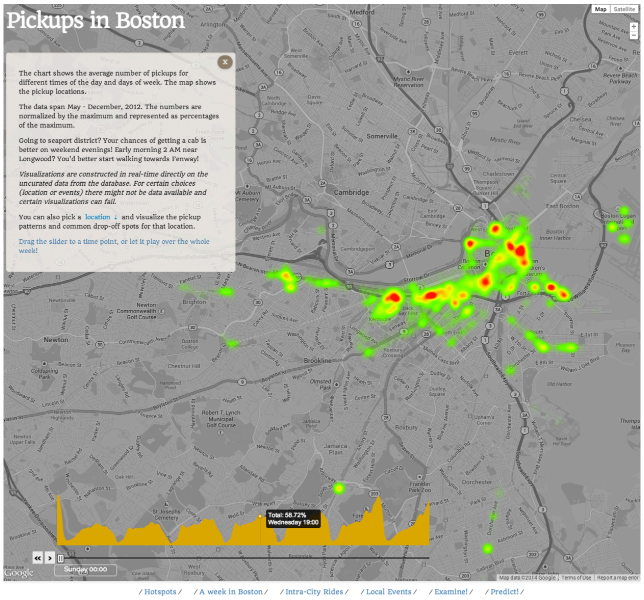
We used the ML_Ridge_Linear module to explore the accuracy achievable in predicting the ridership in an hour with only three features: day of the week, hour of the day, and month with query `|MODULE:ml_ridge_linear TABLE: public.d_pickup2 FIELD:sum(((latitude-42.354008)^2+(longitude-(-71.062569))^2<0.00224946357^2)::int)as ridership FIELD:min(extract(dow from time))as dayOfWeek FIELD:min(extract(hour from time))as hour FIELD:min(extract(month from time))as month FIELD: to_char(time, 'YYYYMMDDHH24')as t GROUPBY:5 LIMIT:6000|`.

We can see the resulting page Pickup: Ridge Linear ➥ in Figure 12. Even with this basic model, we achieve a coefficient of determination ($R^2 = 0.22$) when predicting 10% of the time windows by learning the model on the rest. Using GNoT, we can also try different feature combinations, feature transformations such as applying interactions and quadratic transformations on the features, and pre-processing such as applying Z-score or PCA transformation.

6.2 Custom solution with GNoT

The visualizations built with GNoT to assist with rapid visual data mining, called *Boston Rides* ➥[10], won the first prize in the competition. *Boston Rides* is a customized version of GNoT where the user's queries are hardcoded, and they are guided through a predefined exploration path.

Boston Rides allows the user to explore the data through 6 different visualization types: hotspots for pickups, daily and hourly variations of ridership (Figure 13), popular intracity routes, factors contributing to variations in pickups, and machine learning generated models that predict the number of pickups in an hour window using various features.



The chart shows the average number of pickups for different times of the day and days of week. The map shows the pickup locations.

The data span May - December, 2012. The numbers are normalized by the maximum and represented as percentages of the maximum.

Going to seaport district? Your chances of getting a cab is better on weekend evenings! Early morning 2 AM near Longwood? You'd better start walking towards Fenway!

Visualizations are constructed in real-time directly on the uncurated data from the database. For certain choices (location or events) there might not be data available and certain visualizations can fail.

You can also pick a location ↓ and visualize the pickup patterns and common drop-off spots for that location.

Drag the slider to a time point, or let it play over the whole week!

Figure 13: Daily and hourly variation in ridership as visualized in Boston rides with GNoT. The module makes use of D3 and the Google maps API to construct the visualization. The popup box in the figure contains the guide.

6.3 Discussion

One of the biggest advantages with GNoT is that it offers the choice between the ease of use and flexibility.

In the first part of the case study (Section 6.1), we used existing modules. This requires neither programming nor software engineering expertise, and allows us to rapidly explore the data with great ease. Each query requires very

[10] http://bostonrides.info

minimal input from the user, and the queries are intuitive as they follow the SQL style.

For the second part of the case study (Section 6.2), we demonstrated the use of customized modules. Customizing GNoT to build Boston Rides took less than 10 human-hours. Using a low-level tool or a language framework such as R would have been far more timing consuming and would have lacked the interactive features offered by GNoT. Using an integrated solution such as IBM's many eyes or Tableau would constrain the flexibility in making use of external libraries to offer complex machine learning capabilities. The resulting solution would still lack the latest interactive features offered by D3. Finally, building a system from the scratch by connecting individual components: an RDBMS (e.g., Vertica), machine learning library (Weka), and a graphics library (D3), would result in a solution similar to that of GNoT, but only after spending many more human hours.

7. SUMMARY

Visual data mining combines the flexibility, creativity, and general knowledge of a human with brute computational power. In this paper, we describe a novel system, GNoT, that supports interactive knowledge discovery by interconnecting state of the art tools for visualization, relational database management, and machine learning. The system provides the glue connecting these kinds of components, and thus is able to "ride the wave" of improvements in each of these areas.

8. REFERENCES

[1] A. Aiken, A. Woodruff, J. Chen, and M. Stonebraker. Tioga-2: A direct manipulation database visualization environment. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 208–208. IEEE Computer Society, 1996.

[2] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[3] C. M. Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[4] M. Bostock, V. Ogievetsky, and J. Heer. D3 Data-Driven Documents. *Visualization and Computer Graphics, IEEE Transactions on*, 2011.

[5] C. Cortes and V. Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[6] L. Denuzière, A. Granicz, and A. Tayanovskyy. Visualizing data on the web. In *DDFP '13: Proceedings of the 2013 workshop on Data driven functional programming*, 2013.

[7] M. Derthick, J. Kolojejchick, and S. F. Roth. An interactive visualization environment for data exploration. In *Proceedings of the Knowledge Discovery in Databases*, pages 2–9. Press, 1997.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[9] D. A. Keim. Information visualization and visual data mining. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):1–8, 2002.

[10] A. Lamb, M. Fuller, R. Varadarajan, N. Tran, B. Vandier, L. Doshi, and C. Bear. The Vertica Analytic Database: C-Store 7 Years Later. *arXiv.org*, 2012.

[11] C. North and B. Shneiderman. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In *Proceedings of the working conference on Advanced visual interfaces*, pages 128–135. ACM, 2000.

[12] S. F. Roth, P. Lucas, J. A. Senn, C. C. Gomberg, M. B. Burks, P. J. Stroffolino, A. Kolojechick, and C. Dunmire. Visage: a user interface environment for exploring information. In *Information Visualization'96, Proceedings IEEE Symposium on*, pages 3–12. IEEE, 1996.

[13] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):52–65, 2002.

[14] M. Stonebraker. Sql databases v. nosql databases. *Communications of the ACM*, 53(4):10–11, 2010.

[15] J. Varia and S. Mathew. Overview of amazon web services. *Amazon Web Services*, 2012.

[16] 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.

[17] C. Weaver. Building highly-coordinated visualizations in improvise. *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 159–166, 2004.

APPENDIX

Table 1 lists the basic set of core modules. The table also details the fields supported by the modules. A few other fields are also optionally supported by many modules.

- orderBy: Order by field is useful in selecting a desired region when used together with limit and start. Time series (the modules that expect X Field) are always ordered by X Field.
- groupBy: When groupBy fields are used, values fields must be aggregates.

In addition to the required and optional inputs lists in the table, all modules support the following optional fields.

- where: Text entry used to filter data (e.g. *total* > 1000).
- limit: A numeric entry to limit the number of records retrieved (e.g. 1000). Typically each module assumes a reasonable limit when this option is not specified.
- start: A numeric entry representing the offset (e.g. 1000).
- reload: A binary switch instructing GNoT to ignore the cache.
- view: When fields are derived from a complex query, it is best specified inside as a view. Then, the supplied table is ignored and query is executed against this view.

Module	Description	Required Inputs	Optional Inputs
Raw	Outputs the raw data	Table T, Fields $f_1 \dots f_n$	groupBy fields $f_{a_1} \dots f_{a_r}$, orderBy fields $f_{b_1} \dots f_{b_s}$
Calendar	Displays date-based heatmap of data	Table T, X Field f_{date}, Value field f_{val} [1]	-
Field	Shows relative frequency of values within a field	Table T, Field f	-
Multi-Field	Shows relative frequency of values of multiple fields in a hierarchical manner	Table T, Fields $f_1 \dots f_n$	-
Series	Plots multiple time series. Can be visualized as an area, bar, line, or scatter plot with multiple options for combining series and smoothing plots. Can be annotated by a field in the same table	Table T, X Field f_x, Y Fields $f_{y1} \dots f_{yn}$	Annotation field f_a, orderBy fields $f_{b_1} \dots f_{b_s}$
Diff	Plots the difference of two fields in a time series	Table T, X Field f_x, Y Fields f_{y1}, f_{y2}	orderBy fields $f_{b_1} \dots f_{b_s}$
Word	Uses word cloud to display frequency of words in a collection of text	Table T, Text field f_t	-
Word Series	Plots the frequency of words over time.	Table T, X Field f_x, Text Field f_y	-
Graph	Undirected graph with distinguishable node types	Table T, Source Field f_s, Target Field f_t	orderBy fields $f_{b_1} \dots f_{b_s}$
Matrix	Visualization of a sortable matrix where cells represent values between entities. Optional field Linkgroup can be used to cluster nodes.	Table T, Source Field f_s, Target Field f_t, Value Field f_{val}	Linkgroup Field f_c, orderBy fields $f_{b_1} \dots f_{b_s}$
Digraph	Directed graph with node values	Table T, Source Field f_s, Target Field f_t, Value Field f_{val}	orderBy fields $f_{b_1} \dots f_{b_s}$
Scatter	Visualization of scatter plot on a two dimensional plane f_x, f_y. Two additional fields determine the radius (f_z) of the markers and grouping (class f_c).	Table T, Value fields f_x, f_y, f_z, f_c	groupBy fields $f_{a_1} \dots f_{a_r}$, orderBy fields $f_{b_1} \dots f_{b_s}$
Correlations	Creates a scatter plot matrix where each node is a plot of the values of one field against the values of another. Values can be filtered on all plots by selecting a region on a single plot. First field represents the sample classes.	Table T, Value fields $f_c, f_{v_1} \dots f_{v_n}$. Limit number of fields to 5	groupBy fields $f_{a_1} \dots f_{a_r}$, orderBy fields $f_{b_1} \dots f_{b_s}$
Bar	Simple bar graph	Table T, Fields f_x, f_y	-
Crossfilter	Plots distribution of values for each field. Values can be filtered on all plots by selecting a region on a single plot.	Table T, Value Fields $f_{v_1} \dots f_{v_n}$	groupBy fields $f_{a_1} \dots f_{a_r}$, orderBy fields $f_{b_1} \dots f_{b_s}$
gMaps	The distribution of locations is displayed in heatmap format with an interactive map display. If f_x is specified, then time play of the heatmap against f_x is provided.	Table T, Longitude field f_{lon}, Latitude field f_{lat}	X Field f_x, Value field f_1 [1]
gMaps Crossfilter	In addition to the heatmap, user is also given a histogram of each of the specified filtering fields $f_{f1} \dots f_{fn}$. The user can select ranges of values within these distributions to display on the map.	Table T, Longitude field f_{lon}, Latitude field f_{lat}, filtering fields $f_{f1} \dots f_{fn}$	
ML K-Means	Cluster into k clusters using the K-means clustering algorithm	Table T, Value fields $f_1 \dots f_n$, Number of clusters k	Pre-processing method (PCA, Whitened PCA, or Z-Score), pre-transform method (Quadratic, Purely quadratic, or Interaction), groupBy fields $f_{a_1} \dots f_{a_r}$, orderBy fields $f_{b_1} \dots f_{b_s}$
ML Ridge	Fraction r of X is used to train a linear regression model. The remaining $(1 - r)$ fraction is then used as a test set.	Table T, Value fields fields $f_Y, f_{X_1} \dots f_{X_n}$, regularizer α, ratio r	
ML SVM	Similar to ML Ridge, but performs classification using SVM algorithm	Table T, Value fields fields $f_Y, f_{X_1} \dots f_{X_n}$,regularizer α, ratio r	

Table 1: List of core modules

Decomposing a Sequence into Independent Subsequences Using Compression Algorithms

Hoang Thanh Lam
Technische Universiteit
Eindhoven
2 Den Dolech
Eindhoven, the Netherlands
t.l.hoang@ie.ibm.com

Julia Kiseleva
Technische Universiteit
Eindhoven
2 Den Dolech
Eindhoven, the Netherlands
j.kiseleva@tue.nl

Mykola Pechenizkiy
Technische Universiteit
Eindhoven
2 Den Dolech
Eindhoven, the Netherlands
m.pechenizkiy@tue.nl

Toon Calders
Universite Libre de Bruxelles
CP 165/15 Avenue F.D.
Roosevelt 50
B-1050 Bruxelles
toon.calders@ulb.ac.be

ABSTRACT

Given a sequence generated by a random mixture of independent processes, we study compression-based methods for decomposing the sequence into independent subsequences each corresponds to an independent process. We first show that the decomposition which results in the optimal compression length in expectation actually corresponds to an independent decomposition. This theoretical result encourages us to look for the decomposition that incurs the minimum description length to solve the independent decomposition problem. A hierarchical clustering algorithm is proposed to find that decomposition. We perform experiments with both synthetic and real-life datasets to show the effectiveness of our method in comparison with the state of the art method.

General Terms

Data mining

Keywords

Pattern mining, independent component decomposition, minimum description length principle, data compression

1. INTRODUCTION

Many processes produce extensive sequences of events, e.g. alarm messages from different components of industrial machines or telecommunication networks, web-access logs, clickstream data, geographical events record, etc. In many cases, a sequence consists of a random mixture of independent or loosely connected processes where each process produces a specific disjoint set of events that are independent from the other events.

It is useful to decompose the sequence into a number of independent subsequences. This data preprocessing step provides us with a lot of conveniences for further analysis with each independent process separately. In fact, independent sequence decomposition was used to improve the accuracy of predictive models by building local predictive model for each independent process separately instead of building it for the whole data [1, 2]. Besides, in descriptive data mining, people are usually interested in summarizing the data. They are eager to discover the dependency structure between events in the data and also want to know how strong the dependency is in each independent component [14]. If the dependency in a component is strong, it maybe associated with an explainable context that can help people understand the data better.

The sequence independent decomposition problem was first introduced by Mannila et al. in [3]. The authors proposed a method based upon a statistical hypothesis testing for the dependency between events. A drawback of the statistical hypothesis testing method is that the p-value derived from the test is a score subjective to the null hypothesis. The p-value does not convey any information about how strong the dependency between the events is. Moreover, the method introduces two parameters which require manual tunings for different applications. The method is also easily vulnerable to false detected connections between events (see the discussion in the next section for an example).

In this paper, we revisit the independent decomposition problem from the prospect of data compression. In order to illustrate the connection between data compression algorithms and the independent decomposition problem let us first consider a simple example. Assume that we have two sequences:

$$S_1 = abababababababababababababababab$$

$$S_2 = aaababbbabbaabbaababababaaabbbabab$$

The first sequence is very regular, after an occurrence of a there is an occurrence of b. In this case, it is clear that a and b are two dependent events. We can exploit this information to compress the sequence as follows: send the decoder the number of repetitions of ab, i.e. 16 times in S_1. Then we send the binary representations of a and b together. If the *Elias code* [4] is used to compress the natural number 16 and one bit is used to represent either a or b, the compressed size of S_1 is $9 + 1 + 1 = 11$ bits. On the other hand, if we do not exploit the dependency between a and b we need at least 32 bits to represent S_1. Therefore, by exploiting the knowledge

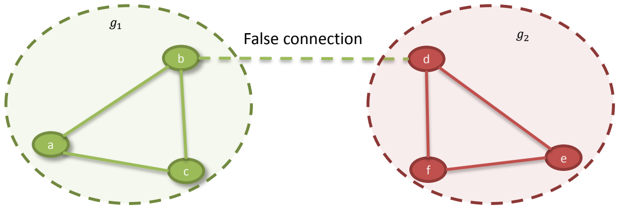


Figure 1: An example of two strongly connected and independent components. False connection between b and d is recognized just by chance or due to noise. The Dtest algorithm will merge two components together even only one false connection happens.

about the dependency between a and b we compress the sequence far better than the compression that considers a and b separately.

The second sequence seems like a random mixture between two independent events a and b. In this case, it does not matter how a compression algorithm does, the compression result will be very similar to the result of the compression algorithm that considers a and b separately.

In common-sense, two examples lead us to an intuition that if we can find the best way to compress the data then that compression algorithm may help us to reveal the dependency structure between events in a sequence simply because it will exploit that information for doing compression better. This intuition is inline with the general idea of the *Minimum Description Length* (MDL) principle [5] which always suggests that the best model is the one that describes the data in the shortest way. We get to the point to ask a fundamental question: What is the connection between the best model by the definition of the MDL principle [5] and the independent decomposition problem?

In this work, we study theoretical answers for the aforementioned question. In particular, we prove that the best model by the definition of the MDL principle actually corresponds to an independent decomposition of the sequence. This theoretical result motivates us to propose a data compression based algorithm to solve the independent decomposition problem by looking for a decomposition that can be used to compress the data most.

Beside being parameterless, the compression-based method provides us with a measure based on compression ratios showing how strong the connection between events of an independent component is. It can be considered as an interestingness measure to rank different independent components. We validate our method and compare it to the statistical hypothesis testing based approach [3] in an experiment with synthetic and real-life datasets.

2. RELATED WORKS

The sequence independent decomposition problem was first studied by Mannila et al. in [3]. The authors proposed a method based on statistical hypothesis testing for dependency between events. In this work, we call their method *Dtest* as for *Dependency Test*. The algorithm first performs dependency tests for every pair of events. Subsequently, it builds a dependency graph in which vertices correspond to events and edges connecting two dependent events.

An independent component of the output decomposition corresponds to a connected component of the graph. The Dtest approach has a drawback: it can merge two independent components together even when there is only one false connection (not a connection but erroneously detected as a connection) between two vertices across two components. For instance, Figure 1 shows two strongly connected components g_1 and g_2 of the dependency graph. If the dependency test between $b \in g_1$ and $d \in g_2$ produces wrong result, i.e. b and d pass the dependency test even they are independent, two independent components g_1 and g_2 will be merged into a single component.

In the experiment, we show that false connection is usually the case because of the following two reasons. First, the Dtest algorithm has two parameters. Setting of these parameters to avoid false connections is not always a trivial task. Second, the dependency between b and d can be incorrectly detected due to noises. Being different from the Dtest algorithm, our compression-based method is not easily vulnerable to false connections. Indeed, if two strongly connected components are independent to each other, the loose connection between b and d is not an important factor that can improve the compression ratio significantly when the two components are compressed together.

Indeed, independent component analysis for other types of data is a well studied problem in the literature [6]. For example, the ICA method was proposed to decompose a time series into independent components. However, the ICA method does not handle event sequence data.

Another closely related work concerns the item clustering problem studied under the context of itemset data [7]. The authors proposed a method to find clusters of strongly related items for data summarization. The work relies on the MDL principle which clusters items together such that it minimizes the description length of the data when the cluster structure is exploited for compressing the data. On one hand, our work proposes a different encoding to handle sequence data which is not handled by the encoding of [7]. On the other hand, we show a theoretical connection between the MDL principle and the independent component analysis. It gives a theoretical judgement for the model usually neutrally accepted as *the best* model by the definition of the MDL principle.

Finally, the idea of using data compression algorithms in data mining is not new. In fact, data compression algorithms were successfully used for many data mining tasks including data clustering [8] or data classification [9]. It was also used for mining non-redundant set of patterns in itemset data [10] and in sequence data [11]. Our work is the first one that proposes to use data compression for solving the independent sequence decomposition problem.

3. PROBLEM DEFINITION

Let $\sum = \{a_1, a_2, \cdots, a_N\}$ be the alphabet of events; denote $S_t = x_1 x_2 \cdots x_t$ as a sequence, where each $x_i \in \sum$ is generated by a random variable X_i ordered by its timestamp. The length of a sequence S is denoted as $|S|$.

We assume that S is generated by a stochastic process $\mathfrak{P}$. For any natural number n, we denote $P_t(X_t = x_t, X_{t+1} = x_{t+1}, \cdots, X_{t+n-1} = x_{t+n-1})$ as the joint probability of the sequence $X_{t+1}, X_{t+2}, \cdots, X_{t+n-1}$ governed by the stochastic process $\mathfrak{P}$, i.e. the probability of observing the subsequence $x_t x_{t+1} \cdots x_{t+n-1}$ at time point t.

A stochastic process is called *stationary* [4] if for any n the joint probability $P_t(X_t = x_t, X_{t+1} = x_{t+1}, \cdots, X_{t+n-1} = x_{t+n-1})$ does not depend on t, which means that $P_t(X_t = x_t, X_{t+1} = x_{t+1}, \cdots, X_{t+n-1} = x_{t+n-1}) = P_1(X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n)$ for any $t \geq 0$. In this work, we consider only stationary processes as in practice a lot of datasets are generated by a stationary process [4]. This assumption is made for convenience in our theoretical analysis although it is not a requirement for our algorithms to operate properly. Therefore, for a stationary process the joint probability $P_t(X_t, X_{t+2}, \cdots, X_{t+n-1})$ is simply denoted as $P(X_1, X_2, \cdots, X_n)$. For a given sequence S the probability of observing the sequence is simply denoted as $P(S)$.

Let $C = \{C_1, C_2, \cdots, C_k\}$ be a partition of the alphabet $\sum$ into k pairwise disjoint parts, where $C_i \bigcap C_j = \emptyset \; \forall i \neq j$ and $\bigcup_{i=1}^k C_i = \sum$. Given a sequence S, the partition C decomposes S into k disjoint subsequences denoted as $S(C_i)$ for $i = 1, 2, \cdots, k$. Let $P_i(S(C_i) = s)$ denote the marginal distribution defined on a set of subsequence with fixed size $|s| < |S|$.

Example 1. *Let the alphabet $\sum = \{a, b, c, d, e, f, g\}$ be partitioned into three disjoint parts: $C = \{C_1, C_2, C_3\}$ where $C_1 = \{a, b, c\}$, $C_2 = \{d, e\}$ and $C_3 = \{f, g\}$. The partition C decomposes the sequence $S = abdffeadcdeabgg$ into three subsequences $S(C_1) = abacab$, $S(C_2) = dedde$ and $S(C_3) = ffgg$.*

Denote α_i as the probability of observing an event belonging to the cluster C_i. Assume that $\mathfrak{P}_i$ is the stochastic process that generates $S(C_i)$.

Definition 1 (Independent decomposition). *We say that $C = \{C_1, C_2, \cdots, C_k\}$ is an independent decomposition of the alphabet if S is a random mixture of independent subsequences $S(C_i)$:*

$$P\left(S\left(\bigcup_{i=1}^k C_i\right)\right) = \prod_{i=1}^k \alpha_i^{|S(C_i)|} P_i\left(S(C_i)\right)$$

There are many independent decompositions, we are interested in the decomposition with maximum k; denote that decomposition as C^*. The problem of independent sequence decomposing can be formulated as follows:

Definition 2 (Sequence decomposition). *Given a sequence S and an alphabet $\sum$, find the maximum independent decomposition C^* of S.*

Theorem 1 (Unsolvable). *Observing a sequence generated by a stochastic (stationary) process with bounded size M there is no deterministic algorithm that solves the sequence independent decomposition problem exactly.*

Proof. Assume that there is a deterministic algorithm A that can return the maximum independent decomposition exactly when up to $2 * M$ events of a sequence are observed. Consider the following alphabet $\sum = \{a, b\}$ and two different stationary processes:

- The events a and b are drawn independently at random with probability 0.5
- The events a and b are drawn from a simple Markov chain with two states a and b and $P(a \mapsto b) = P(b \mapsto a) = 1.0$

The sequence $S = (ab)^M$ with length $2M$ can be generated by both stationary processes with non-zero probability. Therefore, by observing S, the algorithm A cannot decide the maximum independent decomposition C^* because for the latter process $C^* = \{\{a, b\}\}$ while for the former process $C^* = \{\{a\}, \{b\}\}$. This point leads to contradiction. □

4. SEQUENCE COMPRESSION

Given an observed sequence with bounded size, Theorem 1 shows that the problem in Definition 2 is unsolvable. However, in this section we show that it can be solved asymptotically by using data compression algorithms. We first define encodings that we use to compress a sequence S given a decomposition $C = \{C_1, C_2, \cdots, C_k\}$.

Given an event a_i denote $I(a_i) = j$ as the identifier of the cluster (partition) C_j which contains a_i. Let $S = x_1 x_2 \cdots x_n$ be a sequence, denote $I(S)$ as the cluster identifier sequence, i.e. $I(S) = I(x_1)I(x_2) \cdots I(x_n)$.

Example 2. *In Example 1, given the decomposition $C_1 = \{a, b, c\}$, $C_2 = \{d, e\}$ and $C_3 = \{f, g\}$ the cluster identifier sequence of S is $I(abdffeadcdeabgg) = 112332121221133$.*

If the distribution of the cluster identifiers is given as $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_k)$, where $\sum_{j=1}^k \alpha_j = 1$, the *Huffman code* [4] can be used to encode each cluster identifier j in the sequence $I(S)$ with a codeword with length proportional to $-\log \alpha_j$. In expectation, if the identifiers are independent to each other that encoding results in the minimum compression length for the cluster identifier sequence [4]. Denote $E^*(I(S))$ as the encoded form of $I(S)$ in that *ideal encoding.*

In practice, we don't know the distribution $(\alpha_1, \alpha_2, \cdots, \alpha_k)$. However, the distribution can be estimated from data. An encoding is called *asymptotically optimal* if:

$$\lim_{S \mapsto \infty} \frac{|E^+(I(S))|}{|S|} = H(\alpha)$$

Where $H(\alpha)$ denotes the entropy of the distribution α. An example of E^+ is the one that uses the empirical value $\frac{|S(C_i)|}{|S|}$ as an estimate of α_i.

Let Z be a data compression algorithm that gets the input as a sequence and returns the compressed form of that sequence. Z is an *ideal compression algorithm* denoted as Z^* if $|Z^*(S)| = -\log P(S)$. In expectation, Z^* results in the minimum compression length for the data [4]. In practice, we don't know the distribution $P(S)$ however we can use an asymptotic approximation of the ideal compression algorithm, e.g. the *Lempel-Ziv algorithms* [4]. An algorithm is asymptotically optimal denoted as $Z^+(S)$ if $\lim_{S \mapsto \infty} \frac{|Z^+(S)|}{|S|} = H(\mathfrak{P})$, where $\mathfrak{P}$ is the stationary process that generates S.

Given a decomposition $C = \{C_1, C_2, \cdots, C_k\}$ and a compression algorithm Z, the sequence S can be compressed in two parts, the first part corresponds to the compressed form of the identifier sequence $I(S)$. The second part contains k compressed subsequences $Z(S(C_1)), Z(S(C_2)), \cdots, Z(S(C_k))$. In summary, the compressed sequence consists of the size of the sequence in an encoded form denoted as $E(|S|)$, the compressed form of the cluster identifier sequence and k compressed subsequences. In this work, we use the term *ideal encoding* to refer to the encoding that uses E^* and Z^* and *asymptotic encoding* to refer to the encoding that uses E^+ and Z^+.

Example 3. *In Example 1, given the decomposition $C_1 = \{a, b, c\}$, $C_2 = \{d, e\}$ and $C_3 = \{f, g\}$, the sequence S in Example 1 can be encoded as follows: $E(15)$ $E(I(S))$ $Z(S(C_1))$ $Z(S(C_2))$ $Z(S(C_3))$*

5. THE MDL PRINCIPLE AND INDEPENDENT DECOMPOSITIONS:

The description length of the sequence S using the decomposition C can be calculated as: $L^C(S) = |E(|S|)| + |E(I(S))| + \sum_{i=1}^{k} |Z(S(C_i))|$. In this encoding, the term $|E(|S|)|$ is invariant when S is given. The decomposition C can be considered as a model and the cost to describe that model is equal to the term $|E(I(S))|$, meanwhile the latter term $\sum_{i=1}^{k} |Z(S(C_i))|$ corresponds to cost of describing the data given the model C. Therefore, according to the minimum description length principle we may try to find a decomposition resulting in the minimum description length in expectation, which is believed to be the best model for describing the data.

This section introduces two key theoretical results: subsection 5.1 shows an ideal analysis that given the data with bounded size, under an ideal encoding the best model describing the data corresponds to an independent decomposition and vice versa. Subsection 5.2 discusses an asymptotic result showing that under an asymptotic encoding, any independent decomposition corresponds to the best model by the definition of the MDL principle.

5.1 Analysis under the ideal encoding

We recall some definitions in information theory. Given a discrete probability distribution $P = \{\alpha_1, \alpha_2, \cdots, \alpha_k\}$ where $\sum_{i=1}^{k} \alpha_i = 1$, the *entropy* of the distribution P denoted as $H(P)$ is calculated as $-\sum_{i=1}^{k} \alpha_i \log \alpha_i$.

Given a stochastic process $\mathfrak{P}$ which generates the sequence S, denote $H(\mathfrak{P})$ as the *entropy rate* or entropy for short of the stochastic process $\mathfrak{P}$. Recall that $H(\mathfrak{P})$ is defined as $\lim_{n \mapsto \infty} \frac{1}{n} H(X_1, X_2, \cdots, X_n)$, where $H(X_1, X_2, \cdots, X_n)$ stands for the joint entropy of the random variables $X_1, X_2, \cdots, X_n$. It has been shown that when $\mathfrak{P}$ is a stationary process $\lim_{n \mapsto \infty} \frac{1}{n} H(X_1, X_2, \cdots, X_n)$ exists [4].

Theorem 2 (MDL vs. independent decompositio). *Under an ideal encoding, given data with bounded size, the best decomposition which results in the minimum data description length in expectation is an independent decomposition and vice versa.*

Proof. Given a decomposition $C = \{C_1, C_2, \cdots, C_k\}$, for a given n assume that S is a sequence with length n. Under an ideal encoding, the description length of the cluster identifier sequence of S is $|E^*(I(S))| = -\sum_{i=1}^{k} |S(C_i)| \log \alpha_i$. In the ideal encoding, since the length of the compressed subsequence $Z^*(S(C_i))$ is $|Z^*(S(C_i))| = -\log P_i(S(C_i))$ the total description length is:

$$\begin{aligned} L^C(S) &= |E(n)| - \sum_{i=1}^{k} |S(C_i)| \log \alpha_i & (1) \\ & \quad - \sum_{i=1}^{k} \log P_i(S(C_i)) & (2) \end{aligned}$$

$$\begin{aligned} E(L^C(S)) &= \sum_{|S|=n} P(S) * L^C(S) & (3) \\ &= |E(n)| - \sum_{|S|=n} P(S) & (4) \\ & \qquad \log \prod_{i=1}^{k} \alpha_i^{|S(C_i)|} P_i(S(C_i)) & (5) \\ &= |E(n)| + H_P(X_1, X_2, \cdots, X_n) + & (6) \\ & \qquad D(P|Q) & (7) \end{aligned}$$

Where Q is the random mixture of the distributions P_i defined on the space of all sequence $S : |S| = n$, i.e. $Q(S) = \prod_{i=1}^{k} \alpha_i^{|S(C_i)|} P_i(S(C_i))$ and $D(P|Q)$ is the relative entropy or the *Kullback-Leibler* distance between P and Q. Since $D(P|Q) \geq 0$ [4] we can imply that $E(L^C(S)) \geq |E(n)| + H_P(X_1, X_2, \cdots, X_n)$. The equality happens if and only if $D(P|Q) = 0$, i.e. $P \equiv Q$ which proves the theorem. □

5.2 Analysis under the asymptotic encoding

In the ideal analysis, it requires an ideal encoding which is not a practical assumption. However, we can still prove a similar result under an asymptotic encoding. First, we prove a basic supporting lemma. The lemma is a generalized result of the *Cesàro mean* [12].

Lemma 1. *Given a sequence (a_n), a sequence (c_n) is defined as : $c_n = \sum_{i=1}^{n} b_i(n) a_i$ where $\sum_{i=1}^{n} b_i(n) = 1$ and $b_i(n) > 0$ $\forall n > 0$. If $\lim_{n \mapsto \infty} a_n = A$ and $\lim_{n \mapsto \infty} b_i(n) = 0$ $\forall i > 0$ then we also have $\lim_{n \mapsto \infty} c_n = A$.*

Proof. Since $\lim_{n \mapsto \infty} a_n = A$ given any number $\epsilon > 0$ there exists N such that $|a_n - A| < \frac{\epsilon}{2}$ $\forall n > N$. Moreover, because $\lim_{n \mapsto \infty} a_n = A$ there exists an upper bound D on $|a_i - A|$.

Given N, since $\lim_{n \mapsto \infty} b_i(n) = 0$ we can choose M_i $(i = 1, 2, \cdots, N)$ such that $b_i(n) < \frac{\epsilon}{2ND}$ $\forall n > M_i$. Let denote M as the maximum value of the set $\{N, M_1, M_2, \cdots, M_N\}$. For any $n > M$, we have:

$$\begin{aligned} |c_n - A| &= |\sum_{i=1}^{n} b_i(n) a_i - A| & (8) \\ &\leq |\sum_{i=1}^{N-1} b_i(n)(a_i - A)| + & (9) \\ & \qquad |\sum_{i=N}^{n} b_i(n)(a_i - A)| & (10) \\ &\leq (N-1) D \frac{\epsilon}{2ND} + \frac{\epsilon}{2} & (11) \\ &\leq \epsilon & (12) \end{aligned}$$

The last inequality proves the lemma. □

Theorem 3 (Independent decomposition entropy). *Assume that $C = \{C_1, C_2, \cdots, C_k\}$ is an independent decomposition and $\mathfrak{P}_{\mathrm{i}}$ is the stochastic process that generates $S(C_i)$. Denote α_i as the probability that we observe an event belonging to the cluster C_i, we have:*

$$H(\mathfrak{P}) = \sum_{i=1}^{k} \alpha_i H(\mathfrak{P}_{\mathrm{i}}) + H(\alpha_1, \alpha_2, \cdots, \alpha_k) \qquad (13)$$

Proof. We first prove a special case with $k = 2$ from which the general case for any k can be directly implied. Denote $H(X_1, X_2, \cdots, X_n)$ as H for short. In fact, by the definition of the joint entropy we can perform simple calculations as follows:

$$H = -\sum_{|S|=n} P(S) \log P(S) \tag{14}$$

$$= -\sum_{|S|=n} \alpha_1^{|S(C_1)|} P_1(S(C_1)) \alpha_2^{|S(C_2)|} P_2(S(C_2)) \tag{15}$$

$$\log\left(\alpha_1^{|S(C_1)|} P_1(S(C_1)) \alpha_2^{|S(C_2)|} P_2(S(C_2))\right) \tag{16}$$

$$= -C_n^{|S_1|} \sum_{|S_1| \leq n} \sum_{|S_2|=n-|S_1|} \alpha_1^{|S_1|} P_1(S_1) \tag{17}$$

$$\alpha_2^{|S_2|} P_2(S_2) \log\left(\alpha_1^{|S_1|} P_1(S_1) \alpha_2^{|S_2|} P_2(S_2)\right) \tag{18}$$

We denote each term of Equation 18 as follows:

$$X = -C_n^{|S_1|} \sum_{|S_1| \leq n} \sum_{|S_2|=n-|S_1|} \alpha_1^{|S_1|} P_1(S_1) \alpha_2^{|S_2|} \tag{19}$$

$$P_2(S_2) \log \alpha_1^{|S_1|} \tag{20}$$

$$Y = -C_n^{|S_1|} \sum_{|S_1| \leq n} \sum_{|S_2|=n-|S_1|} \alpha_1^{|S_1|} P_1(S_1) \alpha_2^{|S_2|} \tag{21}$$

$$P_2(S_2) \log \alpha_2^{|S_2|} \tag{22}$$

$$Z = -C_n^{|S_1|} \sum_{|S_1| \leq n} \sum_{|S_2|=n-|S_1|} \alpha_1^{|S_1|} P_1(S_1) \alpha_2^{|S_2|} \tag{23}$$

$$P_2(S_2) \log P_1(S_1) \tag{24}$$

$$T = -C_n^{|S_1|} \sum_{|S_1| \leq n} \sum_{|S_2|=n-|S_1|} \alpha_1^{|S_1|} P_1(S_1) \alpha_2^{|S_2|} \tag{25}$$

$$P_2(S_2) \log P_2(S_2) \tag{26}$$

We calculate each term of Equation 18 as follows:

$$X = -\sum_{i=0}^{n} C_n^i \sum_{|S_1|=i} \sum_{|S_2|=n-i} \alpha_1^i \alpha_2^{n-i} \tag{27}$$

$$P_1(S_1) P_2(S_2) \log \alpha_1^i \tag{28}$$

$$= -\sum_{i=0}^{n} C_n^i \alpha_1^i \alpha_2^{n-i} \log \alpha_1^i \tag{29}$$

$$\sum_{|S_1|=i} \sum_{|S_2|=n-i} P_1(S_1) P_2(S_2) \tag{30}$$

$$= -\sum_{i=0}^{n} C_n^i \alpha_1^i \alpha_2^{n-i} \log \alpha_1^i \tag{31}$$

$$= -\log \alpha_1 \sum_{i=0}^{n} i C_n^i \alpha_1^i \alpha_2^{n-i} \tag{32}$$

$$= -n\alpha_1 \log \alpha_1 \tag{33}$$

With similar calculation we have $Y = -n\alpha_2 \log \alpha_2$. We continue with the calculation of Z:

$$Z = -\sum_{i=0}^{n} C_n^i \sum_{|S_1|=i} \sum_{|S_2|=n-i} \alpha_1^i \alpha_2^{n-i} \tag{34}$$

$$P_1(S_1) P_2(S_2) \log P_1(S_1) \tag{35}$$

$$= -\sum_{i=0}^{n} C_n^i \sum_{|S_1|=i} \alpha_1^i \alpha_2^{n-i} P_1(S_1) \log P_1(S_1) \tag{36}$$

$$\sum_{|S_2|=n-i} P_2(S_2) \tag{37}$$

$$= -\sum_{i=0}^{n} C_n^i \sum_{|S_1|=i} \alpha_1^i \alpha_2^{n-i} P_1(S_1) \log P_1(S_1) \tag{38}$$

$$= -\sum_{i=0}^{n} C_n^i \alpha_1^i \alpha_2^{n-i} \sum_{|S_1|=i} P_1(S_1) \log P_1(S_1) \tag{39}$$

$$= \sum_{i=1}^{n} C_n^i \alpha_1^i \alpha_2^{n-i} H_{P_1}(X_1, X_2, \cdots, X_i) \tag{40}$$

With similar calculation we also have:

$T = \sum_{i=1}^{n} C_n^i \alpha_1^{n-i} \alpha_2^i H_{P_2}(X_1, X_2, \cdots, X_i)$

Therefore we further imply that:

$$H = X + Y + Z + T \tag{41}$$

$$= nH(\alpha_1, \alpha_2) + \tag{42}$$

$$\sum_{i=1}^{n} C_n^i \alpha_1^i \alpha_2^{n-i} * H_{P_1}(X_1, X_2, \cdots, X_i) + \tag{43}$$

$$\sum_{i=1}^{n} C_n^i \alpha_1^{n-i} \alpha_2^i H_{P_2}(X_1, X_2, \cdots, X_i) \tag{44}$$

$$\frac{H}{n} = H(\alpha_1, \alpha_2) + \tag{45}$$

$$\alpha_1 \sum_{i=1}^{n} C_{n-1}^{i-1} \alpha_1^{i-1} \alpha_2^{n-i} \frac{1}{i} H_{P_1}(X_1, \cdots, X_i) + \tag{46}$$

$$\alpha_2 \sum_{i=1}^{n} C_{n-1}^{i-1} \alpha_1^{n-i} \alpha_2^{i-1} \frac{1}{i} H_{P_2}(X_1, X_2, \cdots, X_i) \tag{47}$$

Besides, we have:

$\lim_{n \mapsto \infty} \frac{1}{n} H(X_1, \cdots, X_n) = H(\mathfrak{P})$

$\lim_{n \mapsto \infty} \frac{1}{n} H_{P_1}(X_1, \cdots, X_n) = H(\mathfrak{P}_1)$

$\lim_{n \mapsto \infty} \frac{1}{n} H_{P_2}(X_1, \cdots, X_n) = H(\mathfrak{P}_2)$

Therefore, according to Lemma 1 from the last equation we can imply that $H(\mathfrak{P}) = \alpha_1 H(\mathfrak{P}_1) + \alpha_2 H(\mathfrak{P}_2) + H(\alpha_1, \alpha_2)$.

The last result can be easily generalized for an independent decomposition with any k clusters by induction. Indeed, we assume that the theorem is correct with $k = l$ we prove that the result holds for $k = l + 1$. Denote α as $\sum_{i=1}^{l} \alpha_i$. Given two independent stochastic processes $\mathfrak{P}$ and $\mathfrak{Q}$ denote the random mixture of them as $\mathfrak{P} \bigoplus \mathfrak{Q}$. Consider the process defined as the random mixture of $\mathfrak{P}_1, \mathfrak{P}_2 \cdots \mathfrak{P}_l$ denoted as $\mathfrak{P}_1 \bigoplus \mathfrak{P}_2 \bigoplus \cdots \bigoplus \mathfrak{P}_l$. Since $\mathfrak{P}_{l+1}$ and the sequence $\mathfrak{P}_{\mathfrak{1}} \bigoplus \mathfrak{P}_{\mathfrak{2}} \bigoplus \cdots \bigoplus \mathfrak{P}_{\mathfrak{l}}$ are independent we have:

$$\begin{aligned} H(\mathfrak{P}) &= H(\mathfrak{P}_{\mathfrak{1}} \bigoplus \mathfrak{P}_{\mathfrak{2}} \bigoplus \cdots \bigoplus \mathfrak{P}_{\mathfrak{l+1}}) && (48) \\ &= \alpha H(\mathfrak{P}_{\mathfrak{1}} \bigoplus \mathfrak{P}_{\mathfrak{2}} \bigoplus \cdots \bigoplus \mathfrak{P}_{\mathfrak{l}}) + && (49) \\ & \quad \alpha_{l+1} H(\mathfrak{P}_{\mathfrak{l+1}}) + H(\alpha, \alpha_{l+1}) && (50) \end{aligned}$$

Moreover, by the induction assumption:

$$\begin{aligned} H(\mathfrak{P}_{\mathfrak{1}} \bigoplus \mathfrak{P}_{\mathfrak{2}} \bigoplus \cdots \bigoplus \mathfrak{P}_{\mathfrak{l}}) &= \sum_{i=1}^{k} \frac{\alpha_i}{\alpha} H(\mathfrak{P}_{\mathfrak{i}}) + && (51) \\ & \quad H(\frac{\alpha_1}{\alpha}, \frac{\alpha_2}{\alpha}, \cdots, \frac{\alpha_l}{\alpha}) && (52) \end{aligned}$$

Replacing this value to Equation 50, we can obtain Equation 13 from which the theorem is proved. □

Theorem 3 shows that the entropy of the stochastic process $\mathfrak{P}$ can be represented as the sum of two meaningful terms. The first term $H(\alpha_1, \alpha_2, \cdots, \alpha_k)$ actually corresponds the average cost per element of the cluster identifier. Meanwhile the second term $\sum_{i=1}^{k} \alpha_i H(\mathfrak{P}_{\mathfrak{i}})$ corresponds to the average cost per element to encode the subsequences $S(C_i)$. By that important observation we can show the following asymptotic result:

Theorem 4 (Asymptotic result). *Under an asymptotical encoding, the data description length in an independent decomposition is asymptotically optimal with probability equal to 1.*

Proof. Let $C = \{C_1, C_2, \cdots, C_k\}$ be an independent decomposition, for any n assume that S is a sequence with length n. Under an asymptotic encoding, the description length of the data is:

$$L^C(S) = |E(n)| + |E^+(I(S))| + \sum_{i=1}^{k} Z^+(S(C_i)) \qquad (53)$$

$$\frac{L^C(S)}{|S|} = \frac{|E(n)|}{|S|} + \frac{|E^+(I(S))|}{|S|} + \frac{\sum_{i=1}^{k} Z^+(S(C_i))}{|S|} \qquad (54)$$

$$\frac{L^C(S)}{|S|} = \frac{|E(n)|}{|S|} + \frac{|E^+(I(S))|}{|S|} + \sum_{i=1}^{k} \frac{|S(C_i)|}{|S|} \frac{Z^+(S(C_i))}{|S(C_i)|} \quad (55)$$

$$Pr\left(\lim_{|S| \mapsto \infty} \frac{L^C(S)}{|S|} = H(\alpha_1, \alpha_2, \cdots, \alpha_k) + \sum_{i=1}^{k} \alpha_i H(\mathfrak{P}_{\mathfrak{i}})\right) = 1 \quad (56)$$

$$Pr\left(\lim_{|S| \mapsto \infty} \frac{L^C(S)}{|S|} = H(\mathfrak{P})\right) = 1 \qquad (57)$$

The last equation is a direct result of Theorem 3. Since $H(\mathfrak{P})$ is the lower-bound on the expectation of the average compression size per element of any data compression algorithm the encoding using the independent decomposition is asymptotically optimal. □

The ideal analysis shows the one-to-one correspondence between the optimal encoding and an independent decomposition. The asymptotic result only shows that an independent decomposition asymptotically corresponds to an optimal encoding. The theorem does not prove the reverse correspondence; however, in experiments we empirically show that the correspondence is one-to-one.

Algorithm 1 Dzip(S)

```
Input: a sequence $S$, an alphabet $\sum = \{a_1 a_2 \cdots a_N\}$
Output: a decomposition $C$
$C \leftarrow \{C_1 = \{a_1\}, C_2 = \{a_2\}, \cdots, C_n = \{a_n\}\}$
while true do
    $max \leftarrow 0$
    $C^* \leftarrow C$
    for $i = 1$ to $|C|$ do
        for $j = i + 1$ to $|C|$ do
            $C^+ \leftarrow C$ with merged $C_i$ and $C_j$
            if $L^C(S) - L^{C^+}(S) > max$ then
                $max \leftarrow L^C(S) - L^{C^+}(S)$
                $C^* \leftarrow C^+$
            end if
        end for
    end for
    if $|C^*| = 1$ or $max = 0$ then
        Return $C^*$
    end if
end while
```

6. ALGORITHMS

The theoretical analysis in Section 5 encourages us to design an algorithm that looks for the best decomposition to find an independent decomposition. When an independent decomposition is found, the algorithm can be recursively repeated on each independent component to find the maximum independent decomposition. Given data S with alphabet $\sum$ this section discusses a hierarchical clustering algorithm called Dzip to find the desired decomposition.

Algorithm 1 shows the main steps of the Dzip algorithm. It starts with N clusters each contains only one character of the alphabet. Subsequently, it evaluates the compression benefit of merging any pair of clusters. The best pair of clusters which results in the smallest compression size is chosen to be merged together. These steps are repeated until there is no compression benefit of merging any two clusters or all the characters are already merged into a single cluster.

Dzip can be recursively applied on each cluster to get the maximum decomposition. However, in our experiment we observe that in most of the cases the cluster found by Dzip cannot be decomposed further because of the bottom-up process which already checks for the benefits of splitting the cluster.

Dzip uses the *Lempel-Ziv-Welch* (LZW) implementation [4] with complexity linear in the size of the data. It utilizes an inverted list data structure to store the list of positions of each character in the sequence. Moreover, it also caches compression size of merged clusters. In doing so, in the worst case the computational complexity of Dzip can be bounded as $O(|S|N^2)$. This number is the same as the amortized complexity of the Dtest algorithm [3].

7. EXPERIMENTS

We consider the dependency test method Dtest [3] as a baseline approach. All the experiments were carried out on a 16 processor cores, 2 Ghz, 12 GB memory, 194 GB local disk, Fedora 14 / 64-bit machine. The source codes of Dtest and Dzip in Java and the datasets are available for download in our project website[1].

Dtest has two parameters: the significance value α and the gap number G. We choose $\alpha = 0.01$ and $G = 300$ as recommended in the paper [3]. We also tried to vary α and G from small to large and observed quite different results. The algorithm is slow when G increases, while smaller value of α results in low false positive yet high false negative rate and vice versa. However, the results with different parameters do not change the comparisons in our experiments.

7.1 Synthetic data

There are three datasets in this category for which the ground truths are known:

- *Parallel:* is a synthetic dataset which mimics a typical situation in practice where the data stream is generated by five independent parallel processes. Each process P_i $(i = 1, 2, \cdots, 5)$ generates one event from the set of events $\{A_i, B_i, C_i, D_i, E_i\}$ in that order. In each step, the generator chooses one of five processes uniformly at random and generates an event by using that process until the stream length is 1M.

- *Noise:* is generated in the same way as the parallel dataset but with additional noises. A noise source generates independent events from a noise alphabet with size 1000. Noise events are randomly mixed with parallel dataset. The amount of noises is 20% of the parallel data. This dataset is considered to see how the methods will be sensitive to noise.

- *HMM:* is generated by a random mixture of two different hidden Markov models. Each hidden Markov model has 10 hidden states and 5 observed states. The transition matrix and the emission matrix are generated randomly according to the standard normal distribution with mean in the diagonal of the matrix. Each Markov model generates 5000 events and the mixture of them contains 10000 events. This dataset is considered to see the performance of the methods in a small dataset.

Since the ground-truths are known we use the *Rand index* [13] to compare two partitions of the alphabet set. The rand index calculates the number of pairs of events that are either in the same cluster in both partitions or in different clusters in both partitions. This number is normalized to have value the interval $[0, 1]$ by dividing by the total number of different pairs. The rand index measures the agreement between two different partitions, where a value of 1 means a perfect match, while 0 means two partitions completely disagree to each other.

Figure 2 shows rand index (y-axis) of two algorithms when the data size (x-axis) is varied. It is clear from the figure that the Dzip algorithm is possible to return a perfect decomposition in all datasets. When the data size is smaller the performance is slightly changed but the rand index is still high.

The performance of the Dtest algorithm is good in the Parallel dataset although the result is not stable when the datasize varied. However, Dtest does not work well in the Noise and the HMM dataset especially when a lot of noises are added. In both datasets, Dtest seems to cluster every events together; this experiment confirms our discussion in section 2 that Dtest is vulnerable to noise.

7.2 Real-life data

There are three datasets in this category:

- *Machine:* is a message log containing about 2.7 million of messages (more than 1700 distinct message types) produced by different components of a photolithography machine.

- *MasterPortal:* is a historical log of user behaviors in the *MasterPortal*[2] website. It contains about 1.7M of events totally of 16 different types of behaviors such as *Program view, University view, Scholarship view, Basic search, Click on ads banner* and so on.

- *Msnbc:* is the clickstream log by the users of the MSNBC website[3]. The log contains 4.6M events of 16 different types each corresponds to a category of the website such as *frontpage, sports, news, technology, weather* and so on.

For the Machine dataset, the Dzip algorithm produced 11 clusters with three major clusters contains a lots of events. Meanwhile, the Dtest algorithm produced 4 clusters with one very big cluster and 3 outlier clusters each contains only one event. The result shows that Dtest seems to cluster every events together. Since the ground truths are unknown we compare the compression ratios when using the decomposition by each algorithm to compress the data. Our observation shows that the data is compressed better with the decomposition produced by the Dzip algorithm because the compression ratio is 2.37 on the Dzip algorithm versus 2.34 on the Dtest algorithm.

Both Dzip and Dtest produced one cluster of events for the Msnbc and the MasterPortal datasets. Therefore, the compression ratios of both algorithms are the same in each dataset. Although we don't know the ground-truths for these datasets the result seems to be reasonable because in the case of clickstream data, users traverse on the web graphs and the relations between the events are inherently induced from the connections in the web graph structure. The Msnbc is compressed better than the MasterPortal (the compression ratios are 1.5 and 1.2 respectively). These numbers also tell us that the dependency between events in the Msnbc dataset seems to be more regular.

7.3 Running time

In section 6, we have shown that the amortized complexity of the Dtest algorithm and the worst case complexity of the Dzip algorithm are the same. The result promises that the Dzip algorithm will be faster than the Dtest algorithm. Indeed, this fact holds for the set of datasets we use in this paper. In Figure 3 we compare Dzip and Dtest in terms of running time. In most datasets, Dzip is about an order of magnitude faster than the Dtest algorithm.

8. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed a compression-based method called Dzip for the sequence independent decomposition problem. Beside being justified by a theoretical analysis, in experiments with both synthetic and real-life datasets, Dzip

[1] www.win.tue.nl/~lamthuy/dzip.htm

[2] www.mastersportal.eu

[3] www.msnbc.com

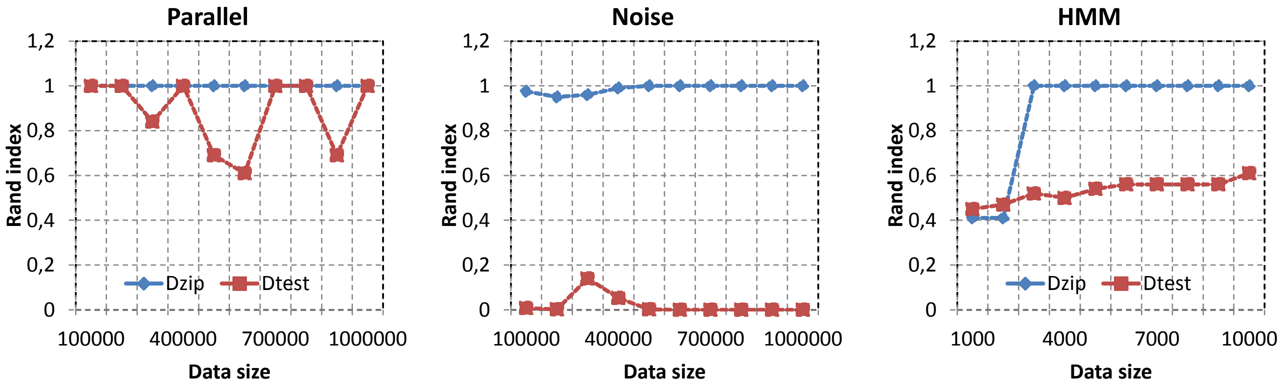
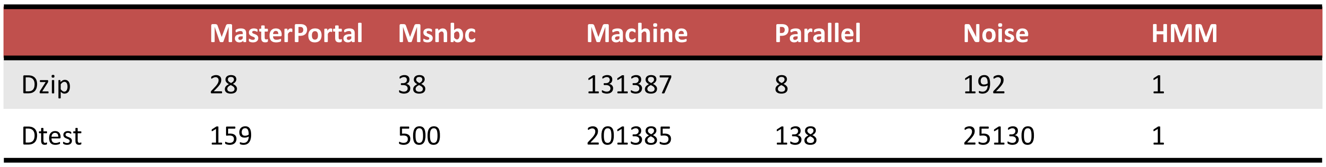


Figure 2: Rand index measures the similarity (higher is better) between the decompositions by the algorithms and the ground-truths. Rand index is equal to 1 if it is a perfect decomposition.



	MasterPortal	Msnbc	Machine	Parallel	Noise	HMM
Dzip	28	38	131387	8	192	1
Dtest	159	500	201385	138	25130	1

Figure 3: Running time in seconds of two algorithms. Dzip is about an order of magnitude faster than Dtest.

was shown to be more effective than the state of the art method based on statistical hypothesis testing. There are various directions to extend the paper for the future works. At the moment, we assume that each independent process must produce a disjoint subset of events. In practice, the case that independent processes produce overlapping subset of events is not rare. Extending the work to this more general case can be considered as an interesting future work.

9. ACKNOWLEDGEMENTS

The work is part of the project *Mining Complex Patterns in Stream* (COMPASS) supported by the *Netherlands Organisation for Scientific Research* (NWO).

10. REFERENCES

[1] Kira Radinsky, Eric Horvitz: Mining the web to predict future events. WSDM 2013: 255-264

[2] Julia Kiseleva, Hoang Thanh Lam, Toon Calders and Mykola Pechenizkiy: Discovery temporal hidden contexts in web sessions for user trail prediction TempWeb workshop at WWW 2013.

[3] Heikki Mannila, Dmitry Rusakov: Decomposition of event sequences into independent components. SDM 2001

[4] Thomas Cover, Joy Thomas: Elements of information theory. Wiley and Son, second edition 2006.

[5] Peter Grünwald: The minimum description length principle. MIT press, 2007.

[6] Hyvärinen A, Oja E. Independent component analysis: algorithms and applications. Journal of Neural Network 2000.

[7] Michael Mampaey, Jilles Vreeken: Summarizing categorical data by clustering attributes. Data Min. Knowl. Discov. 2013

[8] Rudi Cilibrasi, Paul M. B. Vitányi: Clustering by compression. IEEE Transactions on Information Theory 2005

[9] Eamonn J. Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana, Li Wei, Sang-Hee Lee, John Handley: Compression-based data mining of sequential data. Data Min. Knowl. Discov. 2007

[10] Jilles Vreeken, Matthijs van Leeuwen, Arno Siebes: Krimp: mining itemsets that compress. Data Min. Knowl. Discov. 2011

[11] Hoang Thanh Lam, Fabian Moerchen, Dmitriy Fradkin, Toon Calders: Mining Compressing Sequential Patterns. SDM 2012: 319-330

[12] Hardy, G. H. (1992). Divergent Series. Providence: American Mathematical Society. ISBN 978-0-8218-2649-2.

[13] W. M. Rand (1971). Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66 (336).

[14] van der Aalst, W. M. P., Weijters, T. and Maruster, L. (2004). Workflow Mining: Discovering Process Models from Event Logs.. IEEE Trans. Knowl. Data Eng., 16, 1128-1142.

Interactive Visualization Applications for Maritime Anomaly Detection and Analysis

Valérie Lavigne
Defence R&D Canada
2459 PIE-XI Nord, Bat 24
Québec, Qc, Canada, G3J 1X5
Tel: 1 (418) 844-4000
valerie.lavigne@drdc-rddc.gc.ca

ABSTRACT

A study of maritime surveillance operations revealed that visual analytics could enable better maritime situation analysis. For that purpose, we designed the Maritime Visual Analytics Prototype, which is detailed in this demo paper. It supports the detection of marine anomalies and the detailed analysis of vessels of interest through a series of specialized tools. First, the Analysis Set Manager acts as the central repository and starting point for tools launching. The Animated Map and Timeline enable visual anomaly detection related to vessel tracks using Route Ribbons and Close Encounter Icon visualizations added to an interactive geo-temporal display. The Visual Summary Cards presented in the Record Browser display the key vessel characteristics for rapid visual scanning. The Magnets Grid enables a multi-dimensional exploration of factual vessel information, while temporal analysis is performed using the Multi-Timelines. This prototype was tested with operational maritime surveillance data and evaluated through user jury trials with real potential users. Comments from the users indicate that the visual widgets proposed could be valuable to their daily operations.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *graphical user interfaces (GUI), user-centered design.*

General Terms

Management, Design, Experimentation, Security, Human Factors.

Keywords

Visual analytics, anomaly detection, situation analysis, maritime domain awareness, user jury validation.

1. INTRODUCTION

Maritime domain awareness is defined as "the effective understanding of everything on, under, related to, adjacent to or bordering a sea, ocean or other navigable waterway, including all maritime-related activities, infrastructure, people, cargo, and vessels and other conveyances that could impact the security, safety, economy, or environment" [1]. In Canada, ensuring coastal safety by detecting marine threats in a sea of vessel track data falls within the responsibility of the Coastal Marine Security Operation Centres (MSOC), where the staffs gather and analyze information, and produce specialized intelligence products to support operational decision makers during routine and contingency operations [2]. This mandate exceeds awareness and involves focused analysis. To build and maintain a shared understanding of the maritime situation is very challenging and can lead to significant cognitive overload. Visual Analytics (VA) technologies could be beneficial to that endeavor.

In this paper, we present a study of VA applied to maritime domain challenges, including the requirements identified for a maritime VA tool suite, the Maritime Visual Analytics Prototype (MVAP) that we designed and implemented, and the results of the validation activity that followed.

1.1 Application of Visual Analytics for Maritime Domain Analysis

In this project, we conducted a detailed study investigating how VA can benefit maritime domain analysis. We first performed multiple knowledge elicitation sessions to identify important challenges and visual requirements related to maritime domain analysis. We decided to focus our research on the two following tasks: maritime anomaly detection and vessel of interest analysis. In our design, we split the visualization needs into a series of VA tools using a modular approach. That led to the development of the MVAP which is the subject of this paper.

1.2 Validation with Target Users

The MVAP was evaluated by real potential users from each of the five federal departments involved in the MSOCs and representing a variety of maritime security functions. A hands-on group training session was followed by a series of individual tasks to perform. The MVAP and the tools that comprise it were assessed using Standard Usability Scale surveys, user rankings of individual tools, observations and interviews. Moreover, the MVAP was tested with real surveillance data which contributed to showing the users its operational potential.

Validation results and comments from the users indicate that the visual widgets proposed could indeed be valuable to their daily operations. This activity also generated insights useful for VA tool design and assessment.

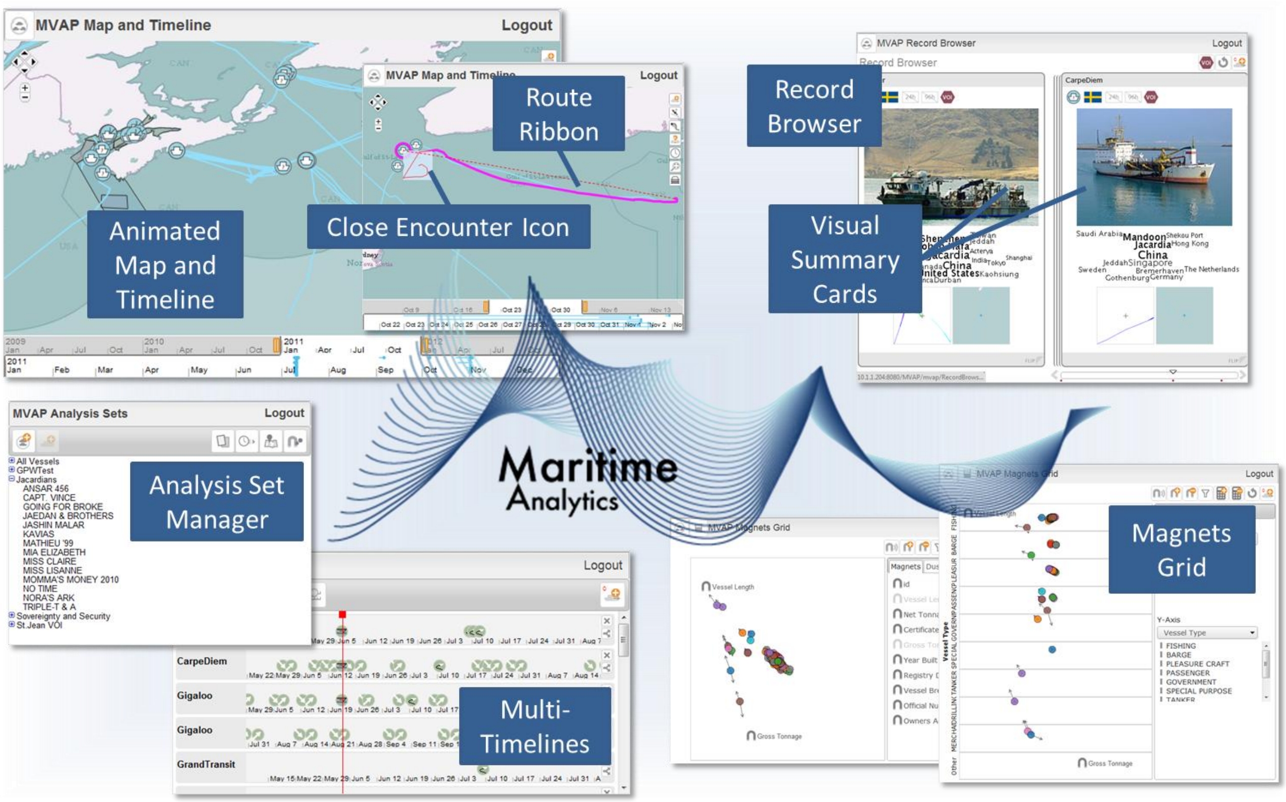


Figure 1. This is an overview of the Maritime Visual Analytics Prototype tools for visual marine threat identification and analysis.

2. REQUIREMENTS ANALYSIS

“The MSOCs provide comprehensive marine domain awareness along Canada’s coasts enabling detection, assessment, and response to threats that could adversely affect the safety, security, environment or economy of Canada. Threats include foreign trans-national organized crime - drug trafficking, piracy, migrant smuggling - emerging terrorist activity, over-fishing, and polluters” [2]. To do that, the MSOCs perform a 24/7 watch over Canada’s three oceans. Heterogeneous sources provide information related to the thousands of vessels that can cross their area of responsibility.

After a review of previous knowledge elicitation studies from related DRDC research projects [3, 4], we conducted requirements analysis sessions with interagency civilian and military staff involved in the MSOCs. We were also able to observe duty personnel in action in their work environment. The domain knowledge was gathered through a mix of interviews, observations, and group discussions. We identified a series of tasks that could benefit from VA tools [5], but chose to focus our limited research resources on the two activities which presented the highest improvement potential from the VA science and technology: identifying anomalies related to vessel tracks and information, and performing a focused analysis of a situation that involves a vessel of interest.

2.1 Anomaly Detection

The MSOCs do not have enough manpower to conduct a full analysis of every vessel in their areas of responsibility, so they use triage to identify vessels that may present a threat and require detailed analysis. Unfortunately, vessels with anomalous behaviours or suspicious connections are not easily detected among the very large number of vessels going about normal, legitimate activities.

They need the ability to recognize outliers that do not behave as expected as well as the ability to spot individuals who behave in a way similar to previously identified threats.

2.2 Vessel of Interest Analysis

Vessels of Interest (VOIs) are those vessels that require special attention. They are not declared lightly and a procedure must be followed. When a vessel is designated as a VOI, a number of people will gather information and conduct detailed analysis in order to make a judgment about the case. This detailed analysis can also lead to the identification of other potentially interesting individuals.

2.3 Information to Represent

To detect maritime anomalies and to represent what can be known about a VOI, three main categories of information must be considered [5]:

- Geo-temporal: mainly the vessel’s track, including related data such as speed
- Temporal: significant events involving the vessel
- Factual: known information elements about the vessel, related people or organizations, and physical properties.

2.4 Other Considerations

Additionally, we must consider the constraints that come from working in a defence and security environment like the MSOCs. Frequent staff changes are expected and the time that can be devoted to training is very limited. Thus, the proposed tools should be easy to learn and provide an intuitive interface.

The Canadian Charter of Rights and Freedoms, and the Privacy Act must be taken into account when collecting, using and sharing information. For that reason, the data that different governmental agencies collect are not shared directly in a common data repository. Tools should allow users to import/export the data from/to different formats to facilitate sharing when working on a case with other agencies.

3. LITERATURE REVIEW

3.1 Anomaly Detection

3.1.1 Automated Detection

In the maritime domain, many researchers attempted to detect anomalous vessel behaviours automatically relying on models of normal/abnormal vessel kinetic behaviour.

Laxhammar [6] uses a Gaussian mixture model for maritime anomaly detection while Johansson and Falkman [7] use a Bayesian network. Spline-based trajectory clustering techniques were proposed by Dahlbom and Niklasson [8] to represent normal vessel behaviour for coastal surveillance. Rhodes et al [9, 10] suggest the use of a neurobiologically inspired algorithm for probabilistic associative learning of vessel motion.

Riveiro et al [11, 12] opt for self-organizing maps but allow user involvement in the anomaly detection by providing interactive visualizations and a data mining module that supports the insertion of the user's knowledge and experience.

An automated approach for anomalous vessel behaviour detection was also researched at DRDC, employing a rule-based expert system which allowed an operator to express anomaly rules that take advantage of both kinetic and non-kinematic vessel properties [13]. A similar approach was also employed by Edlund et al [14].

These automated approaches produce very good results for suspicious patterns that were previously noticed and can be clearly expressed. However, they do not allow for first time discovery of new patterns, so a visual anomaly detection approach could very well complement automated systems.

3.1.2 Visual Detection

Extensive work can be found regarding the visual analysis of trajectories, but they often focus on identifying larger trends in the data, not in detecting outliers.

Willems et al [15] produced ship density landscapes in which ships off historic routes and regular traffic lanes visually stand out. Vessel movement patterns can also be characterized using hybrid fractal/velocity signatures [16] to recognize anomalous activities. TripVista [17] offers spatial, temporal and multiple-dimensional perspectives to analyze micro terrestrial traffic data for finding regular patterns and anomalies of traffic flows. These projects address several types of trajectory anomalies but ignore abnormal situations that can be detected using non-kinetic data.

Most of the VA literature concerning anomaly detection is concerned with network security and the techniques employed are not easily transferable to the maritime domain where the geo-temporal aspect is a central analysis component.

3.2 Vessel of Interest Analysis

Multi-dimensional visualization of aspects of the traffic trajectory data with parallel coordinates plot was proposed in TripVista [17] and [18] (adding physical ship properties). As for anomaly detection, the focus of current research remains on trajectory data. Although the geospatial trajectory of a vessel is the most salient signature, maritime situation assessment requires the analysis of more varied data such as port visit history, owner relationships and suspected criminal activities.

Not focusing specifically on maritime situations, Keim [19] outlines the advantages of visualization for data mining applications and gives a long list of examples. Perer and Shneiderman [20] also discuss the tight link between VA and data mining. Although visual data mining often helps identifying global trends, the visual analysis of individual entities has received much less attention.

4. THE MARITIME VISUAL ANALYTICS PROTOTYPE

The MVAP was developed to explore the potential of VA techniques for visual anomaly detection and situation analysis in the maritime domain.

4.1 Our Approach

Early in our design process, we decided to adopt a modular approach, where each of the proposed tools would be independent and focus on a specific analysis perspective. We designed a series of concepts [21], presented them to potential users and selected the most promising ones to be included in MVAP.

The map rapidly appeared as the natural choice for representing the vessel track information, with an important focus on combining the geographical and temporal aspects together. We added a few innovative concepts in an attempt to make some types of trajectory anomalies more salient. As temporal information is very important for pattern detection, we created a specialized widget that focusses on temporal patterns detection. Factual information exploration was divided between multi-dimensional simultaneous exploration and visual summaries for quick scan and comparison.

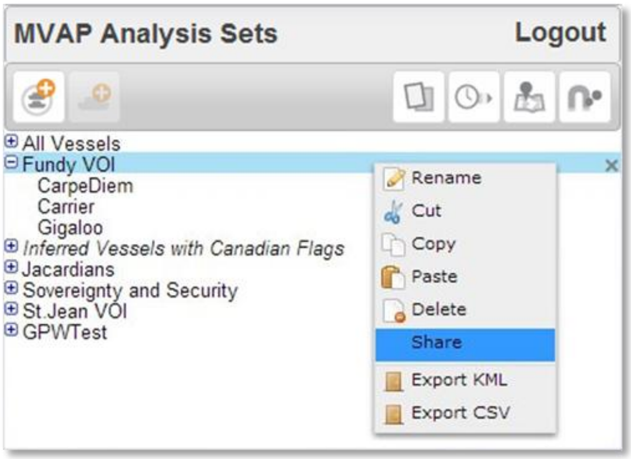
The strength of splitting the different functionalities across specialized widgets is that it avoids imposing a steep learning curve to a user who only needs a fraction of the MVAP capabilities. We expect simple single purpose tools will also require less training than a complex application offering all these features together. On the other hand, a potential drawback is that it may result in a fractioned analysis where the overall picture is hard to grasp.

This decoupled approach also facilitates the eventual operational deployment of the tools by not imposing a large application to end users who may only be interested in one widget. An added benefit is to facilitate the reuse of some of the tools for other application domains, such as social network analysis in a counter-insurgency context [22].

The prototype interface is implemented in JavaScript and runs in a web browser with a service oriented architecture backend. The following subsections describe the functionalities that each MVAP widget offers. Figure 1 provides an overview of all the MVAP tools.

4.2 Analysis Set Manager

The Analysis Set Manager serves as a central repository for the analysis sets that are built using the other tools of the MVAP. It features a hierarchical tree of analysis sets listing all the vessels that they contain (Figure 2). With the Analysis Set Manager, the user can organize objects into meaningful analysis groups. For example, a list of vessels that need to be monitored closely could be managed from the Analysis Set Manager. The visual encoding for managing the sets of vessels mimics the use of a file browser, providing a familiar concept to most computer users.



MVAP Analysis Sets Logout

- All Vessels
- Fundy VOI
  - CarpeDiem
  - Carrier
  - Gigaloo
- *Inferred Vessels with Canadian Flags*
- Jacardians
- Sovereignty and Security
- St.Jean VOI
- GPWTest

Figure 2. The Analysis Set Manager is the main repository and launching point for the other VA tools. It features a hierarchical display of vessel sets. Sets displayed in italic were generated using artificial reasoning services.

Analysts are not expected to create all the vessels lists manually. Partial automation of the process is performed by taking advantage of artificial reasoning services to automatically create “smart sets” according to predefined rules. This leverages the previous work performed with expert systems that can detect a large number of anomalous situations automatically [13], effectively combining our interactive visualization approach with automated data mining. These smart sets can be regularly updated and are displayed in italic to differentiate them from sets resulting from manual selection or visual analysis.

The Analysis Set Manager is also the launching point for the other VA widgets, which allow operators to identify more anomalies and patterns, as well as to explore the details of the reported anomalies from the data mining process.

4.3 Animated Map and Timeline with Visual Encodings

The use of a geographical map is central to maritime situation analysis. In the MVAP, this essential capability is provided by the Animated Map and Timeline (Figure 3), which add temporal animation to the map visualizations currently available in operational systems. It contains a geographical display with a timeline added below that lets the user select the active time interval for data display. The top part of the timeline contains an overview of the selected time interval and the bottom part is a zoomed in version. The selected interval can be dragged to animate the vessel tracks on the map, in order to perform spatiotemporal analysis

Two innovative visual representations are integrated into the map display: the Close Encounter Icons and the Route Ribbons (Figure 4). Their purpose is to facilitate visual detection of track anomalies, including encounters and vessels not taking the shortest route to their stated destination.

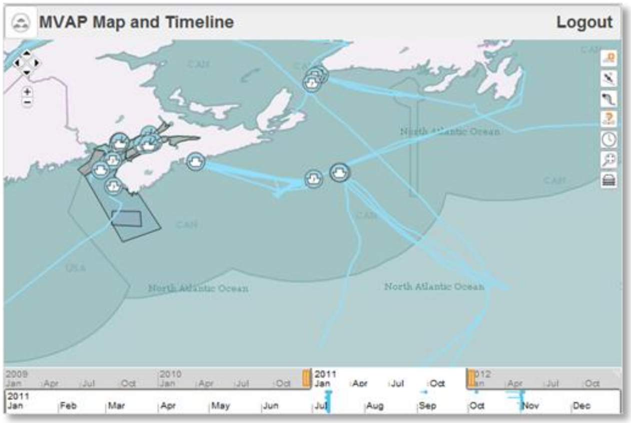


Figure 3. The Animated Map and Timeline offers an interactive display of vessels tracks where the user can select a time interval at the bottom of the interface and drag it to animate the tracks backward or forward.

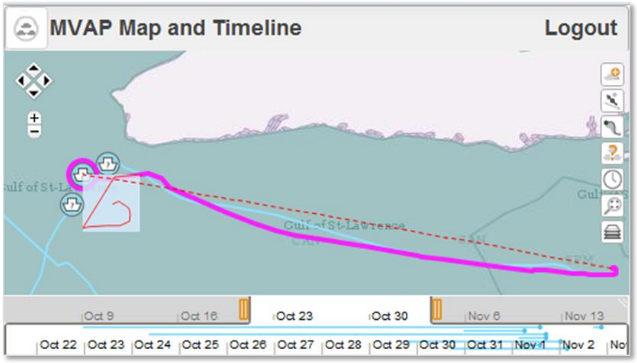
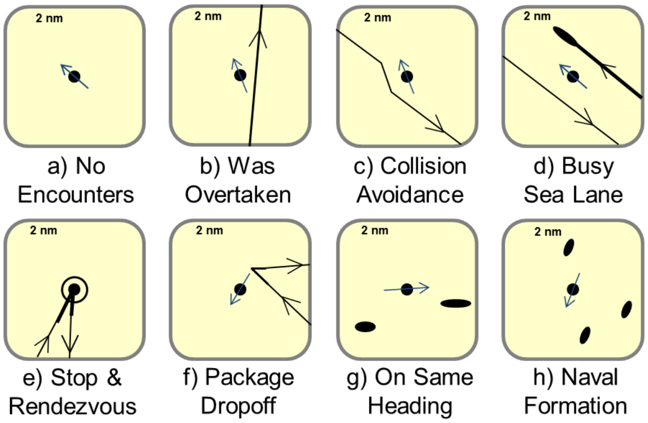


Figure 4. Route Ribbons and Close Encounter Icons appear when a vessel is selected or can be turned on for all vessels. Route Ribbons increase the saliency of vessel tracks following unexpected routes.

A Close Encounter Icon is a square area that is centred on a vessel. This icon follows the vessel during its journey and other vessels that come near it will leave a trace within the icon as they cross it. Figure 5 highlights possible icon patterns that can happen along with their associated meanings. This icon summarizes a vessel’s journey and shows in a single glance whether there were close encounters, helping the analysts triage which tracks to animate for further analysis. The Route Ribbons trace a dotted line representing the expected route for the vessel for comparison against the actual track, resulting in increased saliency for anomalous tracks.

The Animated Map and Timeline widget also serves as a spatio-temporal filtering tool for creating meaningful sets of vessels to analyse with the other MVAP tools.



a) No Encounters
b) Was Overtaken
c) Collision Avoidance
d) Busy Sea Lane
e) Stop & Rendezvous
f) Package Dropoff
g) On Same Heading
h) Naval Formation

Figure 5. The Close Encounter Icon is a visual representation centered on a vessel and moving with it, where other vessels crossing its path leave a trace inside this square area. Here are a few potential examples with their associated meanings [23].

4.3.1 Modified Version for Handling Real Operational Data

A specialized version of the Animated Map and Timeline widget was created to handle the large amount of tracks present in real data. We ingested a 40 GB operational dataset of 112 million position reports spanning over a year, related to 75 000 vessels.

A benefit of creating this widget version is that it contributed to gaining credibility in the eyes of the target users and getting their attention. Real data also tested the limits of the system with regards to dealing with large datasets.

A few features were added to that version, extending its capabilities with automatic detection of loiterers and vessel encounters within a selected region and time. This detection preprocessing and database insertion can take about two days on a standard personal laptop. In an operational setting, it is more likely that we would not process a year's worth of tracks at once but rather analyze the tracks regularly as the information is gathered.

4.4 Magnets Grid

The inspiration for the Magnets Grid was the Dust&Magnets concept [24], which was designed to explore a multi-dimensional space of attributes. This VA tool favours the understanding a maritime situation beyond the exploration of traditional kinematic vessel properties. Interaction with the Magnets Grid can help identify trends and outliers in sets of hundreds of vessels according to multiple properties at once. In our demonstration scenario, vessels associated to a fictive suspicious country called "Jacardia" could be detected with the Magnets Grid using magnets that attracted vessels based on the number of references to Jacardian people and places contained in their profile.

The canvas space is filled with dots representing individual vessels, called the dust. Labelled magnets corresponding to vessel properties can be inserted into the canvas. Clicking the shake button will make the dust move according to the vessels' property values, as depicted in Figure 6. There is no limit to the number of magnets that can be used together, making this tool well fitted to explore multiple dimensions at once. New magnets can be created on the fly by selecting attributes in the vessel record.

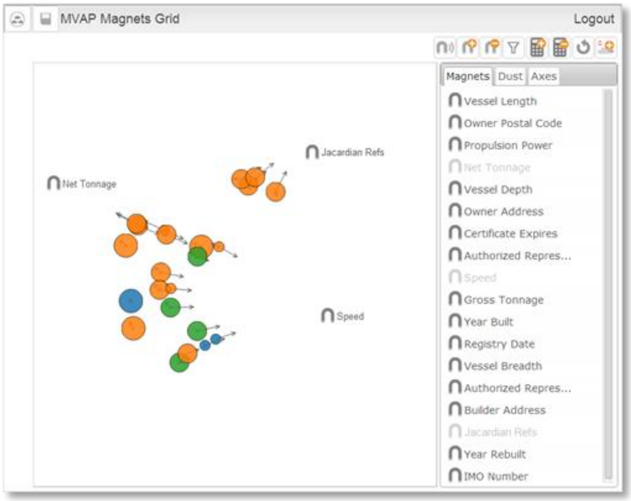
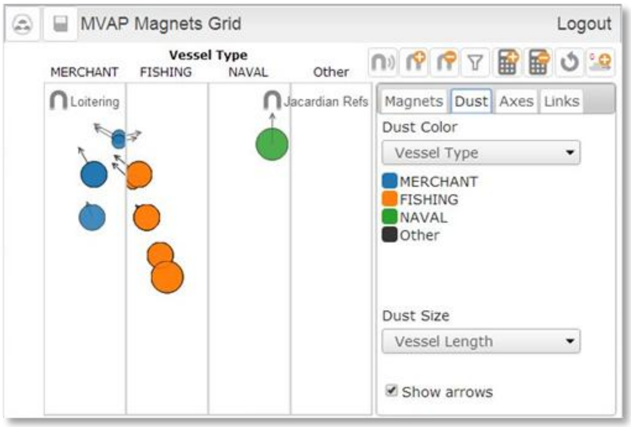

MVAP Magnets Grid

Figure 6. In Magnets Grid, the vessels are displayed as dots and magnets are dropped in the canvas to attract them based on vessel property values. The dots color and size can be associated to vessel properties. Here, the "Jacardian Refs" magnet reveals four outliers that are involved with this fictive suspicious country in our demonstration scenario.

We expanded the Dust&Magnets display with attraction arrows around dust elements to provide a visual cue indicating the attraction strength of the magnets. The arrows reduce the attraction ambiguity when a static snapshot of the Magnets Grid is captured.

We also augmented the tool with scatter plot capabilities and the possibility to constrain the dust movement to vertical or horizontal bands (or both). In Figure 7, associating the Vessel Type to the X-Axis will prevent the dots from leaving the bands to which they belong. This can enable insights about trends across the different categories represented by the bands. If no magnets are added to the canvas, associating the x-axis and y-axis to attributes will enable the use of Magnets Grid as a scatter plot tool (Figure 8).


MVAP Magnets Grid

Dust Color: Vessel Type

🟦 MERCHANT
🟧 FISHING
🟩 NAVAL
⬛ Other

Dust Size: Vessel Length

☑ Show arrows

Figure 7. In Magnets Grid, the dots movement towards the magnets can be constrained by associating vessel properties to either the horizontal or vertical axis, or both. In this example, the vessel dots cannot leave the columns where they belong.

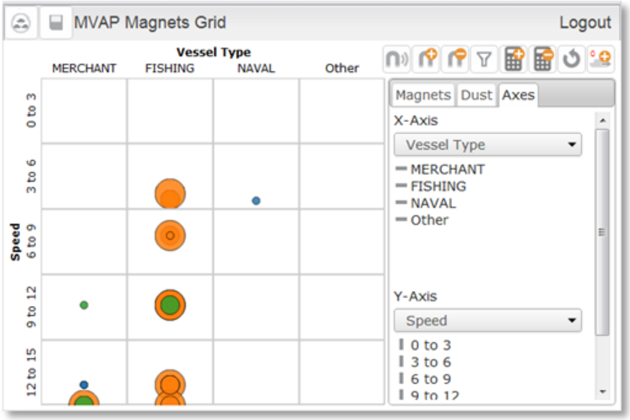


Figure 8. If we remove the magnets and select properties for both axes in Magnets Grid, we obtain a scatterplot display.

4.5 Visual Summary Cards and Record Browser

The intent of the Visual Summary Cards is to communicate the key characteristics of individual vessels in a concise visual format. The flip side of Visual Summary Cards also provides the factual textual information about the vessels. Information is formatted so that the analyst can look for normally present or absent elements rather than having to read each card. When available, a picture of the vessel is provided for visual identification. The top part of the cards contains icons for the vessel type, flag, as well as 24h and 96h call reports status. The word cloud contains the ports that were previously visited. The two icons at the bottom of the card are the Close Encounter Icon and a small snapshot of the vessel's track.

The cards can be displayed in the Record Browser, as shown in Figure 9. It allows an analyst to rapidly flip through a virtual deck of cards, enabling a visual scan for specific information. Dragging a particular card to the left part of the Record Browser facilitates visual comparison between vessels. Cards can be tagged as potential VOIs and a dot will appear on the bottom slider to identify those that were marked.

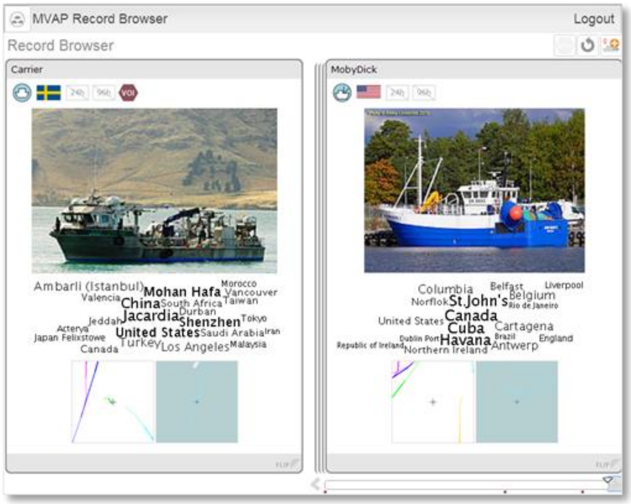


Figure 9. The Visual Summary Cards displayed in the Record Browser show the key vessel characteristics visually to allow for rapid scanning and visual comparison.

4.6 Multi-Timelines

The MVAP component for exploring temporal events is the Multi-Timelines tool (Figure 10). In this interactive visualization, each vessel has a horizontal timeline displaying its associated events. This tool is intended for closer analysis of a small number of vessels. The visual alignments of events along horizontal lines allow the visual comparison of these sequences for multiple vessels or even self-comparison when a vessel's line is duplicated. Unlocking the timelines enable comparison of sequences of events that happened at different time periods. This could lead to the identification of patterns or outliers.

In locked mode, dragging a timeline sideways will move all the other timelines synchronously. When the timelines are unlocked, we can drag them individually to align them on specific events that happened at different dates and visually compare how the situation evolved over time. The red line facilitates the manual alignment of individual timelines. Double-clicking on a specific event will also automatically align all the timelines to the nearest occurrence of this type of event in each timeline.

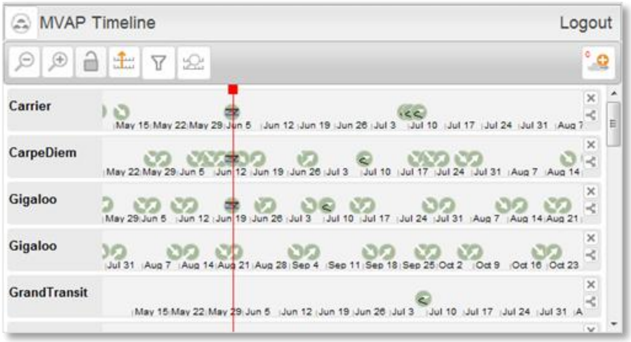


Figure 10. Multi-Timelines allow visual comparison of temporal events. A vessel's timeline can be duplicated to allow self-comparison and all the timelines can be unlocked to compare sequences of events happening at different time periods.

5. VALIDATION WITH TARGET USERS

The MVAP validation trials supported three objectives: evaluate the software usability, assess the potential operational value, and identify future improvements for the system.

Ten months before the validation trials, we held an early planning session onsite with nineteen maritime security analysts to guide the scenario and task development for the validation activity. They were briefed about the MVAP and their input influenced the tasks and datasets used in the validation trials.

5.1 Methodology

We employed a 'user jury' methodology to assess the MVAP, relying on the use of questionnaires and interviews with field experts [25]. We did not perform a direct comparison with operational tools because due to the lack of resources, many of the analyses that are offered by the MVAP are not currently being performed on a daily basis and there are no operational tools currently in use to support them.

Trials were conducted in groups of 3 to 5 participants in a single room. Each session lasted between sixty to ninety minutes and comprised three parts. It began with a hands-on training, followed by a set of tasks to be completed, and ended with participant assessment acquisition through questionnaires and interviews. During the whole process, observers recorded relevant observations about the participants' actions, questions, comments, errors and unanticipated strategies.

The training and scenario tasks relied on a combination of real operational data that was ingested into the MVAP and fictive scenario data where the information was not available.

5.1.1 Hands-on Training

During the training part, a MVAP expert explained the purpose and concepts of the widgets, one at the time. The presenter performed relevant operations on a large display screen, while the participants followed along on their individual stations.

Participants were trained for as many operations as possible in a short time period of 30 minutes. Some widget operations were only performed by the presenter and not carried out by the participants due to limited training time. The widgets were presented in the following order: Analysis Set Manager, Map and Timeline, Record Browser, Magnets Grid, Multi-Timelines, and modified Map and Timeline using real operational data.

5.1.2 Scenario-Based Exercise

We gave participants a scenario worksheet that included a summary of the assigned exercise and a set of six questions to be answered using the skills that they had learned during the MVAP training. They had 15 minutes to complete the challenge tasks. Due to timing constraints, no tasks in the exercise involved the Multi-Timelines.

First, the users manipulated the specialized version of the Animated Map and Timeline to request all tracks on a certain date in a specified area. Then, they had to find and export all the vessels that contained specific keyword to an analysis set in the main MVAP application. They animated the tracks in the Map and Timeline to estimate the expected ports of arrival of these vessels on the map. After that, the Magnets Grid was used to characterize the ships using a combination of construction properties such as build year and tonnage. Finally, the analysis concluded with a scan of the Visual Summary Cards to identify vessels with particular characteristics such as a specific flag.

5.1.3 Questionnaires and Interviews

After the tasks were completed, we handed participants the usability survey and a ranking sheet asking them to rank the usefulness of individual widgets and to write any comments they may have about them.

The sessions concluded with a hot wash group discussion where all participants were invited to share their thoughts and voice any questions they had about the MVAP.

5.2 Assessment Metrics

Three types of metrics were collected to assess the MVAP prototype. First, trial participants made an overall subjective assessment of the perceived effectiveness, efficiency, correctness, satisfaction, and trust of the MVAP features using the internationally recognized System Usability Scale (SUS) questionnaire [26, 27].

Then, they gave an individual rating to each widget according to their perception of the potential usefulness of each tool. The proposed scale included 3 values: "Not Useful (0)", "Possibly Useful (1)" and "Very Useful (2)". They were also invited to add comments about each widget.

Observers took notes to document all the observations, comments and questions that were expressed by participants at any time during the whole process. They also recorded the errors the participants made as well as the unanticipated strategies that they exhibited while working with the MVAP. These comments and observations provided the basis for an informal but insightful assessment.

5.3 Participants Selection

Sixteen maritime security specialists and analysts participated in the trials held on-site at the Halifax MSOC facility on November 13th and 14th, 2013. The positions of the participants included analysts, intelligence officers, Navy lieutenants, a watch officer, a coordinator, a maritime picture manager and maritime information management systems developers. The participants were very knowledgeable about the maritime domain, but were not familiar with advanced interactive visualizations. There was at least one representative from each of the five federal partners involved in the Canadian MSOCs [2]:

- Canada Border Services Agency;
- Canadian Coast Guard;
- Department of National Defence;
- Royal Canadian Mounted Police; and
- Transport Canada

6. TRIAL RESULTS

6.1 Overall MVAP Evaluation

Fourteen of the sixteen participants filled the System Usability Scale (SUS) questionnaire. Globally, the MVAP review was very positive (Figure 11). It obtained an average SUS score of 76, which means that the MVAP scored better than 75% of new software releases (using statistics from [26]). This is considered "Excellent" according to [28]. Observers noted that some participants were even trying features without waiting for the training, hinting that the user interface was intuitive and easy to understand.

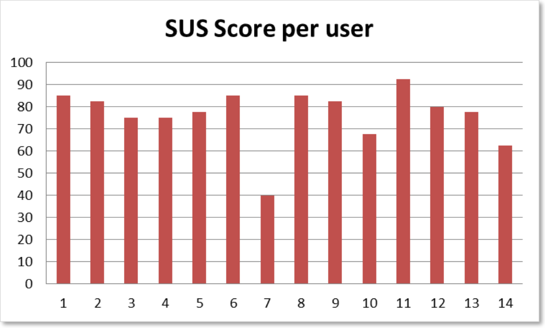


Figure 11. System Usability Score obtained from 14 participants for the overall MVAP evaluation.

Participants were very enthusiastic about the MVAP and indicated that they would definitely use it if it was made available to them. Their verbal and written comments suggested that it offered significant improvements over the current MSOC tools. Many participants were already thinking ahead about how the new features offered by MVAP could be integrated into the

MSOC. They provided insightful suggestions for extending the prototype's capabilities and adapting it to their specific operational context requirements.

6.2 Individual MVAP Widget Assessments

The five MVAP widgets have a high expected operational usefulness as a majority of participants ranking them as "Very useful" (Figure 12). The average rankings ranged between 1.5 and 1.9. One participant had to leave and did not fill this form.

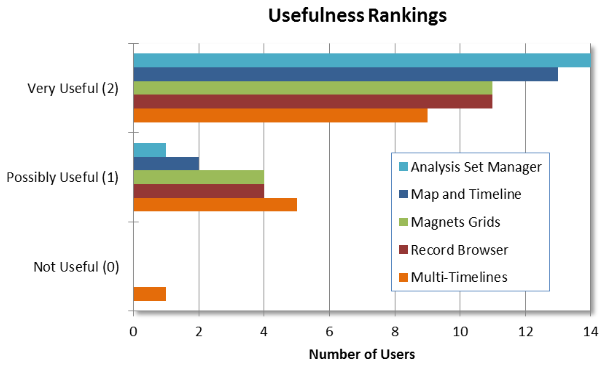


Figure 12. Usefulness rankings for individual widgets: 15 participants assessed each widget individually according to the expected usefulness that they perceived for their own operational work.

6.2.1 Analysis Set Manager

The Analysis Set Manager was among the top ranked widgets with an average of 1.9. Participants understood its role in structuring the analysis data and were quickly able to create analysis sets with the widget. They commented that the widget was clear and easy to understand, and would be great for grouping certain datasets. They recommended adding a number of import and export data formats to increase the analysis sets sharing capability between analysts from different departments or with other systems.

6.2.2 Animated Map and Timeline

The Animated Map and Timeline was also top ranked with a 1.9 average score. It was the widget that generated the most comments and some participants called it "the best app of all". Being able to visualize the time evolution of vessel tracks was one of the most esteemed capabilities. The search feature for close encounters was greatly valued and they stated that it was not possible to do this with their current tools. Participants were especially interested in the modified widget version that used their operational data.

Although moving time back and forth was very popular, the time slider interaction was difficult to master rapidly and led to many manipulation errors. To improve it, they suggested adding a time interval selection field to enter the initial settings.

Not much time was spent on Close Encounter Icons and the patterns created by real vessel tracks were much harder to interpret than Figure 5 would suggest. It is possible that after a longer exposure the patterns in the icons may become familiar. However, the MSOC environment where the turn-over in manpower is high makes this visual representation less desirable. Regarding the Route Ribbons, a similar visual representation is now part of their recognized maritime picture and it appears to be effective.

The comments highlighted how the interactive geo-temporal visualization could help making sense of a developing situation over time. Potential uses were suggested for marine security and regulation operations such as detecting polluters, determining baseline activities through pattern of life analysis, and helping communicating analysis results more clearly in briefings.

6.2.3 Magnets Grid

Although, it scored an average ranking of 1.7, Magnets Grid was very popular among participants, many of them calling it "awesome". This was partly due to its high novelty: there is no comparable visualization tool at the MSOC that enables exploration of these types of vessel characteristics visually.

The visualization looked complex at first but observations of self-learning from participants indicated that it was surprisingly easy to understand. In a very short time, the participants were very proficient with the Magnets Grid and had no major difficulties in completing the related tasks. The attraction arrows concept was not clear to at least one user. Another participant pointed out a weakness that appears in the presence of a very strong outlier, allowing it to dominate the canvas and force the others dots into a tight cluster.

Overall, participants thought the concept was very interesting, but some did not see how it would be useful to them because they did not expect to have access to a rich enough array of operationally relevant data to make the Magnet Grid effective.

Other participants suggested several ways to use the Magnets Grid for maritime security operations, such as comparing vessels, multi-parameter searching and fuzzy analysis. They even proposed new types of calculated attributes for magnets that would turn some geographic properties of vessel tracks into data usable with Magnets Grid, such as boundary crossing detections.

6.2.4 Visual Summary Cards and Record Browser

The Visual Summary Cards got an average ranking of 1.7. Participants thought it was useful and easy to understand but expressed concerns about the availability of the data used in the cards display. This is very important because the usefulness of this visualization is highly dependent on having detailed dynamic information about vessels. Sharing watch lists was proposed as an example of potential operational application.

Some participants wished the card's layout was user configurable so they could tailor it to their analysis needs. A comment was made about a similar tool being used already, although no details were provided regarding its usefulness or adoption rate among target users. A participant wanted to be able to look at both sides of the cards simultaneously.

6.2.5 Multi-Timelines

The Multi-Timeline scored lower than all the other widgets, with an average of 1.5. It is also the only tool that received a "Not Useful" rating. Participants understood that the widget's purpose was to analyse temporal patterns, but they explained that this kind of analysis is not frequent in their work. They still proposed potential ways to use it by adding geographically triggered events such as boundary crossings or close encounters detection. They valued the ability to unlock and shift temporal lines independently.

The Multi-Timelines widget was taught at the end of the training session, very often with little or no time for the user to really try it and was not covered in the challenge tasks. This could partially explain the low ratings from the users.

7. LESSONS LEARNED AND DISCUSSION

7.1 Modular Prototype

Separating the visualization requirements identified into a suite of smaller widgets resulted in a modular prototype. This modular approach gives us the flexibility to provide interested users with only the features that they want without requiring them to adopt a complex application framework.

Focussing on individual widgets allowed us to concentrate on the different aspects of the visual analysis requirements and to assess the effectiveness of the visual representations and interactions independently for each widget. An added benefit is to facilitate widgets reuse for other application domains (see section 8.4).

7.2 Importance of a Working Prototype in Visual Analytics

The ease with which trial participants understood the Magnets Grid widget was very surprising to us. Much earlier in this project, when we first explained the Magnets Grid concept to other maritime specialists in a presentation, they did not grasp its purpose and had difficulty envisioning its use. We thought this tool would require very detailed training and expected many participants to fail the related tasks.

This experience highlights the importance of showing a working prototype using realistic data to target users and not relying solely on their perception of a paper design.

7.3 Real Data Matters for Credibility

Using fake but realistic data helped users understand the purpose of the MVAP when they followed the training. However, we noticed a significant increase in the participants' attention when we moved to the modified Map and Timeline widget that uses their data.

Although real data was not available for all the features that we wanted to demonstrate, showing at least a part of the MVAP capabilities using real operational data was sufficient to prove that the prototype was not only nice research software, but had the potential to become an operational tool. Because we went through the development required to adapt our prototype, we were prepared to address their concerns about the MVAP being able to handle their large datasets.

At the other end of the spectrum, the Multi-Timelines was not showcased in the challenged tasks and users did not really get a chance to experience it with meaningful data. We think this is partly why it got a lower rating than all the other widgets.

7.4 Other Application Domains

A few participants' comments suggested that the MVAP could be useful beyond the MSOC context. By focusing on the analytic tasks to perform rather than the specifics of vessels information, we designed the widgets in a very generic fashion, expecting some to be used in a non-maritime context.

The adaptation of the MVAP widget to the social network analysis in a counter-insurgency context demonstrated the generic property of the visual analysis concepts proposed [29]. It should be noted that moving to a new application domain still requires development work to ingest the different data format and modifications to the visual interface, even though the basic concepts are the same. Notably, the Visual Summary Cards layouts need to be tailored for each type of analysis entity.

8. CONCLUSION

A study of maritime domain analysis performed at the MSOCs revealed that VA has the potential to enable better detection and analysis of marine threats. After identifying requirements related to visual detection of anomalous vessel behaviour and vessel of interest analysis, we designed the Maritime Visual Analytics Prototype using a modular approach. The capabilities of the MVAP are provided by a series of individual specialized visual analysis tools, leveraging both kinetic vessel track data and non-kinetic vessel information.

The Analysis Set Manager is the launching point for the other widgets and offers a hierarchical presentation of the vessel sets to analyse. The Animated Map and Timeline widget contains the geographical display that is central to the analysis and allows users to animate vessel tracks interactively while providing the Close Encounter Icons and Route Ribbons visual representation to highlight potential anomalous vessel activities. Using the Magnets Grid, factual information about vessels can be explored to detect trends and outliers across multiple dimensions simultaneously. The key characteristics of individual vessels appear in Visual Summary Cards. The Record Browser enables a quick scan of a vessel cards deck, as well as the visual comparison between cards for specific visual cues. Finally the Multi-Timelines widget provides an interactive interface to analyse and compare sequences of temporal events related to vessels.

Sixteen maritime security analysts and specialists assessed the potential for operational deployment of the MVAP employing a user jury methodology. They went through a hands-on training, a set of tasks to perform individually and filled out surveys about the tools.

Based on the positive validation trials results, we recommend that the MVAP be made available to MSOC developers so they can turn it into an operational tool.

9. ACKNOWLEDGMENTS

The author wishes to recognize the excellent work of Vincent Bergeron, Dany Dessureault and Régine Lecocq from DRDC; Eric Hall and William Wright from Oculus Info Inc; and Micheal Davenport from Salience Analytics Inc; all of whom contributed to the MVAP development and assessment. The author also thanks LCdr Tom Fredericks, Lt(N) Joseph Collins and Scott Syms who helped organise the validation trials, as well as all the MSOC operational staff who generously contributed to the MVAP evaluation.

10. REFERENCES

[1] 2004. National Security Presidential Directive/Homeland Security Presidential Directive, Maritime Security Policy, NSPD-41/HSPD-13.

[2] 2013. Marine Security Operation Centres Project, http://www.msoc-cosm.gc.ca/en/, (19 July 2013), accessed June 2014.

[3] Davenport, M. and Franklin, S. 2006. *Visions of a Future Maritime Picture*, MacDonald Dettwiler and Associates Ltd.,

DRDC Scientific Authority: David M. F. Chapman, DRDC Atlantic CR 2006-038 (May 2006).

[4] Davenport, M. 2007. *Maritime Domain Awareness Knowledge Management Requirements*, MacDonald Dettwiler and Associates Ltd., DRDC Scientific Authority: Jean Roy, DRDC Valcartier CR 2007-174, (July 2007).

[5] Davenport, M. 2009. *Opportunities for Applying Visual Analytics for Maritime Awareness*, MacDonald Dettwiler and Associates Ltd. & Salience Analytics, DRDC Scientific Authority: Valérie Lavigne, DRDC Valcartier CR 2009-227, (October 2009).

[6] Laxhammar, R. 2008. Anomaly Detection for Sea Surveillance, *Proceedings of the 11th International Conference on Information Fusion (Fusion 08)*, pp. 47–54, (June-July 2008).

[7] Johansson, F. and Falkman, G. 2007. Detection of Vessel Anomalies - a Bayesian Network Approach, *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing*.

[8] Dahlbom, A. and Niklasson, L. 2007. Trajectory Clustering for Coastal Surveillance, *Proceedings of the 10th International Conference on Information Fusion (Fusion 2007)*, (July 2007).

[9] Rhodes, B.J., Bomberger, N.A., Seibert, M. and Waxman, A.M. 2005. Maritime Situation Monitoring and Awareness Using Learning Mechanisms, *Proceedings of IEEE MILCOM 2005 Military Communications Conference*, pp. 646–652.

[10] Rhodes, B.J., Bomberger, N.A. and Zandipour, M. 2007. Probabilistic Associative Learning of Vessel Motion Patterns at Multiple Spatial Scales for Maritime Situation Awareness, *Proceedings of the 10th International Conference on Information Fusion (Fusion 2007)*, (July 2007).

[11] Riveiro, M., Falkman, G. and Ziemke, T. 2008. Visual Analytics for the Detection of Anomalous Maritime Behavior, *IEEE 12th International Conference Information Visualisation*.

[12] Riveiro, M., Falkman, G., Ziemke, T. and Warston, H. 2009. VISAD: an Interactive and Visual Analytical Tool for the Detection of Behavioral Anomalies in Maritime Traffic Data, W.J. Tolone and W. Ribarsky (Eds.), Visual Analytics for Homeland Defense and Security, *Proceedings of SPIE Defense, Security, and Sensing 2009*, (13–17 April 2009), Orlando, Florida, USA. SPIE Volume 7346, 734607 (1-11).

[13] Roy, J. 2009. Rule-Based Expert System for Maritime Anomaly Detection, *Proceedings of the 12th International Conference on Information Fusion (Fusion 2009)*, (July 2009).

[14] Edlund, J., r nkvist, M., Lingvall, A. and Sviestins, E. 2006. Rule Based Situation Assessment for Sea-Surveillance, *Proceedings of SPIE -Multisensor, Multisource Information Fusion: Architectures, Algorithms and Applications*, vol. 6242.

[15] Willems, N., Wetering, H. van de and Wijk, J.J. van 2009. Visualization of Vessel Movements, Computer Graphics Forum, *Proceedings of EuroVis 2009*, 28(3), pp. 959-966.

[16] Enguehard, R.A., Devillers, R. and Hoeber, O. 2011. Geovisualization of Fishing Vessel Movement Patterns Using Hybrid Fractal/Velocity Signatures, *GeoViz Workshop*, Hamburg, Germany, (10-11 March, 2011).

[17] Guo, H., Wang, Z., Yu, B., Zhao, H. and Yuan, X. 2011. TripVista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection, *Proceedings of Pacific Visualization Symposium (PacificVis)*, 2011 IEEE, pp. 163-170.

[18] Vatin, G. and Napoli, A. 2013. Guiding the Controller in Geovisual Analytics to Improve Maritime Surveillance, *Proceedings of GEOProcessing 2013: The Fifth International Conference on Advanced Geographic Information Systems, Applications and Services*.

[19] Keim, D.A. 2002. Information Visualization and Visual Data Mining, *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 1.

[20] Perer, A. and Shneiderman, B. 2009. Integrating Statistics and Visualization for Exploratory Power: From Long-Term Case Studies to Design Guidelines, *IEEE Computer Graphics and Applications*, pp. 39-51.

[21] Lavigne, V. Gouin, D. and Davenport, M. 2012. *Visual Analytics and Collaboration Technologies for the Maritime Domain*, Technical Report, DRDC Valcartier TR 2012-424, (December 2012).

[22] Lecocq, R. and Lavigne, V. 2013. Enabling Efficient Intelligence Analysis for Degraded Environments, *Proceedings of the 18th International Command and Control Research and Technology Symposium (ICCRTS)*, Alexandria, VA, US, (June 2013).

[23] Lavigne, V., Gouin, D. and Davenport, M. 2011. Visual Analytics for Maritime Domain Awareness, *Proceedings of IEEE Homeland Security Technologies 2011*, (15-17 November 2011), Waltham, MA, US.

[24] Yi, J.S., Melton, R., Stasko, J. and Jacko, J. 2005. Dust & Magnet: Multivariate Information Visualization using a Magnet Metaphor, *Information Visualization*, Vol. 4, No. 4, Winter 2005, pp. 239-256.

[25] Hall, E., Davenport, M., Bozowsky, N. and Wright, W. 2014. *Maritime Analytics Prototype: Phase 3 Validation Final Report*, Oculus Info Inc. & Salience Analytics, DRDC Scientific Authority: Valérie Lavigne, DRDC Valcartier, (January 2014).

[26] Sauro, J. 2011. *Measuring Usability With The System Usability Scale (SUS)*, http://www.measuringusability.com/sus.php, accessed June 2014.

[27] U.S. Department of Health & Human Services, 2013. *System Usability Scale (SUS)*, http://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html, accessed June 2014.

[28] Bangor, A., Kortum, P.T. and Miller, J.T. 2008. An Empirical Evaluation of the System Usability Scale, *International Journal of Human-Computer Interaction*, vol. 24, no. 6, pp. 574-594.

[29] Lavigne, V., Lecocq, R., Mokhtari, M. and Martineau, E. 2014. Graph Analyzer Widget: Closer to Agility through Sense-Making, *Proceedings of the 19th International Command and Control Research and Technology Symposium (ICCRTS)*, Alexandria, VA, US, (June, 2014).

Interactive Data Mining Considered Harmful[*] (If Done Wrong)

Pauli Miettinen
Max-Planck-Institut für Informatik
Saarbrücken, Germany
pauli.miettinen@mpi-inf.mpg.de

ABSTRACT

Interactive data mining can be a powerful tool for data analysis. But in this short opinion piece I argue that this power comes with new pitfalls that can undermine the value of interactive mining, if not properly addressed. Most notably, there is a serious risk that the user of powerful interactive data mining tools will only find the results she was expecting. The purpose of this piece is to raise awareness of this potential issue, stimulate discussion on it, and hopefully give rise to new research directions in addressing it.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*

Keywords

Interactive data analysis; statistical testing; white paper

1. INTRODUCTION

Traditionally, the KDD process was presented as a waterfall, going from pre-processing to data mining to post-processing (solid lines in Figure 1). This—of course—has never been true, and more modern models of data mining, such as Shearer's CRISP-DM model [12], reflect that. Data analysis is an iterative process: the user prepares the data, selects analysis methods and their parameters, runs the methods, studies the outcome, and returns to any of the earlier steps, possibly preparing the data differently, or using different analysis method or different parameters (dashed lines in Figure 1).

But this iterative process is arduous and each step that needs to be repeated can take a significant amount of time. To help with this is what the *interactive* data mining is for: to allow the user to pinpoint the analysis method to the interesting results without the time-consuming iteration. Done well, interactive data mining methods can be extremely powerful, giving the user unprecedented machinery to better understand her data. But with great power comes great responsibility, as the saying goes. By allowing the user to control the data mining process in (near) real time, interactive data mining systems posses the risk of undermining the very promise of data mining: discovering new and unexpected knowledge.

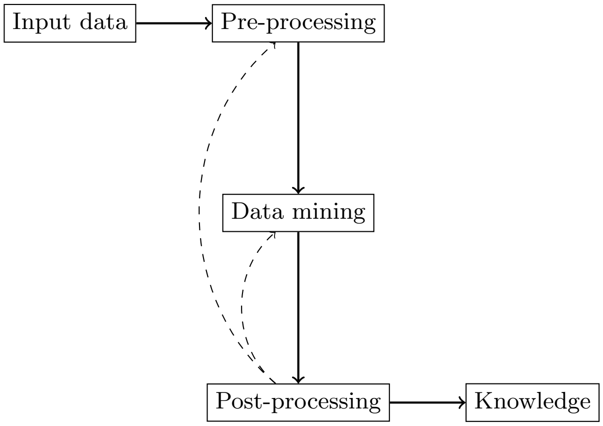


Figure 1: The iterative KDD process

2. THE PROBLEM

The goal of data mining, in the words of one textbook, is

> [T]o find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner. [4]

Data mining community has always been good at inventing novel ways to mine the data, but has perhaps struggled more with the understandability and usefulness parts. It is these two areas that interactive data mining tries to improve by letting the user to tell the algorithm, during the mining process, what she finds useful and understandable. But doing so, it threatens a very important aspect of data mining mentioned in the above quote: the results should be *unsuspected*.

The user using the interactive data mining method is (hopefully) familiar with the data and what it represents. Consequently, the user has some prior ideas what the potential results could be, and what kind of a result would be a useful result in this domain. But these prior ideas might—indeed, I argue they will—make the user steer the algorithm towards the kind of results that she a priori considered useful and interesting, and never find the kind of results she did not expect to find. This can make the interactive data mining, intended to be exploratory by nature, a confirmatory data analysis technique—and not necessarily very good method at that, even.

To give a more concrete example, consider an interactive data mining algorithm that presents the user with partial results in an anytime fashion and lets her to guide the search with feedback such as "more like this" or "less like this."

[*] With apologies to Edsger W. Dijkstra

Contemporary interactive data mining methods might not quite achieve this level of interaction yet, but it is clear that it would be desirable if they would. It should be obvious, however, how the user can, possibly unintentionally, use this feedback mechanism in such a way that the algorithm only returns results that she was expecting.

3. IS IT A (NEW) PROBLEM AT ALL?

Is this a real problem? Is it not a far-fetched idea that the user would have on her mind the exact results the mining algorithm will find? It indeed is, but it is important to note that this problem appears as soon as the user has even a vague a priori idea on what would be a useful result from the algorithm. And for a user with only a faint idea on what could be useful, what is the purpose of interactive data mining, what is its added value? The potential lose of surprising results is the price to pay for the power of interaction, the Jekyll and Hyde of interactive data mining.

But has this problem not been part of data mining all the time? As already discussed, the process of knowledge discovery is iterative and the user can repeat the steps trying to extract more understandable and useful results, potentially removing the more surprising results while doing so. But interactive data mining tools can emphasize this problem significantly by giving the user a faster access to the mining process; indeed, interactive, rather than iterative, access. Again, the problem lies in the heart of interactive data mining: the power that interactive data mining gives to the user over the iterative data mining is exactly the same power that lets the user to only find the unsurprising results.

The users, one could argue, would not intentionally avoid the unsuspected results. But oftentimes, it is hard to appreciate such results in the first glance. The results, being unsuspected, might look like noise or random occurrences as they do not fit into our thinking of the data. They might require us to update our understanding of the data, possibly running more experiments, before we can appreciate them, all of which makes the process significantly less interactive. Yet, it is precisely the change in understanding the data these results require that makes them so valuable for the mining process.

A related problem in statistics and machine learning is that of over-fitting. By steering the data mining process away from unsuspected results, the user is effectively over-fitting the results into her prior assumptions. But this kind of over-fitting is much harder to address than the more common one. The final arbitrator for the quality of a machine-learning algorithm is its predictive power. But data mining is descriptive, rather than predictive, and in many cases, there is no clear prediction stemming from the results. There is no objective quality measure, either, as here the user is the arbitrator of the quality.

4. POSSIBLE SOLUTIONS

Arguably the simplest solution is user education. The power to interact with the algorithm is vested in the user, and she should be taught how to use this power. Unfortunately, education alone cannot solve all the problems. The risk of missing important but unsuspected results exists whenever the user is allowed to interact with the algorithm, any education notwithstanding, and if this power is removed from the user, there is not much interactive data mining left.

Another simple approach is to restrict the power of the interaction, keeping the situation closer to status quo. It should go without saying that this approach is sub-optimal.

The potential for data mining algorithms, and their users alike, to concentrate on "wrong" results has existed all the time. Significant amount of data mining research is devoted to testing whether a specific result is significant with respect to some null hypothesis (e.g. [2, 3, 8, 9, 11]) or even with respect to user's prior knowledge (e.g. [1, 5, 10]), to say nothing about the vast body of statistical literature on measuring the statistical significance. In principle, the approach these papers take can be used to steer the user and the algorithm away from expected results: encode the users prior knowledge in the null hypothesis and discard results that are not significant under this null hypothesis (and interactively update the null hypothesis when new results are obtained).

While the general approach of using significance testing is very appealing, it is not clear at all whether it can be used to actually alleviate the problem in the interactive setting. First, the significance testing must be instantaneous—or at least fast enough to be used interactively. Some methods, for example the maximum entropy methods, should be able to pass this hurdle, while others, such as permutation test style swap randomization, most probably will not. Second, the user should be able to communicate her a priori assumptions to the method so that they can be build in to the null hypothesis. Given that even a vague prior belief can have a negative effect, this might be too tall an order. It could be circumvented to some extend by simply relying on the interactive nature of the algorithm: updating the null hypothesis based on user's interaction with the algorithm and her reactions to the new results could reveal enough of her latent a priori assumptions for the method to work.

The biggest hurdle for this method, however, is in its very nature: significance testing is designed to spot insignificant results, but it does not, per se, help at finding new significant results. For example Mampaey et al.'s method [10] rely on clever algorithms to actually find the patterns. Should such algorithm be endowed with "more like this/less like this" kind of functionality, there would still be nothing stopping the user from steering the algorithm away from unsuspected results. It could well happen that the user would find almost nothing of significance: her own actions would guide the algorithm away from the unsuspected results, while the significance testing would deem almost all of the remaining results redundant or insignificant with respect to the prior knowledge.

In fact, it might well be that there is no (computationally efficient) solution to the problem, at least not unless we place strong assumptions on the users' behavior. In the statistical query model of Kearns [7], the user asks questions about the expected value of a predicate over some (finite) distribution. The algorithm, called oracle, does not know the distribution, but has access to a sample of size n from it. The algorithm's task is to give valid answers, that is, answers that do not deviate too much from the true expectation, based only on the sample. In their recent paper, Hardt and Ullman [6] showed that there is no computationally efficient algorithm that can give valid answers to $n^{3+o(1)}$ *adaptive* statistical queries assuming one-way functions exist[1].

[1] A *one-way function* is, informally, a function which is easy to compute for any input, but hard to invert given an image of a random input. Their existence is a standard assumption in much of modern cryptography.

The crux in Hardt and Ullman's result is the adaptivity, as giving valid answers to even exponential number of non-adaptive statistical queries is easy. We can interpret the result in two ways: On one hand, it at least shows that adaptive queries are significantly harder to answer correctly than non-adaptive ones. On the other hand, we can interpret the result to tell even more about the computational limitations of interactive (and iterative, for that matter) data analysis systems: that it is impossible to prove that our results are even correct, to say nothing of surprising, assuming that the user can ask sufficiently many adaptive questions.

5. TESTS

The final, and perhaps the most important, piece on addressing the problem is testing it. Without testing, we do not know if the problem even exists, nor can we assess the effects of proposed solutions. Developing tests to measure if the interaction makes the users to miss unexpected results is, unfortunately, not easy. It does not seem likely that it could be tested without involving humans to act as users. A potential test could have two groups of users, a test group using the interactive algorithm, and a control group using non-interactive algorithm. Their findings would then be evaluated to measure whether the test group missed results the control group found, or vice versa. But even this seemingly simple test setup requires many design decisions to be made—where are the test subjects found, what are the group sizes, how can it be ensured that the test is fair, and how are the results interpreted—and traditionally data miners have not been the ones with best knowledge about and keenest interest on human experiments. Luckily, this is a problem that should be very easy to solve by collaborating with experts.

6. CONCLUDING REMARKS AND CALL FOR ACTIONS

Interactive data mining is a powerful form of data analysis with the potential of becoming the standard format of data mining. But it comes with new pitfalls that need to be taken into account when new interactive data mining methods are developed and analyzed, lest the results become void of unexpectedness. The community should, therefore, start addressing the problem of finding only expected results: we need methods to test the seriousness of the problem and the effects of the attempts to alleviate it; we need general frameworks to help avoiding the problem; and we need interactive algorithms that try to steer the user away from discovering only the expected results. But above all, we need to realize that this is a potential problem and start thinking about it.

7. REFERENCES

[1] T. De Bie. Maximum entropy models and subjective interestingness: An application to tiles in binary databases. *Data Min. Knowl. Discov.*, 23(3):407–446, 2011.

[2] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. Assessing data mining results via swap randomization. *ACM Trans. Knowl. Discov. Data*, 1(3), 2007.

[3] W. Hämäläinen. *Efficient search for statistically significant dependency rules in binary data*. PhD thesis, University of Helsinki, Oct. 2010.

[4] D. J. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. MIT Press, Cambridge, Massachusetts, 2001.

[5] 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 *KDD '09*, pages 379–388, June 2009.

[6] M. Hardt and J. Ullman. Preventing false discovery in interactive data analysis is hard. In *FOCS '14*, 2014. To appear.

[7] M. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, Nov. 1998.

[8] A. Kirsch, M. Mitzenmacher, A. Pietracaprina, G. Pucci, E. Upfal, and F. Vandin. An efficient rigorous approach for identifying statistically significant frequent itemsets. *J. ACM*, 59(3), June 2012.

[9] K.-N. Kontonasios, J. Vreeken, and T. De Bie. Maximum entropy modelling for assessing results on real-valued data. In *ICDM '11*, pages 350–359, 2011.

[10] M. Mampaey, N. Tatti, and J. Vreeken. Tell me what i need to know: Succinctly summarizing data with itemsets. In *KDD '11*, pages 573–581, Aug. 2011.

[11] M. Ojala. Assessing data mining results on matrices with randomization. In *ICDM '10*, pages 959–964, 2010.

[12] C. Shearer. The CRISP-DM model: The new blueprint for data mining. *J. Data Warehous.*, 5(4), 2000.

Interactive Exploration of Comparative Dependency Network Learning

Diane Oyen
Los Alamos National Lab
Los Alamos, New Mexico, USA
doyen@lanl.gov

Terran Lane
Google, Inc
Cambridge, MA, USA
terran@cs.unm.edu

ABSTRACT

Comparative dependency network learning is a growing field of research, especially in systems biology. Domain scientists would like to discover patterns of variable dependency that are conserved across conditions or discover pathways that are disrupted due to disease. In machine learning, multitask graphical structure learning algorithms have been developed to help solve this problem by learning network models from multiple related datasets. These algorithms typically have regularization hyper-parameters that have the effect of reducing the number of spurious edges learned and the number of spurious differences learned. We propose a mechanism to allow the end-user to control these regularization hyper-parameters in real-time to interactively explore the huge space of potential dependency network solutions. This is a critical element of a visualization system that enables domain scientists to discover interesting patterns in multivariate data. Yet, this is a computationally challenging endeavor as complex models must be learned in real-time and, additionally the number of differences learned in each network and the number of differences between them must be translated by the machine learning algorithm into the correct change in the setting of the hyper-parameters. This paper introduces a general framework for interactively exploring the similarities and differences among a set of dependency networks and demonstrates our work-in-progress on a specific implementation for multiple Bayesian networks.

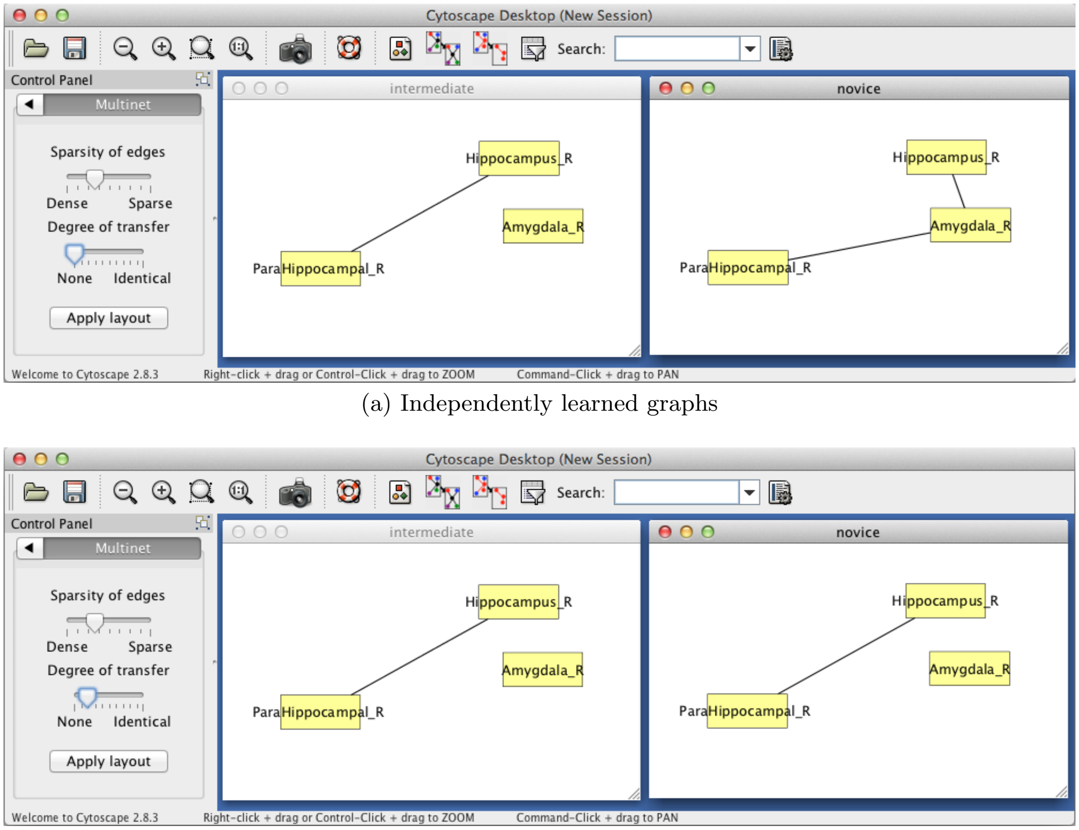
1. INTRODUCTION

Probabilistic graphical models encode patterns of dependency among variables in multivariate data [11]. Attention is turning to the problem of comparative network analysis; that is, identifying dependencies that are conserved or different among related sets of data. In machine learning, multitask graph learning algorithms have been developed to address this problem [17, 4]. Multiple graphs are learned simultaneously, producing models that are similar except where the data strongly supports differences, easing comparison (see example in Figure 1). The results of these learning algorithms are multiple graphical models, requiring visualization software to help the user understand the results.

Multitask graph structure learning is a promising direction for knowledge discovery in many scientific domains [25, 16, 5, 19]. However, there remain issues of practical concern; namely, the exploration of the solution space for different settings of hyper-parameters. The solution space includes many graph structures that fit the data nearly equally well, but the learned solutions vary based on the choice of hyper-parameters given to the learning algorithm. Multitask graph structure learning algorithms typically have two hyper-parameters, one that affects the number of edges learned (sparsity) and the other that controls the strength of transfer bias (how similar the graphs will be to each other). Machine learning typically treats these hyper-parameters as nuisance parameters that must be tuned to learn an optimal model [14, 13, 24, 15, 17]. Yet, to the end user, all learned models — no matter the choice of hyper-parameters — are good fits to the data. There is a natural tradeoff between sensitivity and specificity in machine learning algorithms governed by the hyper-parameters. Domain scientists would like to be able to vary the level of confidence in learned models to see which are the highest confidence edges and/or differences to explore all potential dependency patterns in the data.

Naively, exploration of the solution space is achieved by performing a grid search over hyper-parameter settings and then presenting each of these learned graphs to the end user. Yet, this grid search is not an ideal approach. To illustrate the problem of grid search, we take a look at example results from a neuroimaging study. Using an existing multitask graph learning algorithm [4], we assign the sparsity hyper-parameter to 10 values evenly spaced in the range $[0.1, 1]$ and we assign the transfer hyper-parameter to 11 values evenly spaced in the range $[0, 1]$. All 110 combinations of sparsity and transfer settings were run through a multitask algorithm and the solutions were displayed to a domain expert who found which solution was most interesting. Taking a broader look at the results from all 110 settings of these parameters, Figure 2a summarizes the number of edges learned in the networks for all of the values in the grid. We can see from this that if the sparsity setting is too high, then no edges are learned in the graphs. Yet, if the sparsity parameter is too small, then all variables are dependent on each other and this gives little new information to the end-user. Figure 2b shows the number of edges that are different between the two learned networks. We see that for many settings of transfer and sparsity there are many solutions that are uninteresting because there are no differences. Another frustration with this grid search is that the number of edges or differences learned does not change linearly with evenly-spaced steps in parameter space. Tuning the hyper-parameters over a coarse grid like this could easily miss the optimal hyper-parameter setting.

After noticing that the most interesting results are located within a narrow range of hyper-parameter settings, we can re-run the multitask network learning algorithm for



(a) Independently learned graphs

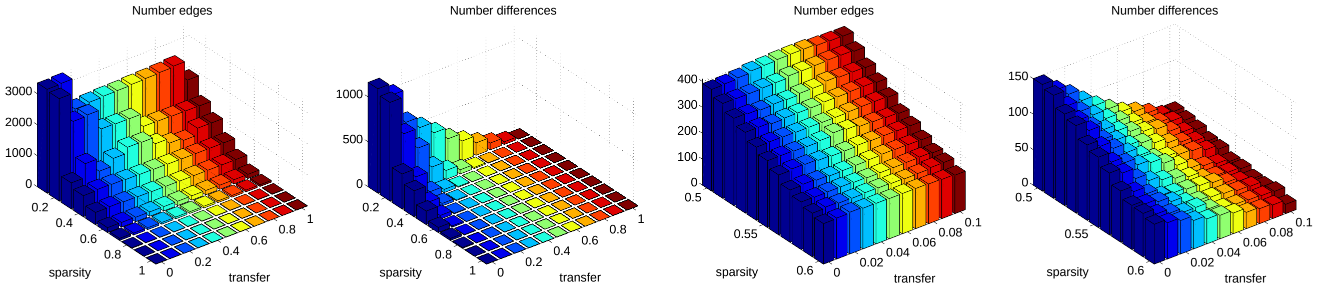
(b) Graphs learned with some transfer

Figure 1: Example of a sub-graph learned from neuroimaging data. The nodes in the graph are regions of the brain. Edges indicate a direct dependent relationship in functional activity as modeled by a multinomial distribution; i.e. an excitatory or inhibitory pathway between brain regions. When the graphs are learned independently (a), the connections are different. If this were the only result given to the domain scientist, we might conclude that the Amygdala has a regulating effect on the pathway between the Hippocampus and ParaHippocampus in one group of subjects, but not in the other group. However, with even a little bit of transfer bias encouraging them to be similar (b), the differences disappear, suggesting low confidence that this difference is real. This is a small sub-graph of a much larger graph. Higher confidence differences in the larger graph could still remain at this low value of the transfer hyper-parameter.

new values of hyper-parameters. As an example, we “zoom in” on the range $[0.5, 0.6]$ for the sparsity parameter and the range $[0, 0.1]$ for the transfer parameter. The algorithm is run with another 100 combinations of values for hyper-parameters evenly spaced in this new range. Figure 2c gives the number of individual edges learned in each networks while Figure 2d displays the number of differences between the two networks. Here we see that the numbers of edges and differences learned change smoothly in this local region of hyper-parameter space.

The above example demonstrates key limitations of existing exploratory approaches in comparative dependency networks, which we address with our interactive approach. First, instead of changing the hyper-parameters, we must think about how the results will change. In this case, the end user thinks about seeing more or fewer edges or differences (assuming that fewer edges are higher confidence edges, etc). Therefore, we need a computational model that translates the human desires (number of edges and differences) into the domain of the hyper-parameters of the machine learning algorithm. Second, the end user needs to be able to get fine-grained results in realtime to effectively explore the space of solutions that are of interest. Thus, an effective interactive exploration of dependencies must translate human desire into machine learning objectives and update in realtime in response to user feedback.

This paper introduces an interactive machine learning algorithm for multitask graphical models as part of ongoing research into creating an interactive data exploration visualization system (pictured in Figure 3). A single setting of the hyper-parameters does not give the full picture that domain scientists want to see. Therefore, we propose a graphical structure learning algorithm that allows the user to interactively adjust the number of edges and the number of differences learned between graphs. As the user makes selections about increasing/decreasing the number of edges or the number of differences between graphs, we estimate the necessary change in the hyper-parameter values and re-learn the networks, displaying the results and allowing further interaction. This approach gives the user an exploration of the solution space directly, rather than having to guess pairs of values for hyper-parameters. Essentially, we are giving the user the ability to explore fine-grained steps in the solution space, and making the appropriate steps in the hyper-



(a) Number of learned edges (b) Number of differences (c) Number of learned edges (d) Number of differences

Figure 2: Neuroimaging study: Summary statistics about learned network models for a course (a,b) and fine (c,d) grid of values of sparsity and transfer hyper-parameters.

parameters to achieve that result, rather than using a typical grid search in hyper-parameter space.

2. RELATED WORK

To interactively explore graphical models, we need to provide a means to adjust parameters of interest to the user and display the resulting graphs. Display of the graphs is handled through the Cytoscape software that is popular in bioinformatics [23]. The plugin interface allows us to customize the display for comparison of multiple graphs (see Figure 3). We incorporate sliders that allow a user to modify the sparsity and degree of transfer among networks. Previously, in our implementation, these sliders simply looked up the pre-computed graphs learned from a list of hyper-parameter values [20]. The user did not have any control over the granularity of the slider, and furthermore, changing a hyper-parameter value may not always have the desired effect (for example, on sparse graphs, even a small amount of transfer will cause the graphs to be identical). Therefore, we propose to provide more intuitive controls to the user, allowing them to change the number of edges or the number of similarities directly.

The idea of interactive parameter search is inspired by work in supervised learning models that show that with human interaction, the optimal parameter settings are found faster [1] and gives the user control over the objective function [9]. As in these papers, to achieve this interactive exploration in multitask graph structure learning, we must be able to estimate the values of hyper-parameters that will produce the desired change in the solution space. We achieve this by calculating the gradient of the solution with respect to the hyper-parameters and then taking a step in the direction of the gradient to produce a new solution that meets the requirements of the user.

In graph structure learning literature, much research has gone into optimizing the selection of the sparsity parameter [14, 13] without a clear resolution to the problem. Traditionally, the hyper-parameters are tuned through trial and error after examining the learned graphs [4] or through a computationally expensive grid search that optimizes with respect to holdout data [24, 15, 17, 19]. Graph structure learning is an unsupervised learning domain and so there may not be an optimal parameter setting. Even using the oracle value of hyper-parameters does not guarantee optimal performance [15], instead that paper recommends using known non-interactions to gauge the optimal level of sparsity. Selecting the ideal setting of transfer parameters has received less attention, with cross-validation being the preferred method [17, 19] and subjective human-selection being another choice [4]. Cross-validation selects the best model to match the empirical distribution; yet, distribution matching is not always the primary goal for using transfer learning, and therefore cross-validation will not give optimal results. Giving the user the ability to explore the solution space is even more important in unsupervised learning. The user may have desires about learned models that are not expressible until the learned models are seen [3]. Furthermore, allowing a user to give feedback about the solutions is more intuitive than asking the user to adjust hyper-parameters in the hopes that the adjustments will have the desired effect.

3. FRAMEWORK OF USER INTERACTION

We formalize the general problem of learning multiple graphical model structures and describe a method for including user input in response to learned models.

3.1 Problem Formulation

A graphical model is a joint probability distribution of a random vector $X = [x_1, x_2, \ldots, x_p]$ that can be represented compactly as factors of local structure, $P(X) = \prod_{i=1}^{p} f(x_i, ne(x_i))$, where the set of neighbors of each node, $ne(x_i)$, is some subset of variables. The elements of vector $X = [x_1, x_2, \ldots, x_p]$ are random variables represented in the graphical model as vertices (or nodes) as the set V. If $x_p \in ne(x_q)$, then there is said to be a direct dependency between these two variables which is represented with an edge e_{pq}. The set of all edges is called E. In many cases, the graph structure itself $G = \{V, E\}$ is of particular interest.

In the problem of multitask graph structure learning, we have several sets of data, D_k for $k \in \{1, 2, \ldots, K\}$, from which we learn several graphs $\mathcal{G} = \{G_1, \ldots, G_K\}$. The multitask structure learning algorithm relies on two hyper-parameters, which we call $\Lambda = [\lambda_1, \lambda_2]$, where generally $0 < \lambda_1 \leq 1$ controls the sparsity and $0 \leq \lambda_2 \leq 1$ controls the strength of transfer. We treat the graph structures $\mathcal{G}$ and Λ as unknowns to be learned. For a fixed Λ, the graphs can be learned from the data with existing algorithms. The user will interactively learn Λ by giving feedback to the learning algorithm about the number of edges and edge similarities that they would like to see in the learned graphs.

In this paper, we represent the set of edges in all of the K graphs with $\mathbf{G}$, an $m \times K$ binary matrix, where m is the total

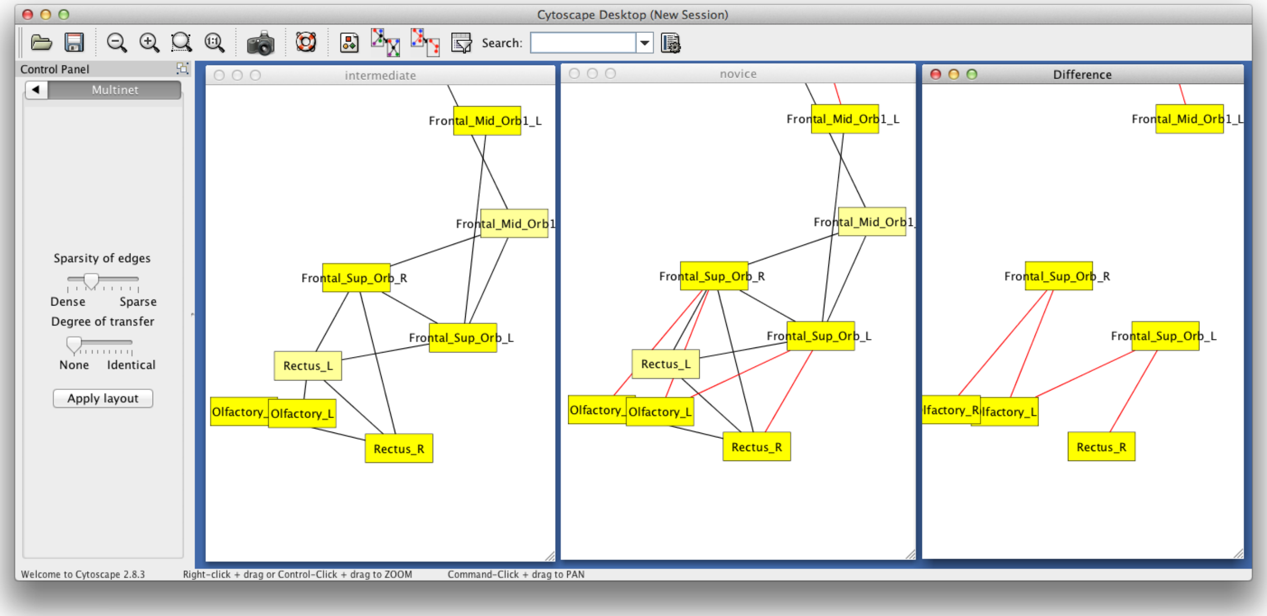


Figure 3: Interactive multi-graph visualization. Our system consists of the following components: a visual display of multiple learned graphs, user controls to increase/decrease the number of edges in each graph, user controls to increase/decrease the degree of similarity among pairs of graphs, efficient update of learned graphs in response to user controls.

number of potential edges (for directed models $m = p(p-1)$ and for undirected models $m = p(p-1)/2$). Each entry G_{ik} represents the presence ($G_{ik} = 1$) or absence ($G_{ik} = 0$) of the edge i in task k. In a slight abuse of notation, we use G_{ik} to refer to edge i in task k, while G_k refers to all potential edges in task k. The structure of the learned graphs depends on the training data and the hyper-parameters $\Lambda = [\lambda_1, \lambda_2]$. While looking at a given solution, a human end-user may desire to see a solution with more (or fewer) edges in some G_k or with more (or fewer) edge differences between some G_i and G_j for tasks i and j. These desires are encoded in a binary matrix $\mathbf{S}$ that correspond to the edge matrix $\mathbf{G}$ (explained in further detail later).

3.2 Sketch of Interactive Approach

Our interactive approach alternates between learning graph structure, $\mathcal{G} = f(\mathcal{D}, \Lambda)$, and learning hyper-parameters, $\Lambda = g(\mathcal{D}, \mathcal{G}, \mathcal{S})$, based on feedback $\mathcal{S}$ from a human who is looking at a visualization of the learned graphs $\mathcal{G}$. To initialize the interaction, we learn a set of graphs from given datasets. These graphs can be learned with transfer bias ($\lambda_2 = 0$) initially with an arbitrary value for the sparsity (e.g. $\lambda_1 = 0.5$). These graphs will be displayed in Cytoscape, along with information in the Control Panel about the number of edges learned in each graph and the number of differences in edges among the tasks. The user can then adjust the desired number of edges learned (up or down) or adjust the number of differences among pairs of tasks. Based on the user input, $\mathbf{S}$, we compute the necessary Λ to achieve the requested change (details in the next section). Using the computed Λ, we relearn the graphs, $\mathcal{G}$, and update the visualization, allowing the user to further interact until satisfied with the solution.

3.3 Representation of User Feedback

When a user clicks to change the number of edges in a graph or the number of differences among graphs, the user is not directly changing the hyper-parameters. We must translate the feedback into an appropriate change in the hyper-parameters to produce the desired outcome. To represent user preferences, we use a binary matrix, $\mathbf{S}$, the same size as $\mathbf{G}$ ($m \times K$). Each entry, S_{ik}, indicates the user's desire to see the presence or absence of edge i in task k. Using this input, we can move the learned structures $\mathbf{G}$ in the direction of the user preferences $\mathbf{S}$ by finding an appropriate adjustment to Λ.

An example illustrates how the user representation works. Consider the case where a user wishes to see fewer differences between tasks a and b. Let the currently existing set of non-zero edges in graph a be A and the set of non-zero edges in graph b be B. Then the user feedback defines a set U of edges, any one of which could change to satisfy the user. If the user wishes to see fewer edges that exist in a but not b then the set difference $A \setminus B$ must get smaller. Therefore $U = A \setminus B$. We set $S_{ea} = 0 \; \forall e \in U$ and $S_{eb} = 1 \; \forall e \in U$. $\mathbf{S}$ encodes the user-preferences to see one of the specific edges to be added or removed. The remaining entries in $\mathbf{S}$ are set to the current values of $\mathbf{G}$, i.e. $S_{ek} = G_{ek} \; \forall e \notin U$ and $\forall k$.

Formally, the rules for representing user feedback depends on the action taken by the user. The rules are defined as:

- **Fewer edges in task i:**
  Assign $S_{ei} = 0 \quad \forall\, e = \{1, 2, \ldots, m\}$.
  Assign $S_{ek} = G_{ek} \quad \forall\, e = \{1, 2, \ldots, m\}$ and $\forall\, k \neq i$.
- **More edges in task i:**
  Assign $S_{ei} = 1 \quad \forall\, e = \{1, 2, \ldots, m\}$.
  Assign $S_{ek} = G_{ek} \quad \forall\, e = \{1, 2, \ldots, m\}$ and $\forall\, k \neq i$.

- **Fewer edges in task i that are not in task j:** Define set $U = \{e = \{1, 2, \ldots, m\} \mid G_{ei} = 1 \wedge G_{ej} = 0\}$. Assign $S_{ek} = 0 \quad \forall\, e \in U$ and $k = i$. Assign $S_{ek} = G_{ek}$ otherwise.
- **More edges in task i that are not in task j:** Define set $U = \{e = \{1, 2, \ldots, m\} \mid G_{ei} = 0 \wedge G_{ej} = 0\}$. Assign $S_{ek} = 1 \quad \forall\, e \in U$ and $k = i$. Assign $S_{ek} = G_{ek}$ otherwise.

3.4 Local Move Toward User Desires

The goal is to obtain a setting for $\Lambda = [\lambda_1, \lambda_2]$ that creates graphs that are nearly the same as the current solution, but one edge closer to the user's desires $\mathbf{S}$. Therefore, we define an objective function that measures the squared error between $\mathbf{S}$ and $\mathbf{G}$:

$$g(\Lambda) = \sum_{k=1}^{K} \sum_{e \in E} (S_{ek} - G_{ek}(\Lambda))^2 \quad . \tag{1}$$

The user's feedback asks us to take just one step in the direction of this objective (only one edge is added or deleted at a time). We are not fully optimizing the objective. The gradient is given in Eq 2:

$$\begin{aligned} \nabla_{\Lambda} g &= -2 \sum_{k=1}^{K} \sum_{e \in E} (S_{ek} - G_{ek}(\Lambda)) \cdot \nabla_{\Lambda} G_{ek}(\Lambda) \\ &= -2 \cdot \mathbf{J}_{\Lambda}(\vec{G}) \cdot (\vec{S} - \vec{G}) \quad , \end{aligned} \tag{2}$$

where $\vec{S}$ and $\vec{G}$ are vectors formed by stacking the columns of the $\mathbf{S}$ and $\mathbf{G}$ matrices respectively. $\mathbf{J}_{\Lambda}(\vec{G})$ is the $2 \times |\vec{G}|$ Jacobian matrix, with each entry in the first row the partial derivative of G_{ek} with respect to λ_1 while the second row is with respect to λ_2. Our objective is to find the minimum step size η that gives the incremental change requested. Once η is found, the new value of the hyper-parameters is:

$$\Lambda^{\text{new}} = \Lambda - \eta \cdot \nabla_{\Lambda} g \quad . \tag{3}$$

The new hyper-parameter values are fed back into the learning algorithm, the visualized results are updated and the cycle continues if the user gives more feedback.

3.5 Computational Challenges

The above objective requires two computationally expensive steps. The first is the calculation of the Jacobian (the gradient $\nabla_{\Lambda} G_{ek}(\Lambda)$). The computational complexity of this depends on the specific model of multitask graph structure learning used. For the Bayesian discovery of multitask Bayesian networks format given in the next section, the partial derivative with respect to λ_1 (sparsity) is trivial, but the partial derivative with respect to λ_2 (the transfer strength) is computationally equivalent to calculating the multi-task family scores. Which is to say that it is exponential and could take minutes (depending on complexity-reducing approximations). However, we note that the gradient depends only on the current model and not user feedback. Therefore, the gradient can be calculated in the background while the user is looking at the previously learned graphs and making a choice about feedback to give.

The other computationally expensive procedure is the inference of $G(\Lambda)$ for each task. For the Bayesian discovery of multitask Bayesian networks approach given here, to update G the graphs must be re-learned (exponential time, or approximated with MCMC).

4. EXPLORATION OF MULTITASK BAYESIAN NETWORKS

Here we apply the above framework for interactive graph exploration to the specific problem of Bayesian discovery of multiple Bayesian networks, particularly those with transfer bias from related data. Then we discuss how to cache intermediate calculations to make updating the transfer bias faster on subsequent calculations. Finally, we show how discrete graphs are obtained from the expectations on edges.

4.1 Preliminaries

A Bayesian network is a directed acyclic graph that represents a joint probability distribution as $P(X) = \prod_{i=1}^{p} P(x_i|\pi_i)$, where π_i is the parent set of child i. That is, the value of x_i depends directly on the values of all $x_j \in \pi_i$. Bayesian structure discovery produces a posterior estimate of the expectation of each edge in a Bayesian network [7, 10]. For multitask Bayesian networks, there is a posterior estimate of the expectation of each edge in each task, which we organize into a matrix $\mathbf{W}$, denoted $0 \leq w_{ek} \leq 1$. An edge is described by an indicator function $f_i(\pi_i)$ such that the edge $x_v \rightarrow x_i$ exists (and $f_i(\pi_i) = 1$) iff $x_v \in \pi_i$, otherwise $f_i(\pi_i) = 0$. The probability of the edge w_{ek} is therefore the expectation of f in task k for that edge. The expectation is calculated over all orderings, $\prec$, of the nodes in the Bayesian network, as in Equation 4. For a given ordering, the parents of a node i must precede i in the order.

$$w_{ek} = \sum_{\prec} P(\prec) \sum_{G \subseteq \prec} P(G|\prec) P(\mathcal{D}|G) f_e(\pi_e) \ , \tag{4}$$

where $G \subseteq \prec$ means that the graph structure G is consistent with the order $\prec$; and π_e is the parent set of node x_e in graph G.

Relatively efficient methods for exactly calculating each w_e for single-task learning exist [10]. The method breaks down into three steps:

1. Calculate the family scores from data. These are called the β functions, $\beta_i(\pi_i) = P(\pi_i)P(x_i|\pi_i)f_i(\pi_i)$. It is assumed that the computational complexity of each of these is some function $C(n)$ that depends on the number of samples n. The maximum number of parents allowed for any node is typically fixed to a small natural number, r. Therefore, there are $O(p^{r+1})$ of these functions to calculate for a total computational complexity of $O(p^{r+1}C(n))$.

2. Calculate the local contribution of each subset $U \subseteq V - \{i\}$ of potential parents of i. These are called the α functions, $\alpha_i(U) = \sum_{\pi_i \subseteq U} P(\pi_i)P(x_i|\pi_i)f_i(\pi_i)$. There are an exponential number of subsets U, therefore there are an exponential number of α functions. Using a truncated fast Möbius transform [2], all of the α functions can be computed in $O(p2^p)$ time, assuming that the β functions are pre-computed and that there is a limit, r, on the maximum size of the parent sets.

3. Sum over the subset lattice of the various U_i to obtain the sum over orders $\prec$. Although the number of orders is p! there is no need to enumerate each order explicitly. The potential parents of each node i depend only on the set of parents U_i that precede it, not on the ordering of the parents within U_i. Using dynamic programming, this sum takes time $O(p2^p)$ [10].

The total computational complexity for a single task is $O(p2^p + p^{r+1}C(n))$. This is the exact calculation of the posterior. For large networks, roughly $p > 30$, the exponential term is intractable. In these cases, we can use MCMC to approximate the sum over orders [18]. To limit the computation of the polynomial term, we can choose a sufficiently small r or further reduce the number of potential families using candidate parent sets [8].

To learn multiple Bayesian networks simultaneously, we replace the single-task prior $P(\pi_i)$ with a transfer bias $P(\pi_i^{(k)}, \pi_i^{(j)})$ that shares information among tasks k and j [19]. The transfer bias penalizes the number of parents in $\pi_i^{(k)}$ that are not also in $\pi_i^{(j)}$ for all pairs of tasks (k, j). This transfer bias encourages similar graph structures to be learned for each task, and has been shown to produce more robust networks [17, 19]. In terms of the three-step method above [10], this means replacing the β functions with [19]:

$$\begin{aligned}\beta_{ki}(\pi_i, \lambda_2) =& f_i(\pi_i^{(k)}) P(x_i^{(k)}|\pi_i^{(k)}) P(\pi_i^{(k)}, \pi_i^{(j)}) \\ =& f_i(\pi_i^{(k)}) P(x_i^{(k)}|\pi_i^{(k)}) \times \frac{1}{(K-1)(4-\lambda_2)^{|U_i|}} \times \\ & \left[\sum_{j \neq k} \sum_{\pi_i^{(j)} \subseteq U_i} P(x_i^{(j)}|\pi_i^{(j)}) (1-\lambda_2)^{\Delta(\pi_i^{(k)}, \pi_i^{(j)})} \right] , \end{aligned} \tag{5}$$

where $\Delta(\pi_i^{(k)}, \pi_i^{(j)}) = |\pi_i^{(k)} \setminus \pi_i^{(j)}|$. We assume each β function takes time $C(n_k)$ to compute, where n_k is the number of samples in task k. Under transfer learning, there is a now a sum over parent sets for each task, therefore the computational complexity is $O(Kp^r)$ for each β function. There are Kp^{r+1} of these functions to calculate. This gives a total computational complexity for all multitask β functions of $O(K^2 p^{2r+1})$. Note that for visualization and interactive purposes, the number of tasks is typically $K = 2$ for ease of end-user interpretation of the results.

Once the multitask β functions are calculated, the rest of the posterior estimate can be calculated using existing algorithms, such as exact expectation over orders [10, 21] or MCMC approximations [18].

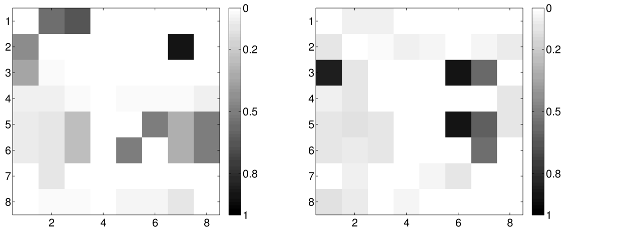
4.2 Efficient Computation of Transfer Bias

We store intermediate calculations to speed up any future calculations with different values for λ_2. We achieve this by noting that the function Δ can only produce a finite number of integer values in the range $[0, r]$. By grouping the parents sets, we can re-arrange terms to group together the parent sets $\pi_i^{(j)}$ that will produce the same value in the Δ function.

$$\begin{aligned}&\sum_{\pi_i^{(j)} \subseteq U_i} P(x_i^{(j)}|\pi_i^{(j)}) (1-\lambda_2)^{\Delta(\pi_i^{(k)}, \pi_i^{(j)})} = \\ &= \sum_{\delta=0}^{r} \sum_{\pi_i^{(j)} | \Delta(\pi_i^{(k)}, \pi_i^{(j)}) = \delta} P(x_i^{(j)}|\pi_i^{(j)}) (1-\lambda_2)^{\delta} \\ &= \sum_{\delta=0}^{r} (1-\lambda_2)^{\delta} \sum_{\pi_i^{(j)} | \Delta(\pi_i^{(k)}, \pi_i^{(j)}) = \delta} P(x_i^{(j)}|\pi_i^{(j)}) \end{aligned} \tag{6}$$

By separating the sum over individual scores, we can store the sums and re-use them later if λ_2 changes. We define the γ functions as these sums:

$$\gamma_{ki\delta}(\pi_i, \delta) = \sum_{j \neq k} \sum_{\pi_i^{(j)} | \Delta(\pi_i^{(k)}, \pi_i^{(j)}) = \delta} P(x_i^{(j)}|\pi_i^{(j)}) \ , \quad \text{for all } \pi_i \subseteq V - \{i\},\ \delta \in \mathbb{Z},\ 0 \leq \delta \leq r \ . \tag{7}$$



(a) W_1

(b) W_2

Figure 4: Estimated posterior likelihoods for two tasks with $\lambda_2 = 0$. There are 8 variables, and 8×7 possible directed edges, which are organized as a weighted adjacency matrix.

With a maximum parent set size r, the maximum value that δ can take is r. Therefore, the number of γ functions to be calculated are: Krp^{r+1}, one for every family in every task for every value of δ. The calculation of all of these γ functions is $O(K^2 r p^{2r+1} C(n))$.

We rewrite the β functions using the pre-computed γ functions. Notice that the computational complexity of the β function is now linear in r. This means that the functions can be computed quickly for various values of λ_2.

$$\beta_{ki}(\pi_i, \lambda_2) = \frac{f_i(\pi_i^{(k)}) P(x_i^{(k)}|\pi_i^{(k)})}{(K-1)(4-\lambda_2)^{|U_i|}} \cdot \sum_{\delta=0}^{r} (1-\lambda_2)^{\delta} \gamma_{ki\delta}(\pi_i, \delta) \tag{8}$$

These γ functions are also used in the calculation of the Jacobian.

4.3 Thresholding for graphs

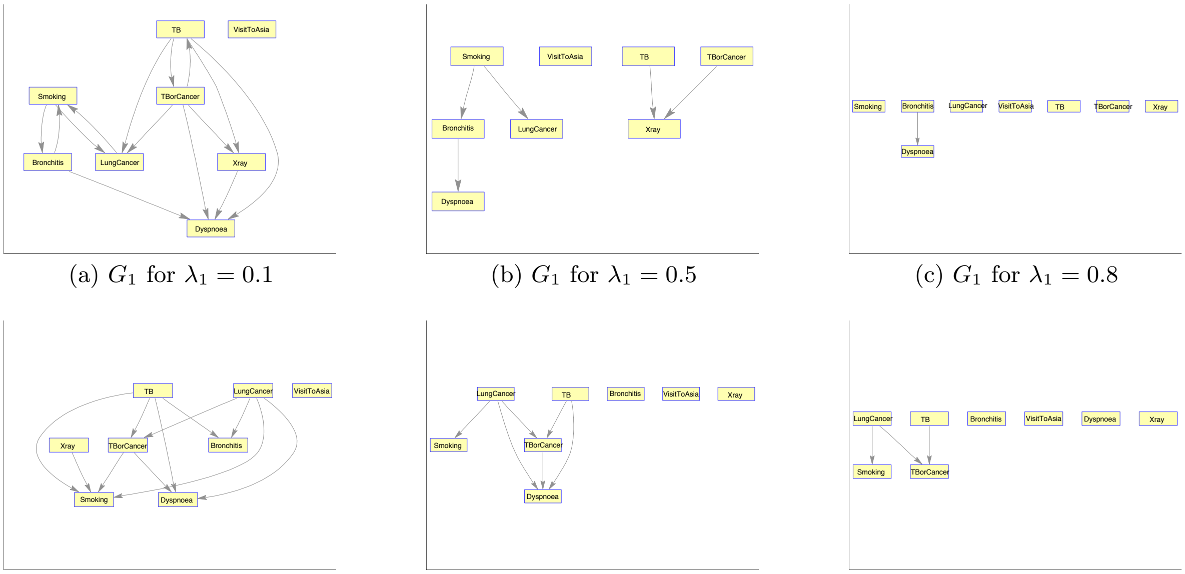
The feature probabilities, w_{ek}, learned from Equation 4 can be organized into square matrices W_k for each task k representing the directed edges of a network. Figure 4 shows an example of these learned feature posterior probabilities.

In order to display graphs to the user (see Figure 5), we threshold the w_{ek} values, showing only the edges with likelihoods greater than some cut-off value $0 \leq \lambda_1 \leq 1$. Clearly, λ_1 will control the density of edges in the displayed graphs. In this work, we employ a soft-threshold sigmoid function to define the learned graph:

$$G_{ek} = \frac{1}{1 + \exp[-\beta(w_{ek} - \lambda_1)]} \ . \tag{9}$$

For sufficiently large values of $\beta > 1$ this is equivalent to a hard threshold at λ_1.

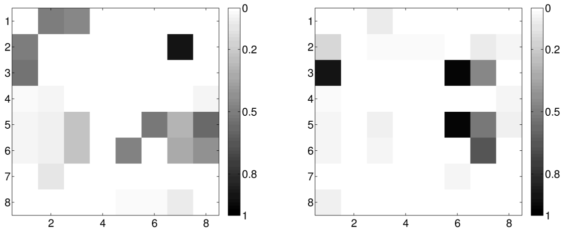
Mathematically, this thresholding of edge expectations is a loss of information. For comparative network analysis, it may seem desirable to keep all edges weighted by their expectation. However, from the end user perspective, nearly all graph visualization systems allow thresholding out the weakest edges to get a clearer picture of the network, and so we treat thresholding as a necessary step for visualization. When comparing the similarities and differences among a set of graphs, it is helpful to be able to control for the number of



(a) G_1 for $\lambda_1 = 0.1$ (b) G_1 for $\lambda_1 = 0.5$ (c) G_1 for $\lambda_1 = 0.8$

(d) G_2 for $\lambda_1 = 0.1$ (e) G_2 for $\lambda_1 = 0.5$ (f) G_2 for $\lambda_1 = 0.8$

Figure 5: By thresholding at λ_1, we obtain graphs G_k from the weighted adjacency matrices W_k.



(a) W_1 (b) W_2

Figure 6: Estimated posterior likelihoods for two tasks with $\lambda_2 \neq 0$. There are 8 variables, and therefore 8×7 possible directed edges, which we have organized into a weighted adjacency matrix.

differences between the graphs. By encouraging the graphs to be similar, we can reduce the number of spurious differences learned, and display only differences that are most likely to be real (see Figures 6 and 7). The λ_2 parameter controls the amount of similarity bias which encourages the w_{ek} and w_{ej} values of tasks k and j to be close, as in Figure 6. This means that as the w values move closer together, some will cross the threshold, as in Figure 7, therefore, both parameters will have a noticeable effect on both the number of edges learned and the number of differences.

5. NUMERICAL ESTIMATION OF HYPER-PARAMETERS

The previous section showed how to learn multiple Bayesian networks given data and hyper-parameter settings. This section outlines our method for incorporating user preferences into the learning algorithm. Once feedback has been received from the user, the hyper-parameters $\Lambda(G, S)$ need to be updated. This is computationally expensive, and so we lay out a numerical estimation of Λ.

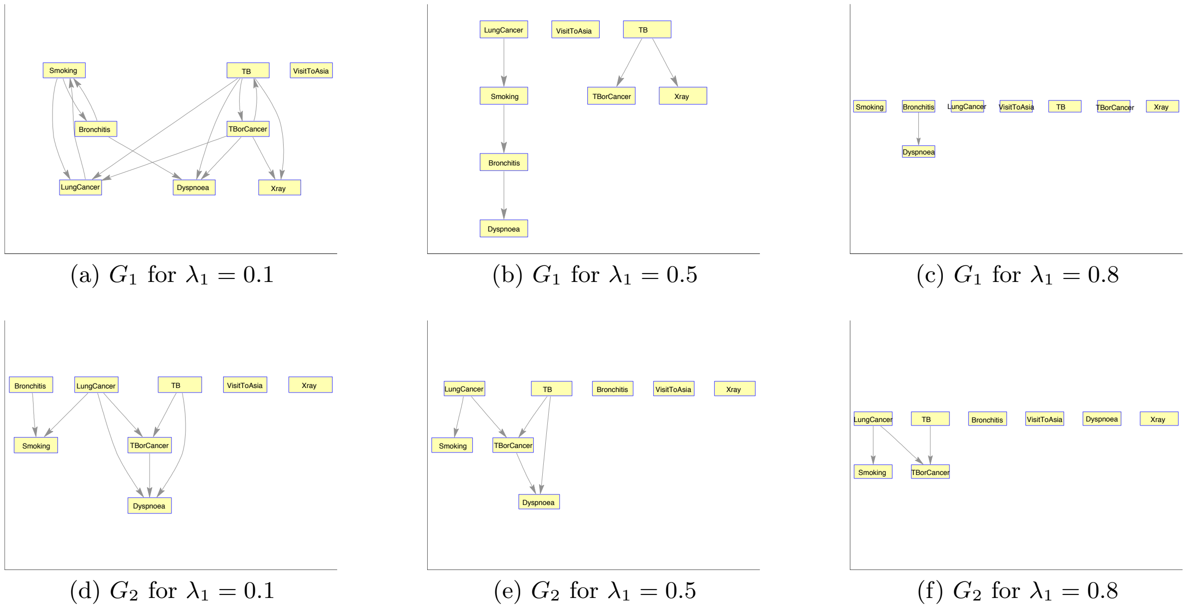
5.1 Estimation of Λ for Multitask Bayesian Networks

To re-learn graphs after getting feedback from the user, we take one step toward minimizing the distance between the learned graphs an the given user preferences, as in Equation 1. First, we need to calculate the Jacobian $\nabla_\Lambda G_{ek}(\Lambda)$ in Equation 2. For multitask Bayesian networks, the partial derivative with respect to λ_1 is fairly straightforward.

$$\begin{aligned} \frac{\partial}{\partial \lambda_1} G_{ek} &= \frac{-\beta e^{-\beta(w_{ek}-\lambda_1)}}{1+e^{-\beta(w_{ek}-\lambda_1)}} \\ &= -\beta e^{-\beta(w_{ek}-\lambda_1)} G_{ek} \end{aligned} \tag{10}$$

The partial derivative with respect to λ_2, on the other hand, is more complicated to calculate because the family scores within the sums depend on λ_2. Therefore, the partial derivative of each of these family scores must be computed, and the sums re-calculated.

$$\begin{aligned} &\frac{\partial}{\partial \lambda_2} G_{ek} = \\ &= -\beta e^{-\beta(w_{ek}(\lambda_2)-\lambda_1)} G_{ek} \sum_{\prec} \sum_{\pi_e^{(k)} \subseteq U_e} f_e(G_k) P(x_e^{(k)} | \pi_e^{(k)}) \times \\ &\left[\sum_{\pi_e^{(j)} \subseteq U_e} P(x_e^{(j)} | \pi_e^{(j)}) \frac{-\Delta(1-\lambda_2)^{\Delta-1} + |U_i|(1-\lambda_2)^{\Delta}(4-\lambda_2)^{-1}}{(4-\lambda_2)^2} \right] \\ &= -\beta e^{-\beta(w_{ek}(\lambda_2)-\lambda_1)} G_{ek} \sum_{\prec} \sum_{\pi_e^{(k)} \subseteq U_e} f_e(G_k) P(x_e^{(k)} | \pi_e^{(k)}) \times \\ &\left[\sum_{\pi_e^{(j)} \subseteq U_e} P(x_e^{(j)} | \pi_e^{(j)}) \frac{(1-\lambda_2)^{\Delta_{ikj}}}{(4-\lambda_2)^2} \cdot \left(\frac{|U_i|}{4-\lambda_2} - \frac{\Delta_{ikj}}{1-\lambda_2} \right) \right] \end{aligned} \tag{11}$$



(a) G_1 for $\lambda_1 = 0.1$ (b) G_1 for $\lambda_1 = 0.5$ (c) G_1 for $\lambda_1 = 0.8$

(d) G_2 for $\lambda_1 = 0.1$ (e) G_2 for $\lambda_1 = 0.5$ (f) G_2 for $\lambda_1 = 0.8$

Figure 7: By thresholding at λ_1, we obtain graphs G_k from the weighted adjacency matrices W_k with $\lambda_2 > 0$.

This can be re-written using the pre-computed γ functions.

$$\frac{\partial}{\partial\lambda_2}G_{ek} = \\ = -\beta e^{-\beta(w_{ek}(\lambda_2)-\lambda_1)}G_{ek}\sum_{\prec}\sum_{\pi_e^{(k)}\subseteq U_e} f_e(G_k)P(x_e^{(k)}|\pi_e^{(k)})\times \\ \left[\sum_{\delta=0}^{r}\frac{(1-\lambda_2)^{\delta}}{(4-\lambda_2)^2}\cdot\left(\frac{|U_i|}{4-\lambda_2}-\frac{\delta}{1-\lambda_2}\right)\gamma_{ke\delta}(\pi_e^{(k)},\delta)\right] \tag{12}$$

The minimum step size is η such that $\Lambda^{\text{new}} = \Lambda - \eta\nabla_\Lambda g$ gets $G(\Lambda^{\text{new}})$ one edge closer to S.

$$\sum_{e,k}|S_{ek} - G_{ek}(\Lambda^{\text{new}})| - \sum_{e,k}|S_{ek} - G_{ek}(\Lambda)| = -1 \tag{13}$$

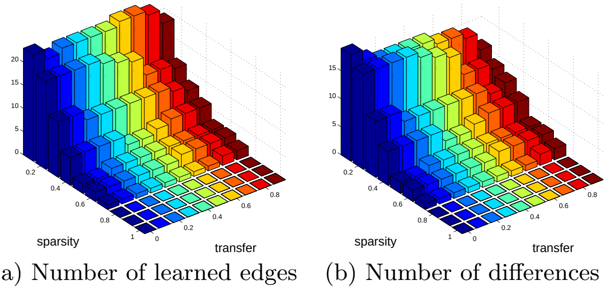
We solve for η using binary search until the above criteria is met.

6. DISCUSSION

There is strong motivation for creating interactive human-in-the-loop algorithms for exploring comparative dependency networks. Here we discuss our initial findings on benchmark networks, share case studies on real data and then suggest directions for future work.

6.1 Demonstration on Benchmark Networks

We use the benchmark *Asia* network to explore the practicality of this interactive approach. The *Asia* network contains 8 discrete variables [12]. In order to produce multiple networks with some edges different, we randomly delete each edge with probability $p = 0.1$. If an edge is deleted, the conditional probability table for the child is modified by summing over the removed parent. This produces a set of networks that are similar to the original *Asia* network but with a few edges different.



(a) Number of learned edges (b) Number of differences

Figure 8: Modified *asia* networks: summary statistics about learned network models for various values of sparsity and transfer hyper-parameters.

Using two tasks, and starting with an initial value for $\Lambda = [0.5, 0]$, we learn networks $\mathbf{G}$. Then simulated feedback responses, $\mathbf{S}$, are given by randomly choosing a user action at each stage. For each of these feedback matrices, we track the movement in Λ to investigate the effect of $\mathbf{S}$ on Λ. For comparison, we also perform a grid search by running the multitask network learning algorithm for combinations of settings of λ_1 and λ_2 evenly spaced in the valid range for each parameter. Figure 8 shows results of a grid search for one set of data, with 100 samples drawn from modified *Asia* networks. As expected, neither the number of edges nor the number of differences learned vary linearly with the input hyper-parameters. Whereas, by design, the interactive algorithm takes steps evenly in terms of the number of edges or differences learned.

One question is whether the gradient direction is typically aligned with just one hyper-parameter, or if it is usually more "diagonal". If it is typically aligned with just one hyper-parameter, then we could adjust each parameter independently rather than calculating the gradient. We observe that it is typically diagonal (takes a step in both λ_1 and λ_2 directions), however there are some cases where the gradient direction is nearly zero for either λ_1 or λ_2.

It is difficult to ascertain the interestingness of a solution for these benchmark networks. We have shown that grid search covers objectively uninteresting solutions; in the form of redundant solutions, overly dense solutions and empty solutions. These benchmark networks are not from a real domain (or it is an overly simplistic domain), therefore there is not a practical way to judge the subjective interestingness of the solutions to an uninteresting benchmark problem. To analyze the usefulness of the interactive algorithm from the end-user perspective, we therefore rely on case studies from real data.

6.2 Case Studies

Results on both neuroimaging and protein studies were presented to domain scientists using our interactive comparative network visualization. In both cases, a machine learning expert initially loaded the result networks into the visualization system and then manned the controls for adjusting the number of edges and the number of differences. After a few minutes of looking through network solutions with various numbers of edges and differences, the domain experts typically made requests, such as to see "the highest confidence edges shared by both tasks." The domain experts were able to take over the controls themselves and expressed appreciation for being able to visualize so many solutions quickly.

Anecdotally, we find that different domain experts are interested in different levels of confidence in edges and differences. For the neuroimaging study, the domain expert was most interested in extremely high confidence differences, selecting difference networks with only three dependencies in each. On the other hand, the biologists looking at protein data were interested in difference networks with 100 dependencies. These two anecdotes support the idea that different users could have different inexpressible objective functions in mind. However, we need to have different domain experts analyze the same data to see if the various interests are due to the users or if it is inherent in the data.

Often in machine learning, the goal is to find the single best solution to a problem. However, while looking through the various solutions produced by different hyper-parameter settings, the domain experts did not ask how to select the single best solution. They fully understand the concept of exploring the precision-recall tradeoff. Yet, they did ask whether there is any way to get a confidence interval for the dependencies and differences. Instead of adjusting the number of edges/differences, they would find it preferable to be able to quantify the confidence of edges/differences.

6.3 Future Work

This paper provides a framework for creating interactive multitask graph structure learning algorithms. These algorithms remain computationally challenging. The Bayesian posterior distributions on multiple Bayesian networks, in particular, do not scale well to large networks. The scalability problem is endemic to the problem of Bayesian network learning. Performing updates in real-time for large networks will be computationally difficult. We could alleviate this problem through the use of approximate or heuristic network structure learning. Doing so requires extensive evaluation on the tradeoffs between speed and accuracy, and so we leave this for future work.

Other graph learning algorithms, such as graphical lasso [15, 4], scale to large networks much better than Bayesian networks. Therefore, we would like to apply the proposed interactive method to multitask graphical lasso. However, the graphical lasso objective with respect to λ_1 and λ_2 is discontinuous; therefore, the gradient (Equation 2) is undefined at precisely the points that we care about. Currently, we are investigating numerical approximations to the regularization path or heuristics for finding the discontinuous "hinge" points quickly. Such algorithms that calculate the regularization path for individual networks have been developed [6, 22]. However, there is not any such algorithm for multitask network learning.

Typical grid search methods are inefficient and information criteria based tuning guidelines often are not ideal. Interactive guidance provides fine-grained control over exploration of the solution space in those areas that are of highest interest to the user. We could take a hybrid approach, first computing results over a coarse grid, then giving the user the ability to take small local steps or to jump to another area of the hyper-parameter space using the pre-computed results.

Other forms of feedback could be incorporated rather than just increasing or decreasing the number edges and differences. For example, one request from domain scientists is being able to query a specific edge, and see what the whole network looks like at the threshold point where that edge appears. A similar query could be imagined for edge differences. These type of queries should be straightforward to implement algorithmically. The challenge is in creating a user interface to gather this type of feedback. Working closely with domain scientists, we could find other queries that would make exploring solutions easier for the user.

The interactive approach presented here assumes that a human will guide the objective function via feedback about the hyper-parameters. However, the idea of beginning at an initial point in the solution space and exploring solutions by modifying hyper-parameters could be accomplished without a human. A virtual user that begins with no transfer and repeatedly requests fewer differences, is essentially an automated process for exploring the regularization path along the "differences" axis. The result of such a solution path is a ranking of the strength of the differences found. Therefore, the updates to the algorithm presented in this paper could be used as steps in an automated iterative algorithm, instead of an interactive human-in-the-loop algorithm. This is an interesting direction to explore to see whether the human users or the automated approaches are more effective at finding interesting solutions.

7. CONCLUSIONS

The concept of interactive network comparison is compelling. The hypothesis space is large and the learned models are complex. Presenting only a single solution (even if it fits the data well) is unsatisfactory. Yet, it is not easy to display all possible solutions at once and summary statistics about the potential patterns only tell a part of the solution. Graphical models are frequently used in knowledge discovery because they help to quickly visualize complex patterns of dependency. In an increasingly interactive world, it is frustrating to the end user to see static results of a learning algorithm and not able to explore alternative solutions on the fly. Therefore, human-in-the-loop interaction is necessary for comparative dependency network learning. The first challenge in making a machine learning algorithm interactive, is to translate user feedback into changes in hyper-parameters that control the learning algorithm. The second challenge is to efficiently update results to be seen in real-time. We introduce a framework for interactive multitask graph structure learning with a specific implementation of multitask Bayesian networks and show that the results are preferable to the standard grid search over hyper-parameter space. In practice, all machine learning applications involve some form of interaction between looking at results and adjusting the algorithm to investigate alternative results. Automating this interactive process allows domain scientists and other end users to work more efficiently while discovering patterns in their data.

8. REFERENCES

[1] S. Amershi, J. Fogarty, A. Kapoor, and D. Tan. Effective end-user interaction with machine learning. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[2] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: fast subset convolution. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, pages 67–74. ACM, 2007.

[3] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, 2009.

[4] P. Danaher, P. Wang, and D. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *arXiv stat.ME*, 1111(00324v1), November 2011.

[5] A. de la Fuente. From 'differential expression' to 'differential networking' – identification of dysfunctional regulatory networks in diseases. *Trends in Genetics*, 26(7):326 – 333, 2010.

[6] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[7] N. Friedman and D. Koller. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1):95–125, 2003.

[8] N. Friedman, I. Nachman, and D. Peér. Learning Bayesian network structure from massive datasets: the sparse candidate algorithm. In *Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 206–215, 1999.

[9] A. Kapoor, B. Lee, D. Tan, and E. Horvitz. Performance and preferences: Interactive refinement of machine learning procedures. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[10] M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5:549–573, 2004.

[11] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques.* Adaptive Computation and Machine Learning. MIT Press, 2009.

[12] S. Lauritzen and D. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224, 1988.

[13] H. Liu, K. Roeder, and L. Wasserman. Stability approach to regularization selection (stars) for high dimensional graphical models. In *Neural Information Processing Systems*, 2010.

[14] S. Maslov and K. Sneppen. Specificity and stability in topology of protein networks. *Science*, 296(5569):910, 2002.

[15] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, June 2006.

[16] K. Mohan, M. Chung, S. Han, D. Witten, S.-I. Lee, and M. Fazel. Structured learning of Gaussian graphical models. In *Advances in Neural Information Processing Systems 25*, pages 629–637. 2012.

[17] A. Niculescu-Mizil and R. Caruana. Inductive transfer for Bayesian network structure learning. In *Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.

[18] T. Niinimaki, P. Parviainen, and M. Koivisto. Partial order MCMC for structure discovery in Bayesian networks. In *Twenty-Seventh Annual Conference on Uncertainty in Artificial Intelligence*, pages 557–564, 2011.

[19] D. Oyen and T. Lane. Bayesian discovery of multiple Bayesian networks via transfer learning. In *IEEE International Conference on Data Mining*, 2013.

[20] D. Oyen, A. Niculescu-Mizil, R. Ostroff, and A. Stewart. Controlling the precision-recall tradeoff in differential dependency network analysis. In *The Seventh Workshop on Machine Learning in Systems Biology*, 2013.

[21] P. Parviainen and M. Koivisto. Exact structure discovery in Bayesian networks with less space. In *Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 436–443, 2009.

[22] M. Schmidt, A. Niculescu-Mizil, and K. Murphy. Learning graphical model structure using l1-regularization paths. In *National Conference On Artificial Intelligence*, volume 22, page 1278. AAAI Press, 2007.

[23] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003.

[24] T. Van Allen and R. Greiner. Model selection criteria for learning belief nets: An empirical comparison. In *Seventeenth International Conference on Machine Learning*, pages 1047–1054, 2000.

[25] A. V. Werhli, M. Grzegorczyk, and D. Husmeier. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models and Bayesian networks. *Bioinformatics*, 22(20):2523–2531, 2006.

Interactive Exploration of Larger Pattern Collections: A Case Study on a Cocktail Dataset

Daniel Paurat
University of Bonn
daniel.paurat@uni-bonn.de

Roman Garnett
University of Bonn
rgarnett@uni-bonn.de

Thomas Gärtner
University of Bonn and Fraunhofer IAIS
thomas.gaertner@uni-bonn.de

ABSTRACT

We present a general method for employing interactive embedding techniques to enable an analyst to explore a larger collection of local patterns. The common idea among pattern-mining methods is to list descriptions of subsets of a dataset according to some interestingness measure. Because the space of all patterns in a dataset is exponentially large in the number of attributes, most pattern-mining algorithms reduce the output for the analyst to a small set of highly interesting and diverse patterns. However, by discarding most of the patterns, these methods have to make a trade-off between ruling out potentially insightful patterns and possibly drowning the analyst in results. We propose an alternative. To counteract information overload, we mine a rather large set of patterns and study this collection using an interactive embedding technique. Using this interactive, visually driven exploration technique, the analyst can develop an understanding of the patterns, their distribution, the concepts underlying them, and how they interrelate.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Interaction styles*

General Terms

Rule and pattern mining, Exploratory analysis

1. INTRODUCTION

We propose an extension to the classical pattern-mining approach. Our idea is to not focus on condensing the resulting output to a small set of high-quality patterns, but rather to visually explore the distribution of a larger collection of patterns as a whole. To do so, we empower the analyst to actively steer the perspective of a two dimensional projection of the mined patterns. Altering the perspective and seeing how related patterns move, zooming and filtering the collection and inspecting structures of interest closer, lets the analyst keep the overview even on larger pattern collections.

Classical pattern-mining algorithms, like closed frequent item set mining, subgroup discovery, and exceptional model mining—to name just a few—search for patterns of high interest to the analyst in a dataset. The goal is to retrieve a small collection of easy-to-understand patterns that expose main concepts occurring frequently within the dataset. In Section 2 we briefly discuss several pattern-mining algorithms and their main objectives. The formal definition of a pattern, how its interestingness is measured, and how the final result is compiled differs from method to method. In general, one can say that a pattern is a description of a subset of the dataset that should be easy to understand. A very commonly used pattern format, which is also used throughout this paper, is the conjunction of different *attribute=value* assignments. For instance, the pattern *"type=fish and color=blue"* describes all blue fishes in a dataset at hand. The result set that is finally delivered to the analyst is usually determined by considering the support of the patterns, a quality measure, and the redundancy among the patterns of the result set. To keep the result set at a convenient size, classical pattern-mining algorithms have to carefully consider whether the information in each pattern bears insight or might contribute to overload.

We propose an interactive, visually driven extension to the classical pattern-mining procedure that does not discard any discovered patterns before presenting the results to an analyst. The idea is not to deliver a condensed result set, but rather to mine a larger collection of patterns first and then project them into a two-dimensional space, with similar patterns being close to each other. This enables further visual analysis. The insights gained from actively exploring the pattern distribution help the analyst to understand and interpret the results of the classical pattern-mining methods. The exploration of the pattern distribution follows Shneiderman's information-seeking mantra *"Overview first, zoom and filter, then details-on-demand"* [30]. Our proposed approach enables the analyst to grasp the pattern collection as a whole and then to further discover and dig deeper into regions of interest. In earlier publications, we investigated different algorithms that enable direct interaction with an embedding to explore a dataset interactively. The direct visual feedback of seeing how the distribution of all data records changes upon interaction can help the analyst understand the underlying structure of the data and formulate hypotheses. One common way to provide the interface for the interaction is to let the analyst select data points as control points and relocate them in a "drag-and-drop" manner within the embedding. Altering the positions of these control points triggers the embedding technique to recalculate the whole projection, subject to the updated control-point locations. The recalculation can usually be done efficiently, such that the updates resulting from the interaction can be rendered live. For an impression on the update-rate of the here used implementation please have a look at Appendix A.1. Note that there are also other methods of interacting with an embedding, e.g., employing *must-link* / *cannot-link* constraints, filtering and inspecting the sub-selection, or simply highlighting and brushing.

The remainder of the paper is organized as follows. In Section 2 we discuss related research and in Section 3 we introduce a general framework for interactive pattern exploration. Section 4 demonstrates our approach in several scenarios on a cocktail ingredient dataset before we finally conclude in Section 5.

2. RELATED WORK

Related to our work are basically two areas of research, pattern mining and interactive embedding methods. For the pattern-mining methods we have to distinguish whether a label is considered. Probably the most-known pattern-discovery technique that does not consider a label is frequent item set mining. Here all conjunctions of *attribute=value* assignments are listed in decreasing order of the number of data records that support the pattern [1, 15]. Because the set of all 1-frequent patterns of a dataset can be exponentially large in the number of attributes of the dataset, usually only the top-k patterns with a thresholded minimum support are considered. However, often the set of frequent patterns contains redundant descriptions; i.e., the same set of data records is described by different patterns. Closed frequent items-set mining methods [4, 32] counteract this by only listing the closure of each of these sets as a unique descriptor. Other ways to discover interesting patterns in an unlabeled dataset are, e.g. to compile an output set of small size that possesses a high entropy [25] or to find large tiles of 1-assignments in a binary dataset [12].

For labeled datasets, (closed frequent) subgroup-discovery algorithms [18, 20, 33] find patterns with a significant difference between the label distribution of the whole dataset and the one exposed by the patterns. Exceptional model mining [21], a generalization of subgroup discovery, allows for more-complicated target concepts, like multiple labels. Another generalization applies the theory of relevance [11, 19] to the found subgroups. Relevant subgroup discovery algorithms [11, 13, 24] deliver only patterns that are not covered by any other pattern in the result set. The term 'covering' implies that there is no generalization of a subgroup that extends the subgroup's support set by strictly positively labeled data records. Δ- and ϵ-relevant subgroup discovery methods [14, 23] loosen this tight formulation and allow the considered generalizations of a subgroup to have a controlled amount of additional negatively labeled data records in the support set.

A different approach to the discovery of interesting patterns is to sample from the space of all patterns. Note that pattern sampling does not aim at delivering a condensed result set, but instead samples the patterns with a probability proportional to a given interestingness measure. Possible measures are, e.g., sampling proportional to a pattern's frequency, its squared frequency, its lift, or the area it tiles in the dataset [5]. In addition, pattern sampling can also take labels into account, such that patterns with a high positively and a low negatively labeled share in the support set are more likely to be drawn. The probability of a pattern being drawn can be calculated efficiently [6] by using the sampling technique *coupling from the past*.

Pattern sampling is a good showcase to demonstrate our approach and can be a good option in cases where the space of all patterns is extremely large, such that classical pattern-mining algorithms take too long to terminate. This is especially important if the analyst is on a time budget and the listing strategy of the mining algorithm does not correlate with the relevance of the patterns to the analyst. In this case, sampling from the whole pattern space can yield interesting patterns much earlier. Boley et al. [5] show such an example on the *primary-tumor* dataset, where the patterns that are most discriminating between the labels are among the least-frequent.

Apart from local pattern discovery, there is also related work in the area of embedding data into a lower-dimensional space for visualization and interaction. Many classic techniques are unsupervised and static, like the well known principle component analysis (PCA) [16], multi-dimensional scaling (MDS) [8], isometric mapping [31] and locally linear embedding [29]. These methods consider the distances between the data records in different ways and find lower dimensional embeddings which exhibit similar the distance relations. The projection pursuit method [10] follows a different objective, it searches for interesting projections of the data that display a high degree oy non-gaussianity.

In order to incorporate interaction into the dimensionality-reduction algorithms, the static embedding approaches are typically extended to consider additional user feedback and thus provide an interface with the lower-dimensional embedding of the data to the analyst. There are different approaches for deriving the embedding and incorporating interaction. Some techniques enable the user to relocate selected points within the embedding and incorporate the placement of these control points as constraints or regularization into the optimization problem of a (kernelized) principal component analysis (PCA) [28, 26]. Other techniques embed the data via MDS user-suggested locations of the control points [7, 9, 22]. In contrast to these methods, least squared error projections [27] calculate the embedding solely by considering the control points' original attributes and user-specified embedding locations, ignoring the covariance among the rest of the data records. The interactive embedding technique used in our upcoming study in Section 4 minimizes the uncertainty of the resulting embedding, given a prior belief about it, conditioned on the control points' placements [17]. Throughout this paper we refer to this technique as *most-likely embedding* (MLE). In addition, this method can also be used to actively propose control points to the analyst that minimize the uncertainty about the resulting embedding and thus should be placed next.

Finally, but without a focus on interaction, Berardi et al. proposed to embed collections of patterns in order to discover structures among them by using MDS as the embedding technique [3]. The pairwise similarities between the patterns, required by MDS, were derived by calculating the Jaccard index between two patterns.

3. A GENERAL INTERACTIVE PATTERN EXPLORATION PROCEDURE

Our approach to studying a larger collection of patterns is to embed them into a lower-dimensional space for further interactive visual analysis. Due to the many different ways this can be done, we do not want to propose one particular exploration technique, but rather give a guiding framework on how to gain insights from a larger pattern collection by exploring it interactively. Our proposed procedure comprises the following steps:

1. Mine a large collection of patterns.
2. Represent the patterns in a canonical way as vectors.
3. Embed these vectors with an interactive embedding method and explore the pattern distribution.
4. Inspect the emerging structures of interest deeper.

In our upcoming exemplary study, we utilize a two-dimensional scatterplot for visualization, with each pattern being a point within the plot. Often the initial visualization of the pattern distribution, before any interaction at all, already exhibits interesting structures that invite the analyst to deeper inspection. By further interacting with the embedding by, e.g., selecting single patterns as control

Table 1: Exemplary results of the ten highest quality patterns, delivered by different pattern-mining approaches on the cocktail dataset. Note that here the top-10 frequent item sets are also all closed. The high-lift patterns were sampled according to their *rarity* measure [6]. In case of subgroup discovery, the label indicates whether a cocktail is creamy or not.

Unsupervised pattern-mining methods		Supervised pattern-mining methods	
Frequent (closed) item sets	**Sampled patterns with high lift**	**closed subgroups**	**Δ_1-relevant subgroups**
Vodka	Vodka **&** Cranberry juice	Baileys	Baileys
Orange juice	Vodka **&** Triple sec	Crème de cacao	Crème de cacao
Amaretto	Baileys **&** Kahlúa	Milk	Milk
Pineapple juice	Vodka **&** Gin	Kahlúa	Kahlúa
Grenadine	Vodka **&** Blue curaçao	Baileys **&** Kahlúa	Cream
Gin	Pineapple juice **&** Malibu rum	Cream	Irish cream
Baileys	Vodka **&** Amaretto	Irish cream	Crème de banana
Tequila	Vodka **&** Rum	Vodka **&** Baileys	Butterscotch schnapps
Kahlúa	Orange juice **&** Amaretto	Crème de banana	Whipped cream
Triple sec	Vodka **&** Tequila	Baileys **&** Butterscotch schnapps	Vodka **&** Kahlúa

points and relocating them in a playful manner, the analyst can see how other patterns relate, as they move accordingly. On the other hand, the analyst does not have to 'play' with the embedding, but can also directly express desired similarities among patterns by selecting similar ones and placing them close to each other in the embedding. In this way the analyst can also incorporate domain knowledge into the embedding. The above mentioned structures that occur in the visualization can come in various shapes; clusters of patterns, regions of higher density, outliers, or mirroring shapes can all be fruitful to investigate. Reasoning about the contents of these structures and how they differ from another usually uncovers interesting aspects about the patterns and the original dataset.

4. AN EXEMPLARY STUDY

In this section, we demonstrate the use of our interactive pattern-exploration approach by performing an artificial exemplary knowledge discovery session on a cocktail-ingredient dataset. The data is an excerpt of the drinks presented on the website webtender.com. It can be downloaded, together with our interactive embedding tool from http://kdml-bonn.de/InVis. In the following we give an example of a concrete instantiation of the above introduced framework. This setup is precisely the workflow that we use in our exemplary study in Section 4.1. For the other examples in Sections 4.2 and 4.3, only the first step changes, as the pattern collection is retrieved using different algorithms.

1. Mine the 1000 most-frequent item sets from the cocktail dataset. Here, every cocktail is described as the set of ingredients it contains.

2. Represent each of the 1000 frequent item sets by a binary vector over all occurring items of the pattern collection in lexicographical order.

3. Visualize the pattern vectors, using the *most-likely embedding* technique with an initial PCA embedding as the prior mean and interact with it to shape out interesting structures.

4. Inspect these structures by highlighting patterns that contain certain ingredients and by listing the five most-present single items of the structure in a tag cloud.

A list of the ten highest-quality patterns, found by several classical pattern-mining algorithms, is given as a reference in Table 1. The first three methods, *frequent*, *closed frequent*, and sampled *high-lift* patterns, do not consider label information, but provide us with an overview on the most-striking ingredients and ingredient combinations. The *subgroup-* and *relevant-subgroup-discovery* methods on the other hand do use a label and show us ingredients (and their combinations) that are strongly related to it. For these methods, we manually assigned a label to each cocktail according to whether it is "creamy". In Sections 4.1, 4.3 and 4.2 we will apply our interactive approach on the output of different pattern-mining algorithms with the goal of gaining additional insights into the results of Table 1 and to understand the patterns' relations. In each session we mine 1000 patterns and represent them as binary vectors over all items that occur within the patterns, sorted in lexicographical order. We then visualize the mined patterns using an interactive embedding technique and search for emerging structures in an interactive manner.

In the following studies we employ a variant of Iwata, Houlsby and Ghahramani's *most-likely embedding* technique [17] to interact with the embedding via control points. The general idea behind this method is to customize a matrix that projects the data into the embedding space in a probabilistic way. This projection matrix is assumed to be matrix-normal distributed, a matrix-valued extension to the normal distribution. Ultimately, MLE calculates the embedding with the least uncertainty about the placement of the data records, given a prior belief on the projection matrix and conditioned on the control points' placements as evidence. In contrast to Iwata et al.'s method we do not use the Laplacian of the nearest-neighbour graph, but instead the projection onto the first two principal components as prior belief about the embedding (see Appendix A.1).

Finally, inspecting the structures that emerge when interacting with the embedded patterns can be done in various ways. In our exemplary study we use two simple, yet effective methods. The first is highlighting all the patterns within the embedding that contain an item of interest. Second, we also consider presenting the five most-frequently occurring items in a studied structure in a tag cloud. It is also possible to use more-sophisticated methods to study the pattern distribution. For example, we could perform pattern mining on the previously discovered patterns that form such a structure. Alternatively, we can also find a single well-suited representative pattern of the structure However, as our study shows, it is possible to gain insights and craft hypotheses using only our employed naïve methods.

4.1 Frequent Itemsets

In this section we show our proposed approach in action and demonstrate how the frequent patterns reflect rudimentary properties of the original dataset. Note that investigating the most frequent item sets with our proposed method serves mostly the purpose of a sanity check and demonstrating our approach in action. Figure 1 shows the 1000 most-frequent item sets of the cocktail dataset represented as binary vectors over all items, embedded onto their first two principal components. Immediately, we can see two well separated clusters that resemble roughly in their shape. Investigating these clusters closer reveals that the right one contains only patterns that include the ingredient *Vodka*, the most-frequent single item in the original dataset, whereas the left one doesn't (see Figure 1, left). The second most-frequent ingredient, *Orange juice*, determines whether a pattern is mapped to the top or to the bottom of the embedding (see Figure 1, right).

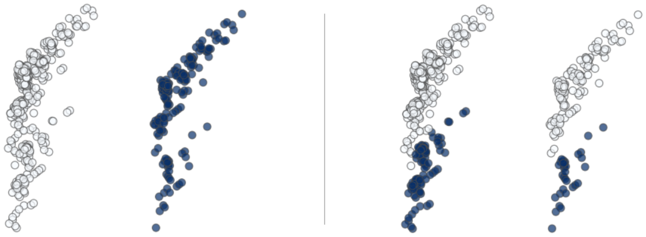


Figure 1: The 1000 most-frequent item sets of the cocktail dataset, embedded onto their first two principal components, labeled by the presence of *Vodka* (left) and *Orange juice* (right).

Interacting with the embedding by relocating two control points, as shown in Figure 2, unravels the blending of the patterns that contain *Orange juice* and the ones that don't. The resulting four clusters clearly separate the patterns by their presence or absence of the ingredients *Vodka* and *Orange juice*.

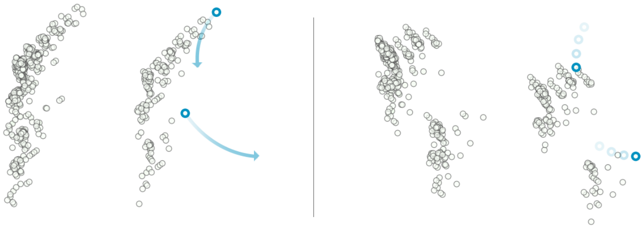


Figure 2: Dragging two control points (emphasized in blue) to new locations, reveals a structure that was previously hidden in the PCA embedding. The four clusters indicate the presence or absence of the two ingredients *Vodka* and *Orange juice*.

Figure 3 inspects one of these emerging structures, the top-right "*Vodka* and no *Orange juice* cluster" from Figure 2, in a closer manner.

With a glance at the top-left picture of Figure 3 we can see that the corresponding patterns containing *Vodka* but no *Orange juice* also frequently contain other strong alcohols, especially *Rum*, *Gin*, and *Triple sec*. We can also observe a sub-cluster structure within this particular embedding, which is determined by the presence or absence of the ingredients *Rum* (top-right, highlighted in green) and *Gin* (bottom-left, highlighted in blue). The ingredient *Triple sec* (bottom-right, highlighted in red), although frequent within this cluster, seems not to contribute to the sub-structure, but can be found in all of the sub-clusters. This is an interesting finding, as *Triple sec* is much more frequent than *Rum*. In fact, *Rum* does not even occur among the ten most-frequent ingredients, yet it has a striking influence on the structure of this cluster. Note that this is an insight that could not have been drawn purely from the results of Table 1. In the following sections we will perform similar studies with pattern collections that were drawn according to more-sophisticated interestingness measures than frequency of occurrence.

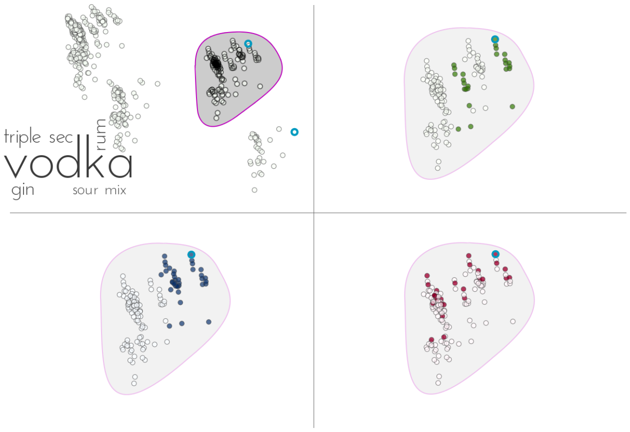


Figure 3: A closer look at the top-right cluster of Figure 2 reveals the ingredients that the patterns from the "*Vodka* and no *Orange juice* cluster" are frequently mixed with (top-left). The other three pictures indicate the presence of *Rum* (highlighted in green), *Gin* (blue), and *Triple sec* (red).

4.2 Sampled Patterns

A fruitful way to quickly draw patterns from a dataset according to different interestingness measures is to sample. Although sampling itself provides diversity among the drawn patterns, sorting them by the measure and listing only the top-k ones can reintroduce a certain amount of redundancy. On the other hand, diversity is not impaired when exploring the set of all sampled patterns in our proposed way and the analyst is further enabled to discover the different concepts among the patterns. In this study, we sampled 1000 patterns from the cocktail dataset, according to their rarity measure, a variant of the lift measure which promotes patterns containing items that are statistically dependent (see Appendix A.2). The samples were drawn using the *direct local pattern sampling tool* which was provided to us by Boley et al. [6] and can be downloaded from http://kdml-bonn.de/?page=software_details&id=23.

The retained collection of the sampled patterns demonstrates well how our proposed approach benefits from the use of interactive embedding techniques. The plain PCA embedding of the frequent patterns in the previous Section 4.1 already exhibited a clear structure, which directly invited the analyst to further explore it. For this particular set of sampled patterns, however, this is not the case. Figure 4 shows the sampled rare patterns embedded into two dimensions, using different techniques, namely PCA, Isomap, and locally linear embedding.[1]

[1] The latter two techniques estimated the assumed lower-dimensional manifold via the 10-nearest-neighbour graph.

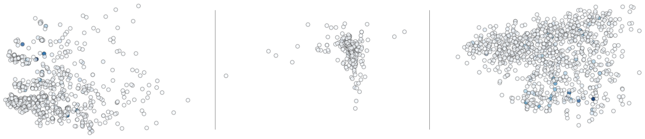


Figure 4: 1000 patterns sampled from the cocktail dataset, according to the *rarity* measure [6] and embedded, using different techniques: principal component analysis (left), locally linear embedding (middle), and isometric mapping (right).

Although these static embeddings exhibit no structures that immediately raise the analysts attention, relocating just one control point in the interactive embedding reveals clusters that were previously obscured, as Figure 5 (top) shows.

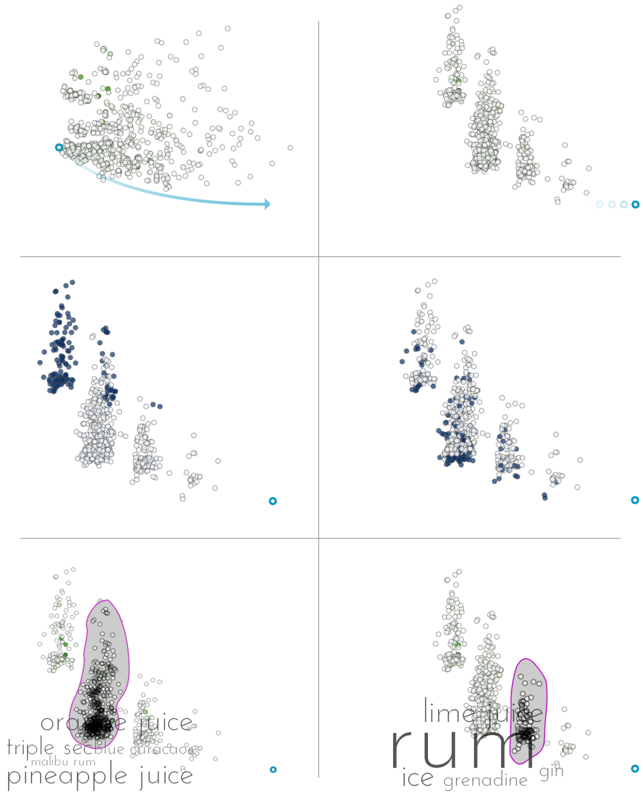


Figure 5: Relocating a control point, using our interactive embedding reveals a clear cluster structure (top). The middle pictures highlight the patterns containing *Vodka* (left) and *Orange juice* (right). The bottom pictures inspect the composition of two of these clusters.

The two middle pictures of the figure highlight the patterns containing *Vodka* (left) and *Orange juice* (right). Clearly we can identify the *Vodka* cluster, but the other clusters come as a surprise. They do not relate to the *Vodka* / *Orange juice* segmentation that was already discovered in Section 4.1, but capture concepts of their own. The two highlighted ones at the bottom of the figure revolve around juicy and *Rum*-heavy cocktails. Because of the initially mentioned redundancy among the highest rated rare patterns, the results from Table 1 mainly exhibit patterns associated with *Vodka*. Our proposed interactive discovery approach, however, was able to overcome this drawback and reveal other, novel concepts among the high-rarity patterns.

4.3 Subgroup Descriptions

Patterns can be discovered according to different measures of interest. In the previous sections we studied pattern sets that were drawn proportional to their measure of frequency or rarity. In some cases, however, the analyst might also want to consider label information. A classic pattern-mining approach that does so is *subgroup discovery*. It ranks the patterns by how much the label distribution of the data records described by the pattern diverges from the label distribution of the whole dataset. In this section we study the top-1000 closed subgroup descriptions from the cocktail dataset, ranked according to the binomial test quality measure [4] (see Appendix A.3). Figure 9 shows the embedding of these 1000 patterns onto their first two principal components.

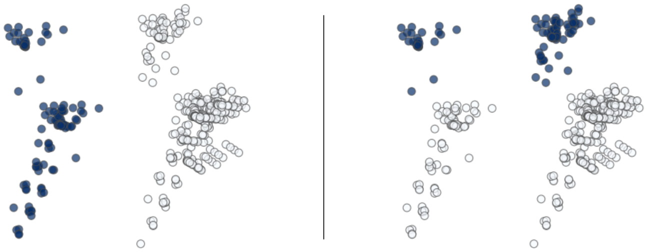


Figure 6: The top-1000 subgroup descriptions associated to the label *creamy*, embedded onto their first two principal components. The four clusters coincide with the presence/absence of the two most striking ingredients among creamy cocktails: *Baileys* (left) and *Kahlúa* (right).

Similar to the embedding of the frequent item sets, but without the help of any interaction, the mined patterns fall directly into four clusters. This time, the clustering goes along with the presence or absence of two other frequently occurring ingredients: *Baileys* (left) and *Kahlúa* (right). From the list of frequent patterns in Table 1 we know that these ingredients are highly frequent, and from the list of subgroups we know that they have a stark impact on the label of a cocktail. In this sense, the observed segmentation doesn't come as a total surprise. However, following the results of Table 1 we might instead have expected *Crème de cacao*, instead of *Kahlúa*. The visualization helps to understand the relations among the listed patterns and invites for further exploration of the exhibited structure. To do so, this time we do not interact with the embedding via the earlier utilized control points, but rather by focusing on a subset of the distribution. We filter the pattern collection to keep only the ones that contain neither *Kahlúa* nor *Baileys* and re-embed them onto their first two principal components. The selection corresponds to the patterns belonging to the bottom right cluster of Figure 6. The re-embedding of these selected patterns can be seen in Figure 7 below.

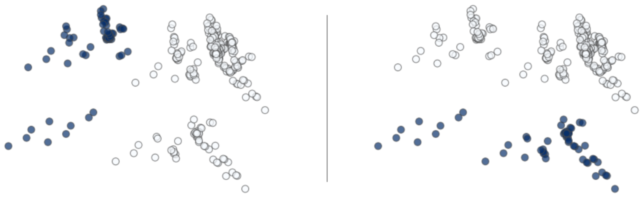


Figure 7: A PCA embedding of the patterns belonging to the bottom right cluster of Figure 6. Again, the embedded patterns can be neatly segmented by the presence of two highly frequent ingredients, this time *Vodka* (left) and *Crème de cacao* (right).

As the re-embedding is not a zoom, but a newly calculated PCA embedding, we are able to discover structures that were previously hidden due to the covariance among the patterns that are now filtered out. Once again we observe that the patterns form four clusters, corresponding to highly frequent ingredients, this time *Vodka* and *Crème de cacao*. Note that this 'four cluster segmentation' is not part of our proposed method, but stems form the sparsity which transactional databases often expose. To achieve a clearer separation of the clusters in the visualization, we use again the placement of a control point, as shown in the following Figure 8.

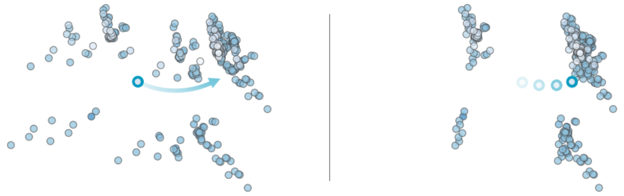


Figure 8: To retrieve a better separation between the clusters, we interact with the embedding by selecting and relocating an appropriate control point.

As an example, we pick two of the clusters from Figure 8 and study their compositions. Figure 9 below shows the five most-frequent ingredients within the patterns of these clusters in a tag cloud.

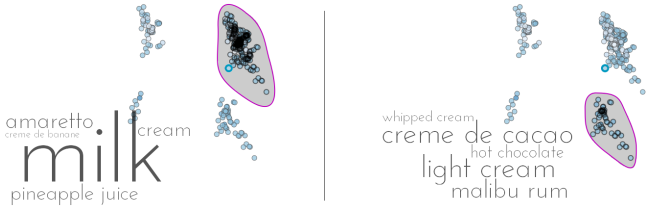


Figure 9: Inspecting the contents of two of the emerging clusters. One interesting finding is the occurring separation between milky and chocolaty patterns. The cluster segmentation stems from the presence of the ingredients *Vodka* and *Crème de cacao*.

We can observe that the inspected regions contain patterns that stem from two different types of creamy cocktails: milky and chocolaty ones. This is an interesting finding, as the strict separation between the clusters does not stem from the milky ingredients within the patterns, but from the ingredients *Vodka* and *Crème de cacao*. However, using our interactive visualization, we were able to craft the hypothesis that milky and chocolaty cocktails form different types of creamy cocktails, offering a good next direction to explore.

4.4 Discussion

Using an interactive embedding of the patterns to visualize and explore it, we were able to remedy the information overload that comes naturally with the consideration of a large pattern collection. Our proposed approach mainly collapses into a two step procedure: (1) mine a large collection of patterns and (2) explore a visualized embedding of the patterns in an interactive way. We demonstrated our approach on pattern collections that resulted from three different mining techniques, namely frequent pattern mining, sampling patterns proportional to their lift, and subgroup discovery. In the second step, we followed the information-seeking mantra and explored the obtained pattern collections in a top-down manner. We started with a visual overview of the whole pattern distribution and dug deeper on striking structures by interacting with the visualization and investigating the emerging structures in different ways, namely by

- reshaping the embedding via relocating control points.
- filtering out and re-embedding the remaining patterns.
- listing the most-frequent items of an inspected structure.
- highlighting all patterns containing an ingredient of interest.

By interactively exploring the pattern collection, we were able to gain some minor insights that we could not draw by purely considering the results of Table 1. To give some examples, from the list of frequent patterns we know that *Vodka* and *Orange juice* are the most-frequent ingredients of the cocktail dataset, but the PCA embedding was able to reveal how much more *Vodka* distinguishes between the cocktails than *Orange juice* does. By inspecting the sub-clusters that emerged from our interaction, we found a surprisingly strong influence of the ingredient *Rum* on the cocktails containing *Vodka* but not those containing *Orange juice*. This discovery is backed up by the high-lift pattern *Vodka & Rum* that we can find in Table 1. However, considering the mirroring of the "no-*Orange juice*-clusters", located at the top in Figure 2, we can also craft a theory about a strong influence of *Rum* among the non-*Vodka* patterns in general. We were also able to discover three strong concepts among the patterns with a high lift: the pattern *Vodka & Something*, fruity cocktails, and *Rum*-heavy cocktails. This is especially interesting, as *Rum* does not rank among the ten most-frequent ingredients. In addition, we were also able to discover independently from Table 1 that *Kahlúa*, *Baileys*, *Crème de cacao* and *Milk* are mainly responsible for a cocktail being labeled as *creamy*.

However, the strength of our approach lies not in these discoveries, but in the deeper understanding of the relations among the patterns that it provides in combination with the classical pattern-mining methods. By exploring the pattern embedding, interacting with it, exposing interesting structures, and always collating the crafted theories and insights with Table 1, we were able to develop an understanding of the different concepts that the original cocktail data revolves around.

5. CONCLUSION

We proposed an extension to the classical pattern-mining approach that enables the analyst to overcome information overload when browsing and exploring a larger collection of patterns. The goal of our proposed method is to help the analyst understand the underlying distribution of the patterns and additionally to invite them to further exploration. Whereas the classical pattern mining approach focuses on presenting a condensed set of high-quality patterns, our approach uses interactive embedding techniques to visualize and explore the distribution of a larger pattern collection. To do so, we proposed a general four-step approach, where each step can be instantiated in different ways. In our exemplary study, we demonstrated the use of our approach by exploring and interacting with three different pattern collections from a cocktail-ingredient dataset. Collating our findings and the results of different pattern-mining algorithms, we were able to forge and test hypotheses and develop an understanding of the mined patterns and the different concepts that they descend from.

6. REFERENCES

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*. 1996.

[2] A. Asuncion and D. J. Newman. Uci machine learning repository, http://archive.ics.uci.edu/ml, 2007.

[3] M. Berardi, A. Appice, C. Loglisci, and P. Leo. Supporting visual exploration of discovered association rules through multi-dimensional scaling. In *Proceedings of Foundations of Intelligent Systems*. Springer, 2006.

[4] M. Boley and H. Grosskreutz. Non-redundant subgroup discovery using a closure system. In *Proceedings of Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD*, 2009.

[5] M. Boley, C. Lucchese, D. Paurat, and T. Gärtner. Direct local pattern sampling by efficient two-step random procedures. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD*. ACM, 2011.

[6] M. Boley, S. Moens, and T. Gärtner. Linear space direct pattern sampling using coupling from the past. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD*. ACM, 2012.

[7] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *Proceedings of Visual Analytics Science and Technology, VAST*. IEEE, 2012.

[8] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman and Hall/CRC, 2000.

[9] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. Observation-level interaction with statistical models for visual analytics. In *Proceedings of Visual Analytics Science and Technology, VAST*. IEEE, 2011.

[10] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *Transactions on Computers*, 1974.

[11] G. C. Garriga, P. Kralj, and N. Lavrač. Closed sets for labeled data. *Journal of Machine Learning Research*, 9, 2008.

[12] F. Geerts, B. Goethals, and T. Mielikäinen. Tiling databases. In *Proceedings of Discovery science*. Springer, 2004.

[13] H. Grosskreutz and D. Paurat. Fast and memory–efficient discovery of the top–k relevant subgroups in a reduced candidate space. In *Proceedings of Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD*, 2011.

[14] H. Grosskreutz, D. Paurat, and S. Rüping. An enhanced relevance criterion for more concise supervised pattern discovery. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD*, 2012.

[15] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of Special Interest Group on Management of Data, SIGMOD*, 2000.

[16] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., 2001.

[17] T. Iwata, N. Houlsby, and Z. Ghahramani. Active learning for interactive visualization. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics, AISTATS*, 2013.

[18] W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *Proceedings of Advances in Knowledge Discovery and Data Mining*. AAAI, 1996.

[19] N. Lavrač and D. Gamberger. Relevancy in constraint-based subgroup discovery. *Constraint-Based Mining and Inductive Databases*, 2005.

[20] N. Lavrač, B. Kavšek, P. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5(Feb), 2004.

[21] D. Leman, A. Feelders, and A. Knobbe. Exceptional model mining. In *Proceedings of Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD*. Springer, 2008.

[22] S. C. Leman, L. House, D. Maiti, A. Endert, and C. North. Visual to parametric interaction (v2pi). *PloS one*, 8(3), 2013.

[23] F. Lemmerich and M. Atzmueller. Incorporating exceptions: Efficient mining of epsilon-relevant subgroup patterns. In *Proceedings of the ECML PKDD Workshop LeGo: From Local Patterns to Global Models*, 2009.

[24] F. Lemmerich and M. Atzmueller. Fast discovery of relevant subgroup patterns. In *Proceedings of Florida Artificial Intelligence Research Society, FLAIRS*, 2010.

[25] M. Mampaey, N. Tatti, and J. Vreeken. Tell me what i need to know: succinctly summarizing data with itemsets. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD*, 2011.

[26] D. Oglic, D. Paurat, and T. Gärtner. Interactive knowledge-based kernel pca. In *Proceedings of Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD*, 2014.

[27] D. Paurat and T. Gärtner. Invis: A tool for interactive visual data analysis. In *Proceedings of Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD*, 2013.

[28] D. Paurat, D. Oglic, and T. Gärtner. Supervised PCA for interactive data analysis. In *Proceedings of the NIPS 2nd Workshop on Spectral Learning*, 2013.

[29] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500), 2000.

[30] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE, 1996.

[31] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2000.

[32] T. Uno, T. Asai, Y. Uchida, and H. Arimura. An efficient algorithm for enumerating closed patterns in transaction databases. In *Proceedings of Discovery Science, DS*, 2004.

[33] S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of Principles of Data Mining and Knowledge Discovery, PKDD*. Springer, 1997.

APPENDIX

A.1 The most likely embedding

Our approach for interacting with a lower-dimensional embedding of data makes use of the matrix-normal distribution, an extension of the multivariate normal distribution to matrix-valued arguments. The idea is to find the linear projection from the original data space into the embedding space that is most likely, given a prior belief about the embedding and conditioned on the placement of selected control points. The $(p \times q)$-dimensional matrix normal distribution $\mathcal{MN}_{p,q}(R; M, \Sigma, \Psi)$ has the density function

$$\mathcal{MN}_{p,q}(R; M, \Sigma, \Psi) = (2\pi)^{-\frac{pq}{2}} |\Sigma|^{-\frac{q}{2}} |\Psi|^{-\frac{p}{2}} \\ \exp\Big(-\tfrac{1}{2} \operatorname{tr}\big[\Sigma^{-1}(R - M)\Psi^{-1}(R - M)^{\top}\big]\Big),$$

where:

- $R \in \mathbb{R}^{p \times q}$ is the matrix-valued argument,
- $M \in \mathbb{R}^{p \times q}$ is the location parameter, and
- $\Sigma \in \mathbb{R}^{p \times p}$, $\Psi \in \mathbb{R}^{q \times q}$ are symmetric positive-definite scale parameters that can be considered the "row" and "column" covariance matrices, respectively.

Why is this useful? Suppose we have a matrix normal distributed belief about a linear embedding matrix R:

$$p(R \mid \theta) = \mathcal{MN}(R; M, \Sigma, \Psi),$$

where θ represents the hyperparameters M, Σ, and Ψ. Now further suppose that we have observed data $X \in \mathbb{R}^{D \times N}$ in a potentially high-dimensional Euclidean space $\mathcal{X} = \mathbb{R}^{D}$ and that the user has selected a total of m control points $Y \in X$ and has placed them in preferred locations $W \in \mathbb{R}^{2 \times m}$ within the two dimensional embedding. We will write $\mathcal{D}$ to indicate these observed data pairs (Y, W).

We also assume that the locations chosen for these points, given by the user, represent the correct latent locations for these points, corrupted by iid zero-mean isotropic Gaussian noise. Consider RY, which represents the embedded locations of Y given knowledge of the latent embedding matrix R. Our assumption is that the control points placed by the users are close to their ideal locations:

$$p(W \mid RY, \theta, \sigma^2) = \mathcal{MN}(W; RY, I, \sigma^2 I),$$

which indicates that each of the values in W differs from RY by entrywise iid Gaussian noise with variance σ^2. Henceforth we will include σ^2 in the set of hyperparameters θ.

Now we can reason about the linear projection matrix R that is most likely, given a prior believe about the embedding and conditioned on the observed values W:

$$p(R \mid Y, W, \theta) = \mathcal{MN}(R; M_{R|\mathcal{D}}, \Sigma, \Psi_{R|\mathcal{D}}),$$

where

$$M_{R|\mathcal{D}} = M + (W - MY)(Y^{\top}\Psi Y + \sigma^2 I)^{-1} Y^{\top}\Psi;$$
$$\Psi_{R|\mathcal{D}} = \Psi - \Psi Y(Y^{\top}\Psi Y + \sigma^2 I)^{-1} Y^{\top}\Psi.$$

In order to retrieve the final most likely embedding of all the data points X, we simply have to calculate the $M_{R|\mathcal{D}}X$.

To utilize this method in a live-update manner, reasonably many updates have to be calculated per second. If the interaction with the embedding is only the movement of control points, then solely $M_{R|\mathcal{D}}$ has to be recalculated and multiplied by X to retrieve the embedding. The following Figure 10 depicts the updates per second for this case, depending on the number of attributes, data records and used control points. However, if the selection of the control points changes, also $\Psi_{R|\mathcal{D}}$ has to be recalculated (which on a regular PC runs in well under a second). As depicted, the update-rate depends the strongest on the number of data records and drops with an increasing amount of them. Using our non-tweaked implementation, a dataset of about 1500 data-records could be interacted with at an update-rate of roughly 10-15 updates per second. The dataset used in this experiment was an excerpt from the *Communities and Crime* dataset, taken from the UCI dataset repository [2].

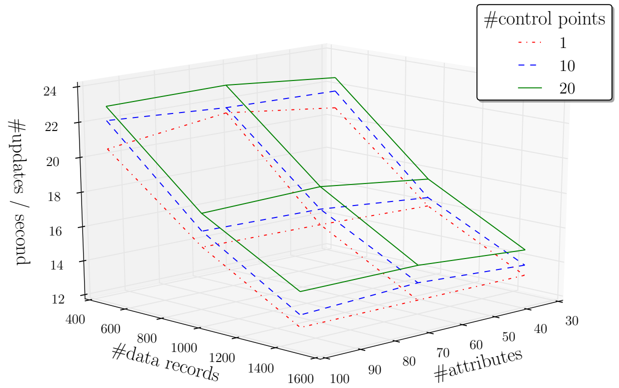


Figure 10: Achieved updates per second for 1,10 and 20 selected control points, depending on the number of data-records and attributes of the dataset.

A.2 The rarity measure

The rarity of a pattern approximates the probability of occurrence of the whole pattern weighted by the probabilities of the single items that build the pattern not occurring. To put it in a formal way, let $\mathcal{D}$ be a transactional database over a fix set of items. Further, let P be a pattern, consisting of k of these items $P = \{p_1, \ldots, p_k\}$. The rarity of P is calculated as

$$\operatorname{rarity}(P, \mathcal{D}) = \operatorname{freq}(P, \mathcal{D}) \prod_{p_i \in P} \big(1 - freq(p_i, \mathcal{D})\big),$$

where $\operatorname{freq}(x, \mathcal{D})$ denotes the observed frequency of occurrence of the pattern x in the database $\mathcal{D}$.

There is a relation to the lift measure of a pattern, which is calculated by

$$\operatorname{lift}(P, \mathcal{D}) = \operatorname{freq}(P, \mathcal{D}) \prod_{p_i \in P} \frac{1}{\operatorname{freq}(p_i, \mathcal{D})}.$$

Whereas *rarity* considers the absence-frequency of the singleton items, *lift* considers the inverse of them.

A.3 Subgroup quality measures

In the context of subgroup discovery, the interestingness of a pattern is measured by a quality function $q(P, \mathcal{D})$ that considers the pattern and the dataset and returns a real-valued number. This function usually combines the size of the support set of the pattern and its unusualness w.r.t. the designated target label in the following way:

$$q(P, \mathcal{D}) = \operatorname{freq}(P, \mathcal{D})^{\alpha} \cdot \big(\operatorname{share}^{+}(P, \mathcal{D}) - \operatorname{share}^{+}(\emptyset, \mathcal{D})\big),$$

where $\text{share}^+(P, \mathcal{D})$ denotes the share of the positively labeled data records in the support set of the pattern P and $share^+(\emptyset, \mathcal{D})$ the share of all positively labeled data records over $\mathcal{D}$. It is defined by

$$\text{share}^+(P, \mathcal{D}) = \frac{\text{freq}(P, \mathcal{D}^+) \cdot |\mathcal{D}^+|}{|\mathcal{D}|},$$

with $\mathcal{D}^+$ denoting the set of positively labeled data records from $\mathcal{D}$. The coefficient α of the quality function is a constant $0 \leq \alpha \leq 1$, characterizing a family that includes some of the most-popular quality functions. For $\alpha = 1$ it is order-equivalent to the weighted relative accuracy (WRACC) and the Piatetsky–Shapiro quality function. For $\alpha = 0.5$ it corresponds to the binomial test quality function, which is used to mine the subgroup description patterns in Section 4.3.

Skim-reading thousands of documents in one minute: Data indexing and visualization for multifarious search

Alessandro Perina
Microsoft Research and
Istituto Italiano di Tecnologia
Redmond WA / Genova Italy

Dongwoo Kim
KAIST
Daejeon, Korea

Andrzej Turski
Microsoft Corporation
Redmond, WA

Nebojsa Jojic*
Microsoft Research
Redmond, WA
**Corresponding author:*
jojic@microsoft.com

ABSTRACT

In this paper we present an interface based on a recent generative model, the counting grid, here re-introduced in its basic version and largely revised to allow it to deal with large corpora. We show that it is possible to visualize thousands of high order word co-occurrence patterns by only viewing for a few minutes a new embedding we propose for text visualization, browsing and search purposes. We performed preliminary experiments with user tasks such as word spotting, rapid content search and collateral information acquisition.

1. INTRODUCTION

Embedding text documents into a 2D space (e.g. [13, 3]) has always been an appealing idea: If we can turn a discrete complex dataset into something that looks like an image, perhaps our brains' low to medium-level processing layers will take the lead and help us consume the dataset in a flash, the way our eyes process almost any natural image. The old idea that various types of knowledge may already be captured in image-like mental representations in our mind [8] further strengthens our expectation that even the knowledge that is inherently as discrete, hierarchical and propositional as that encoded using language, can be transformed into something continuous and referentially isomorphic, a data-driven smooth mapping that our eyes can easily saccade over. Another vehicle for of obtaining a "birds-eye-view" is the notion of the word/tag cloud where a smaller or larger handful of characteristic words is shown to the user as a summary and a very rudimentary index of the data.

However, multiple dangers lurk here. Our eyes saccade over text differently than over natural images [10, 7, 2]. The speed of visual word recognition is highly dependent on the words' immediate context, which can both speed it up *and* slow it down [2].
This of course has consequences to visualization and user interface design. For example, a 2D embedding of titles in a distance-based document embedding is hard to make sense of as the processing required for us to understand the discovered links is at a too high a level to gel well with the visual traversing paradigm. High level category labels are often added to aid the user in making sense of different areas in the embedding, but as indicated above, these labels are likely to make it even more difficult to understand the outliers that happen around the boundaries. This is why some visualizations only show documents as dots of different colors indicating broad categories, but essentially hiding all of their content until the user mouses over. Data that way does become more image-like but is akin to a very simple image.

On the other hand showing a large number of constitutive words from a document is problematic due to the users' reading habits. For instance, alphabetically arranged tags can easily be misinterpreted by a user who tends to look for a meaning in groups of words, and so the sequence of tags "living man missing money news" from a word cloud from one day of CNN news may all refer to different news stories, yet it is difficult for a human reader not to jump to a conclusion that either money or the man is missing.

It has been shown, e.g. in [12], that semantic organization of words significantly affects the user' interaction with the data, making lower-level connections (folksonomy based) better suited for consumption than the higher level language models. Thus it is not surprising that most previous user studies of various text visualization techniques similar to these resulted in the conclusion that when the user is interested in a very specific bit of information, the regular search engine interface will suffice, and that in most other situations the beneficial effects of the visualization are hard to quantify, other than through user satisfaction levels. Users tend to favor these tools, perhaps because, as we stated above, the idea of being able to extract the essence of the data and lay it out onto the screen in a rich, yet easy to grasp manner is just as appealing to the users as it is to the researchers, even if it is hard to realize.

In this paper we present an interface built upon the recent Counting Grid model [6] and we strongly believe that the approach may be a step forward. We also propose few learning algorithm aimed at avoiding local minima and producing more grids for usable for users.

1.1 The counting grid: A way forward?

We imagine a large grid of cells, each with a few words of different weights so chosen so that words collected from any single document in the dataset can be represented well by the weighted words in one small window encompassing several cells in the grid Fig. 1a. Aided with a good optimization technique and a user interface that fits the model well, several very interesting properties of such an embedding arise.

Firstly, it is possible to make the mapping very dense, avoiding the excessive levels of empty space in typical distance-preserving embedding methods (note that our visual system distorts distances, see, for example [5]).

Secondly, in such dense mappings the grid is too small to avoid overlaps of windows, and so then the extent of the similarity of the nearby documents in terms of simple word usage statistics can readily be seen directly in the grid: The words shared between the two documents will tend to be seen in the region of the overlap of the two windows. Thirdly, if we travel slowly across the grid and look at the documents mapped there, we should often see gradual thematic shifts as the words early in our path are dropped and new ones are added, but the overlap in content between our new area of focus and the one just before tends to stay high. Obviously for diverse enough datasets, occasionally the smoothness in theme shifts will have to be violated in areas where two different topics expanding from different points clash in a single area creating a rift between two less related groups of documents. Finally, in most places we look, the words we can get from the nearby cells will tend to be highly related, and this should make it easier to perform visual word recognition tasks if all these words are shown on the screen, such as word spotting in a search for a particular word it should be often easy to pick out document groupings, focus on one of the relevant ones, and then follow the trail to the point of interest, then jump to another grouping of interest and focus on the new area, etc.

We call this model the counting grid, as it is a grid of word counts, and in the next section we state this idea mathematically. Then, we describe the techniques needed to properly optimize and present the counting grid to the user as an interface to various medium-sized datasets (cooking recipes, research papers, movie descriptions, etc.). Finally, we demonstrate that our interface does indeed expose high order statistics (word co-occurrence statistics beyond pairs) which then become a powerful visualization tool for both understanding the extent of the dataset and discovery of items of interest. We show that both the word combinations are meaningful beyond what was previously attempted, increasing the word spotting speeds, and that they lead to good indexing of a diverse dataset enabling users to perform dozens of semi-related search tasks in parallel in mere minutes and then walk away with much more collateral information that seeped into their brains serendipitously.

Algorithm 1: EM-Algorithm to learn the Counting Grids.

```
Input: Bag of words, $c_z^t$ for each sample
while Convergence do
    % E-Step ;
    foreach Sample $t = 1 \dots T$ do
        1. Update $q_{\mathbf{k}}^t \propto \exp \sum_z c_z^t \cdot \log h_{\mathbf{k},z}$ ;
    % M-Step ;
    2. Update $\pi_{\mathbf{k},z} \propto \pi_{\mathbf{k},z}^{old} \cdot \sum_t c_z^t \sum_{\mathbf{i}|\mathbf{k}\in W_{\mathbf{i}}} \frac{q_{\mathbf{i}}^t}{h_{\mathbf{i},z}}$ ;
    3. Compute $h_{\mathbf{k},z} = \frac{1}{W_1 \times W_2} \sum_{\mathbf{i}\in W_{\mathbf{k}}} \pi_{\mathbf{i},z}$;
    4. Compute the Log-Likelihood (Eq. 1) ;
    5. Check for convergence ;
6. Return $\pi_{\mathbf{k},z}$ and $\{q_{\mathbf{k}}^t\}$ ;
```

2. THE COUNTING GRID MODEL

The counting grid consists of a set of discrete locations in a map of arbitrary dimensions (32 × 32 or 64 × 64 in the examples used in this paper). Each location contains a different set of weights for the each of the words in the vocabulary. A document has its own word usage counts c_z and the assumption of the counting grid model is that this word usage pattern is well represented at some location i in the grid. The window floating over the grid captures well variation in certain types of documents where we can see slow evolution of the topics, where certain words are dropped and new ones introduced.

A particular example of a counting grid and its weights are illustrated in Fig. 1 using font size variation, but showing only the top 3 words at each location. The shaded cells are characterized by the presence, with a non-zero probability, of the word "bake"[1]. On the grid we also show the windows $\mathbf{W}$ for 5 recipes. *Nomi* (1), an Afghan egg-based bread, is close to the recipe of the usual *pugliese bread* (2), as indeed they share most of the ingredients and procedure. Note how moving from (1) to (2) the word "egg" is dropped. Moving to the right we encounter the *basic pizza* (3) whose dough is very similar to the bread's. Continuing to the right words often associated to desserts like sugar, almond, etc emerge. It is not surprising that baked desserts such as *cookies* (4), and pastry in general, are mapped here. Finally further up we encounter other desserts which do not require baking, like *tiramisu* (5), or *chocolate crepes*.

Formally, the basic counting grid $\pi_{\mathbf{i},z}$ is a set of normalized counts of words / features indexed by z on the 2-dimensional discrete grid indexed by $\mathbf{i} = (i_1, i_2)$ where each $i_d \in [1 \dots E_d]$ and $\mathbf{E} = [E_1, E_2]$ describes the extent of the counting grid. Since π is a grid of distributions, $\sum_z \pi_{\mathbf{i},z} = 1$ everywhere on the grid. A given bag of words/features, represented by counts $\{c_z\}$ is assumed to follow a count distribution found somewhere in the counting grid. In particular, using windows of dimensions $\mathbf{W} = [W_1, W_2]$, each bag can be generated by first averaging all counts in the window $W_{\mathbf{k}} = [\mathbf{k}, \dots, \mathbf{k} + \mathbf{W}]$ starting at grid location $\mathbf{k}$ and extending in each direction by W_d grid positions to form the histogram $h_{\mathbf{k},z} = \frac{1}{W_1 \times W_2} \sum_{\mathbf{i}\in W_{\mathbf{k}}} \pi_{\mathbf{i},z}$, and then generating a set of features in the bag. In other words, the position of the window $\mathbf{k}$ in the grid is a latent variable given which the probability of the bag of features $\{c_z\}$ is

$$p(\{c_z\}|\mathbf{k}) = \prod_z (h_{\mathbf{k},z})^{c_z} = \frac{1}{W_1 \times W_2} \prod_z \Big(\sum_{\mathbf{i}\in W_{\mathbf{k}}} \pi_{\mathbf{i},z}\Big)^{c_z},$$

Fine variation achievable by moving the windows in between any two close by but non-overlapping windows is useful if we expect such smooth thematic shifts to occur in the data, and we illustrate in our experiments that indeed it does.

To learn a Counting Grid we need to maximize the likelihood of the data:

$$\log P = \sum_t \log \Big(\sum_{\mathbf{k}} \cdot \prod_z (h_{\mathbf{k},z}^{c_z^t}) \Big) \qquad (1)$$

The sum over the latent variables $\mathbf{k}$ makes it difficult to perform assignment to the latent variables while also estimating the model parameters. The problem is solved by employing a variational EM procedure, which iteratively learn the model, alternating E and M-step. The E step aligns all bags of features to grid windows, to match the bags' histograms, inferring , i.e., were each bag maps on the grid. In the M-step we re-estimate the counting grid so that these same histogram matches are better. The procedure is illustrated with algorithm 1; $\pi_{\mathbf{k},z}^{old}$ is the counting grid at the previous iteration.

Even for large corpora the learning algorithm converges in 70-80 iterations, which sums up to minutes for summarizing corpora of

[1] Which may or may not be in the top three

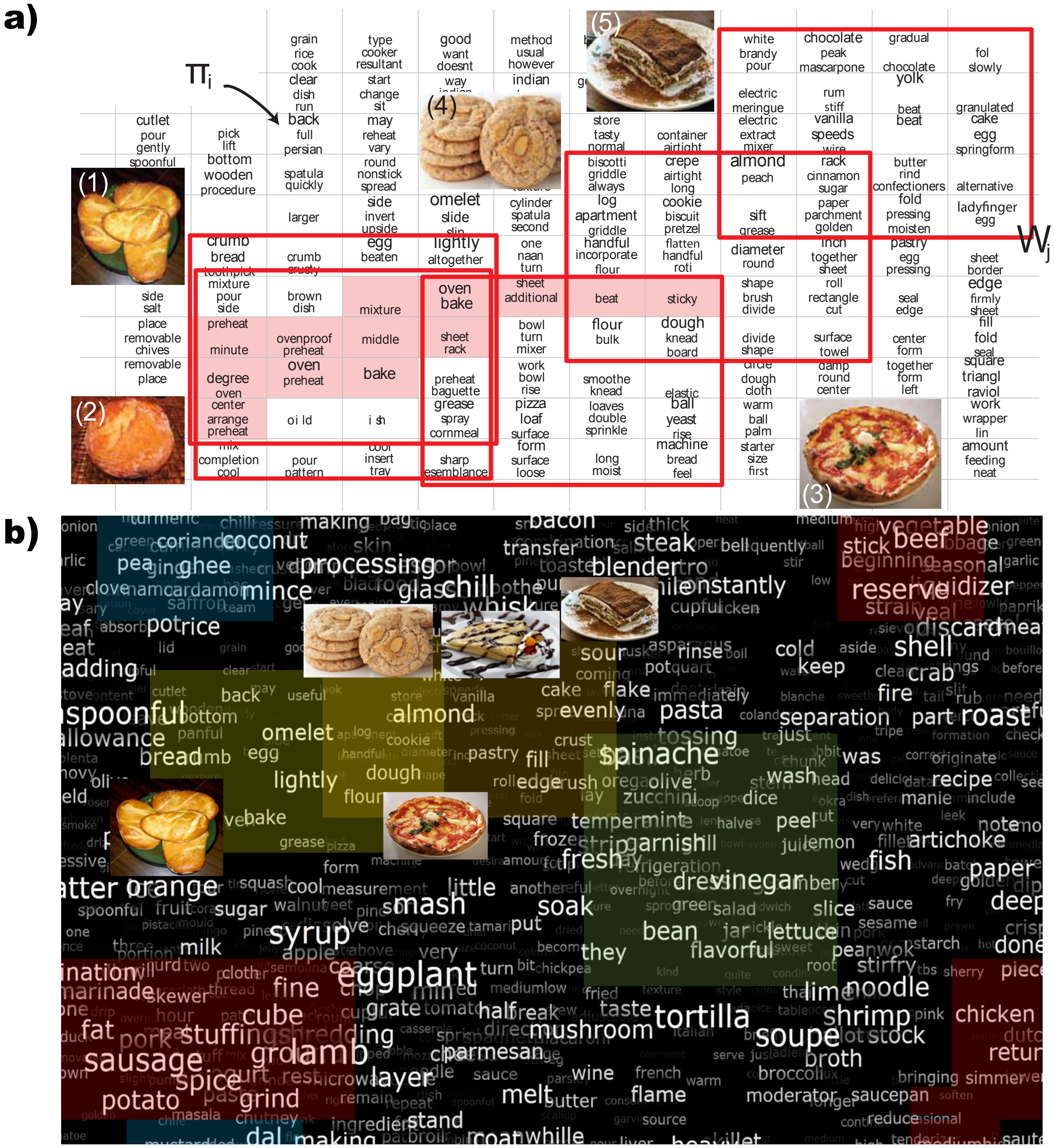


Figure 1: a) A particular of an area of a counting grid π_i learned over a corpus of recipes. In each cell we show the 0-3 most probable words greater than a threshold. The area in shaded red has $\pi('bake') > 0$. b) The interface built upon the (whole) counting grid shown in panel a) (here shaded in yellow). We also highlighted areas relative to spices (blue), vegetables (green), meats (red).

over 40K documents.

As this EM algorithm is prone to local minima, the final grid will depend on the random initialization, and the neighborhood relationships for mapped documents may change from one run of the EM to the next. However, in our experience, the results always appear very similar, and most of the more salient similarity relationships are captured by all the runs.

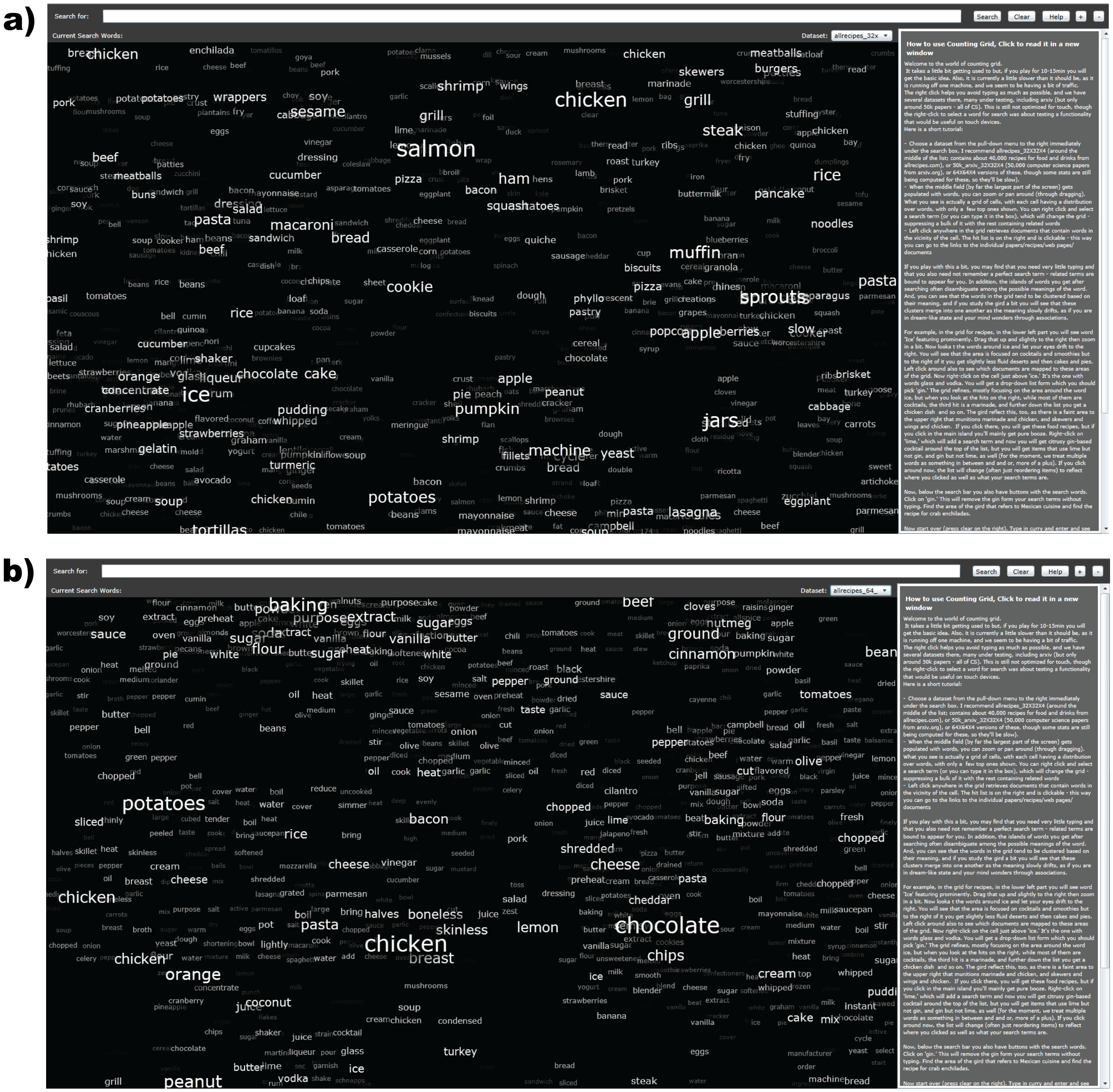
More importantly, a majority of the neighborhood relationships make sense from a human perspective and thus the mapping gels the documents together into logical, slowly evolving (in space) themes. As discussed below, this helps guide our visual attention to the subject of interest.

3. COUNTING GRID AS A USER INTERFACE

In order to use the counting grid as an underlying representation in a powerful UI, we found three improvements necessary.

3.1 Optimization algorithms

Given the considerations in the introduction, the quality of em-



How to use Counting Grid, Click to read it in a new window

Welcome to the world of counting grid.
It takes a little bit getting used to but, if you play for 10-15min you will get the basic idea. Also, it is currently a little slower than it should be, as it is running off one machine, and we seem to be having a bit of traffic.
The right click helps you avoid typing as much as possible, and we have several datasets there, many under testing, including arxiv (but only around 50k papers - all of CS). This is still not optimized for touch, though the right-click to select a word for search was about testing a functionality that would be useful on touch devices.
Here is a short tutorial:

- Choose a dataset from the pull-down menu to the right immediately under the search box. I recommend allrecipes_32X32X4 (around the middle of the list; contains about 40,000 recipes for food and drinks from allrecipes.com), or 50k_arxiv_32X32X4 (50,000 computer science papers from arxiv.org), or 64X64X4 versions of these, though some stats are still being computed for these, so they'll be slow).
- When the middle field (by far the largest part of the screen) gets populated with words, you can zoom or pan around (through dragging). What you see is actually a grid of cells, with each cell having a distribution over words, with only a few top ones shown. You can right click and select a search term (or you can type it in the box), which will change the grid - suppressing a bulk of it with the rest containing related words
- Left click anywhere in the grid retrieves documents that contain words in the vicinity of the cell. The hit list is on the right and is clickable - this way you can go to the links to the individual papers/recipes/web pages/documents

If you play with this a bit, you may find that you need very little typing and that you also need not remember a perfect search term - related terms are bound to appear for you. In addition, the islands of words you get after searching often disambiguate among the possible meanings of the word. And, you can see that the words in the grid tend to be clustered based on their meaning, and if you study the gird a bit you will see that these clusters merge into one another as the meaning slowly drifts, as if you are in dream-like state and your mind wonders through associations.

For example, in the grid for recipes, in the lower left part you will see word 'Ice' featuring prominently. Drag that up and slightly to the right then zoom in a bit. Now looka t the words around ice and let your eyes drift to the right. You will see that the area is focused on cocktails and smoothies but to the right of it you get slightly less fluid deserts and then cakes and pies. Left click around also to see which documents are mapped to these areas of the grid. Now right-click on the cell just above 'ice.' It's the one with words glass and vodka. You will get a drop-down list form which you should pick 'gin.' The grid refines, mostly focusing on the area around the word ice, but when you look at the hits on the right, while most of them are cocktails, the third hit is a marinade, and further down the list you get a chicken dish and so on. The gird reflect this, too, as there is a faint area to the upper right that munitions marinade and chicken, and skewers and wings and chicken. If you click there, you will get these food recipes, but if you click in the main island you'll mainly get pure booze. Right-click on 'lime,' which will add a search term and now you will get citrusy gin-based cocktail around the top of the list, but you will get items that use lime but not gin, and gin but not lime, as well (for the moment, we treat multiple words as something in between and and or, more of a plus). If you click around now, the list will change (often just reordering items) to reflect where you clicked as well as what your search terms are.

Now, below the search bar you also have buttons with the search words. Click on 'gin.' This will remove the gin form your search terms without typing. Find the area of the gird that refers to Mexican cuisine and find the recipe for crab enchiladas.

Now start over (press clear on the right). Type in curry and enter and see

How to use Counting Grid, Click to read it in a new window

Welcome to the world of counting grid.
It takes a little bit getting used to but, if you play for 10-15min you will get the basic idea. Also, it is currently a little slower than it should be, as it is running off one machine, and we seem to be having a bit of traffic.
The right click helps you avoid typing as much as possible, and we have several datasets there, many under testing, including arxiv (but only around 50k papers - all of CS). This is still not optimized for touch, though the right-click to select a word for search was about testing a functionality that would be useful on touch devices.
Here is a short tutorial:

- Choose a dataset from the pull-down menu to the right immediately under the search box. I recommend allrecipes_32X32X4 (around the middle of the list; contains about 40,000 recipes for food and drinks from allrecipes.com), or 50k_arxiv_32X32X4 (50,000 computer science papers from arxiv.org), or 64X64X4 versions of these, though some stats are still being computed for these, so they'll be slow).
- When the middle field (by far the largest part of the screen) gets populated with words, you can zoom or pan around (through dragging). What you see is actually a grid of cells, with each cell having a distribution over words, with only a few top ones shown. You can right click and select a search term (or you can type it in the box), which will change the grid - suppressing a bulk of it with the rest containing related words
- Left click anywhere in the grid retrieves documents that contain words in the vicinity of the cell. The hit list is on the right and is clickable - this way you can go to the links to the individual papers/recipes/web pages/documents

If you play with this a bit, you may find that you need very little typing and that you also need not remember a perfect search term - related terms are bound to appear for you. In addition, the islands of words you get after searching often disambiguate among the possible meanings of the word. And, you can see that the words in the grid tend to be clustered based on their meaning, and if you study the gird a bit you will see that these clusters merge into one another as the meaning slowly drifts, as if you are in dream-like state and your mind wonders through associations.

For example, in the grid for recipes, in the lower left part you will see word 'Ice' featuring prominently. Drag that up and slightly to the right then zoom in a bit. Now looka t the words around ice and let your eyes drift to the right. You will see that the area is focused on cocktails and smoothies but to the right of it you get slightly less fluid deserts and then cakes and pies. Left click around also to see which documents are mapped to these areas of the grid. Now right-click on the cell just above 'ice.' It's the one with words glass and vodka. You will get a drop-down list form which you should pick 'gin.' The grid refines, mostly focusing on the area around the word ice, but when you look at the hits on the right, while most of them are cocktails, the third hit is a marinade, and further down the list you get a chicken dish and so on. The gird reflect this, too, as there is a faint area to the upper right that munitions marinade and chicken, and skewers and wings and chicken. If you click there, you will get these food recipes, but if you click in the main island you'll mainly get pure booze. Right-click on 'lime,' which will add a search term and now you will get citrusy gin-based cocktail around the top of the list, but you will get items that use lime but not gin, and gin but not lime, as well (for the moment, we treat multiple words as something in between and and or, more of a plus). If you click around now, the list will change (often just reordering items) to reflect where you clicked as well as what your search terms are.

Now, below the search bar you also have buttons with the search words. Click on 'gin.' This will remove the gin form your search terms without typing. Find the area of the gird that refers to Mexican cuisine and find the recipe for crab enchiladas.

Now start over (press clear on the right). Type in curry and enter and see

Figure 2: Interface: a) Counting Grids b) Distance Embedding + Keywords.

bedding can have a dramatic effect on the user experience. The CG model is more directly tied to the goal of visualizing higher order statistics in word usage patterns than previous models: It literally attempts to lay the words out so that nearby words can be found commonly in the documents (and even in the intersection of highly related documents). Thus the direct optimization of data likelihood should get us good embeddings. However, there are no globally optimal likelihood optimization methods for this model. Fortunately, the basic Em model derived in [6] does at least provably converge to a local minimum. Furthermore, for the purposes of the browser we tested here, we experimented with various ways of escaping local minima, such as sampling methods, random restarts, online learning/gradient descent, and found that the nicest grids with highest likelihood tend to be created by a multiresolution approaches.

In a first approach an 8×8 grid is first estimated using the 5×5 mapping window size. The grid is then upsampled by replacing each cell with a 2×2 set of cells with the same distribution. Then the EM learning of this 16×16 grid is continued using the same size of the mapping windows (5×5) until convergence. This process is then iterated to the desired size of the grid.

In a second approach we kept fixed the grid size, progressively reducing the window size every 10 iterations until we reached the desired window size.

We found that these multi-resolution approaches create longer thematic shifts and fewer boundaries among areas, which is generally more pleasing to the eye and makes it easier for the user to learn "the lay of the land." We believe that further improvements in optimization algorithms may create dramatically better results, esp. for large datasets.

3.2 Pan-zoom-click-search interface to a CG

The interface, shown in Fig. 2-4, allows several modes of interaction with the data and the grid. The grid itself is rendered so that the font size denotes the local weights of different words directly imported from the model. The weights essentially indicate how likely the words are in context of other nearby words. We have implemented a fast pan-zoom interface for exploration of the grid in Silverlight (Fig.3 shows the zoom). A click (or a tap on touch devices) shows the set of documents whose mapping windows overlap the point we clicked on. The list is shown on the right without changing the grid view. The grid can be filtered in two ways: by typing the search term in the search box, or by simply selecting a word (right click on long tap). Two search results are illustrated by Fig.4: in panel "a" memory, in panel "b", forest (see the text box on the top of the interface).

Assuming that a very specific search goal with a well formulated query cannot be aided much by dataset summarization and diversity exposure, we did not test the counting-grid representations primarily on such tasks. Instead, we have made our interface as close to traditional search-based interfaces as possible for such situations: The user can enter the search terms and the results will be presented in the list on the right hand side of the interface. However, through grid filtering described above, our interface also provides a diversity viewing experience that aims to expose the user to themes related to a specific successful query, as well as a summary/grouping of relevant content for less specific queries and summary, organization and visualization of the entire dataset for multi-objective or free-form browsing experience. Importantly, the counting grid representation combined with the pan-zoom-click-search interface enables a unified way of data consumption across these levels of granularity of user interest. For example, a high-quality query that results in high relevance of returned items will filter the very same grid representing the entire dataset, with the effects shown in-place, so that gradual removal of search query words will expand the scope till the entire dataset is shown. As the relative positioning of topics/themes stays fixed through this experience, moving back and forth among different search goals with possibly varying levels of specificity does not throw the user out of context, which in traditional interfaces poses a barrier for multifarious search and makes the user focus and organize their tasks linearly, rather than in parallel. Perhaps most importantly for the diverse application of the ideas presented here, the user interface is created automatically from the dataset as the input, using an unsupervised machine learning algorithm, and the result can in principle be refined by professional curator/designer or collaboratively by users, who can add their content or labels anywhere in the grid.

4. EVALUATION

We evaluated our interface in several ways. First, we were curious to see how much the direct optimization, in maximum likelihood sense, of embedding word sharing patterns aids the visualization of higher order co-occurrence statistics, and if these improvements indeed yield to increased speeds of resulting word cloud skimming. As these results indicated a clear advantage of counting grids over the alternatives, we next investigated the amount of gleaned information during a short exposure to the data through our interface and compared this directly with the state-of-the art, but traditional web site interface to the same data, as such comparisons in the past tended to not show a quantifiable advantage of word clouds over simple search interfaces, while at the same time the user surveys usually showed that users like word clouds and are under the impression that the clouds may aid them in goal-free exploration of the content.
In all the experiments we employed the multiresolution approach of Sec. 3.1 to learn the grids, removing stop-words and applying the Porter-stemmer algorithm [9].

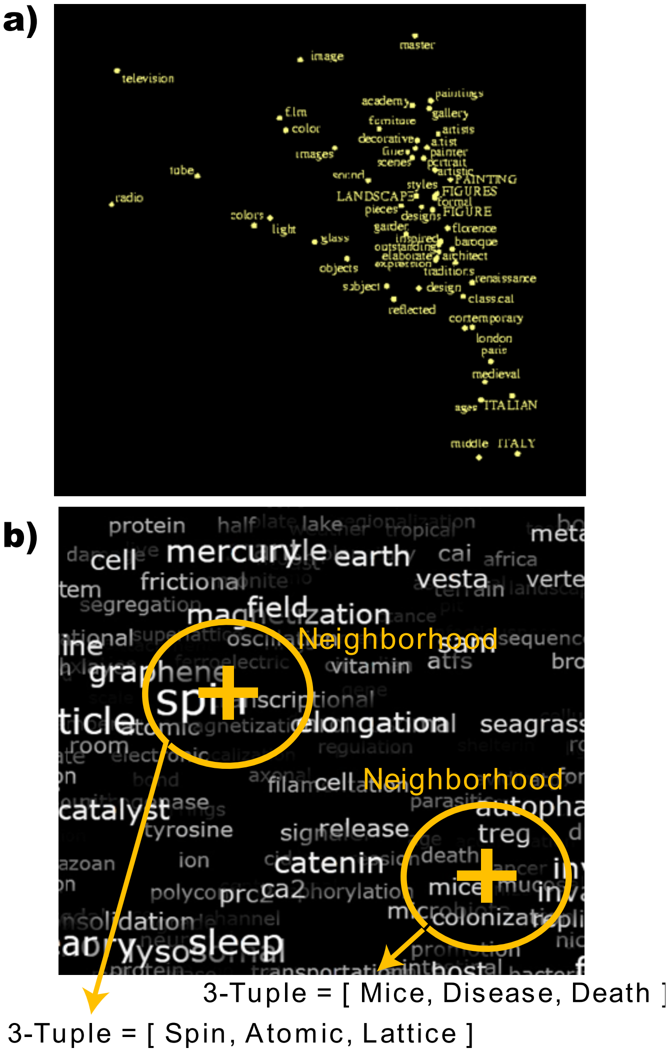

a)
b)
Neighborhood
Neighborhood
3-Tuple = [Mice, Disease, Death]
3-Tuple = [Spin, Atomic, Lattice]

Figure 5: a) A word embedding produced by an euclidean embedding method. b) The process of tuple sampling: A position is randomly picked on the grid and words are sampled from a neighborhood.

4.1 Word combinations at random focus areas: Numerical comparisons

One of the immediate goals of CG optimization is to create a

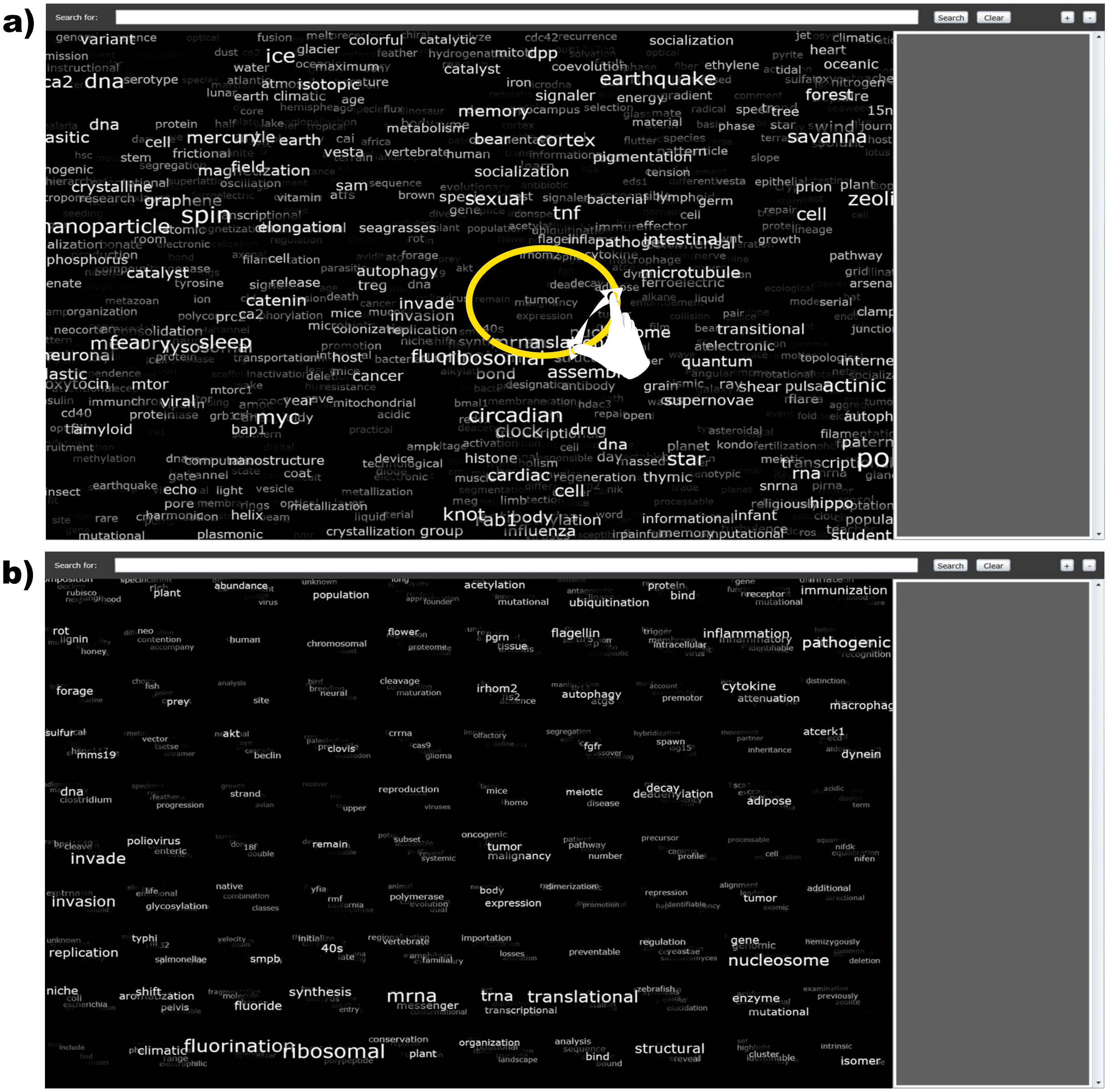


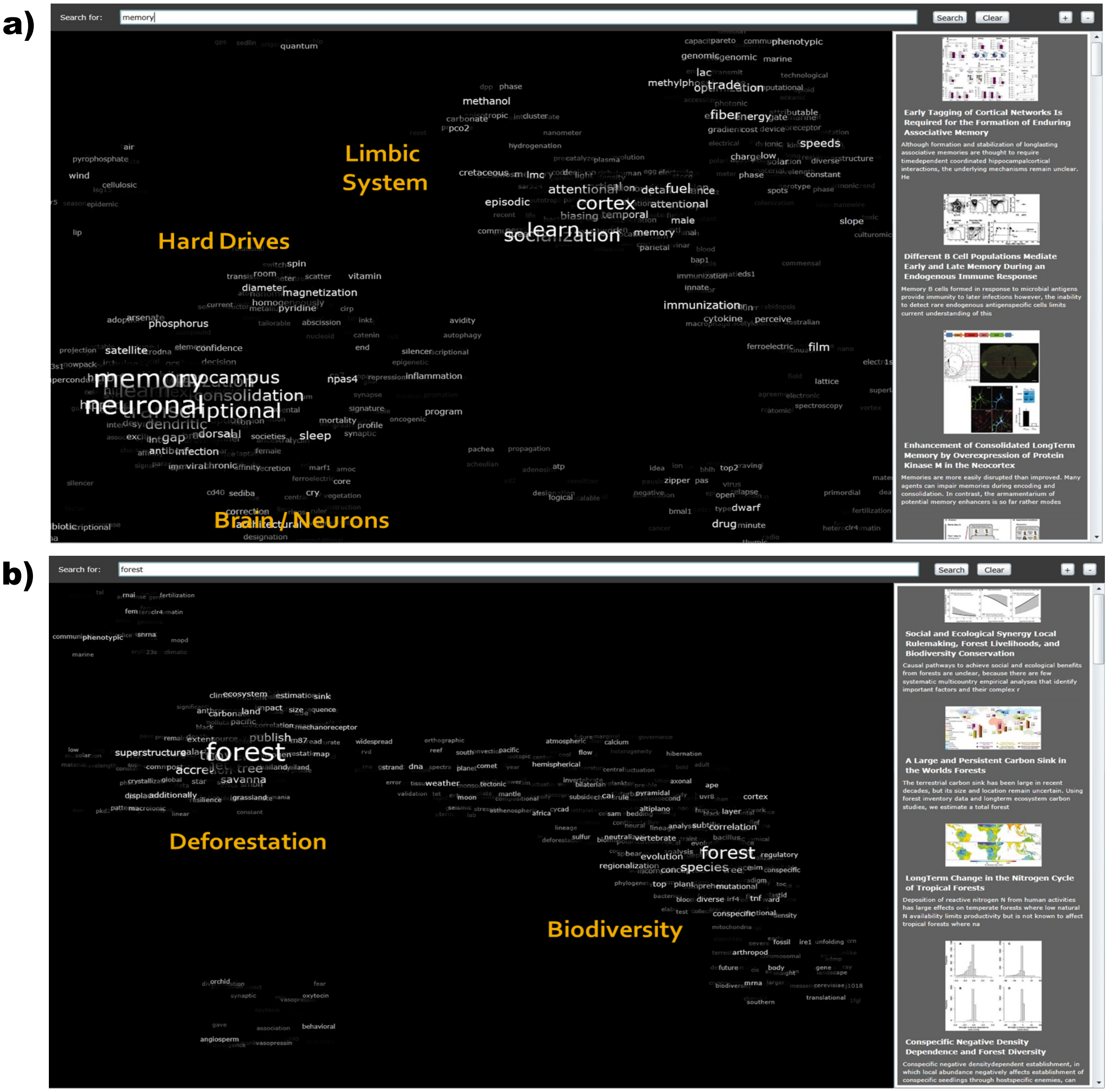
Figure 3: Zoom: a) A counting grid learned using Science magazine papers and reports. The user can zoom until visualizing the top words of each source (panel b)

visualization in which high order statistics of many word combinations can easily be visualized: In any local area of the grid, the words seen in the neighboring cells should "go together" so as to make the consumption of the grid easier. This aspect of the counting grids can be quantified directly without user studies, through hundreds of grid sampling steps.

In each step, a "neighborhood" in the window picked uniformly at random [2], and then k words are drawn from that window according to the local word distribution. This sampling process is illustrated by Fig.5b.
Then these k-tuples are checked for *consistency* and *diversity* of indexed content. The consistency is quantified in terms of the average number of documents from the dataset that contained all k words selected, while the diversity of indexed content is illustrated through the cumulative graph of acquired unique documents as more and more k-tuples are sampled and used to retrieve documents containing them.

We would expect that the CG model should show good consistency of words selected this way as the model is in fact optimized

[2] the curves look very similar for 3×3 to 7×7 window choices even though the grids were learned using assuming that each document maps to a 5×5 window



Early Tagging of Cortical Networks Is Required for the Formation of Enduring Associative Memory

Although formation and stabilization of longlasting associative memories are thought to require timedependent coordinated hippocampalcortical interactions, the underlying mechanisms remain unclear. He

Different B Cell Populations Mediate Early and Late Memory During an Endogenous Immune Response

Memory B cells formed in response to microbial antigens provide immunity to later infections however, the inability to detect rare endogenous antigenspecific cells limits current understanding of this

Enhancement of Consolidated LongTerm Memory by Overexpression of Protein Kinase M in the Neocortex

Memories are more easily disrupted than improved. Many agents can impair memories during encoding and consolidation. In contrast, the armamentarium of potential memory enhancers is so far rather modes

Social and Ecological Synergy Local Rulemaking, Forest Livelihoods, and Biodiversity Conservation

Causal pathways to achieve social and ecological benefits from forests are unclear, because there are few systematic multicountry empirical analyses that identify important factors and their complex r

A Large and Persistent Carbon Sink in the Worlds Forests

The terrestrial carbon sink has been large in recent decades, but its size and location remain uncertain. Using forest inventory data and longterm ecosystem carbon studies, we estimate a total forest

LongTerm Change in the Nitrogen Cycle of Tropical Forests

Deposition of reactive nitrogen N from human activities has large effects on temperate forests where low natural N availability limits productivity but is not known to affect tropical forests where na

Conspecific Negative Density Dependence and Forest Diversity

Conspecific negative densitydependent establishment, in which local abundance negatively affects establishment of conspecific seedlings through hostspecific enemies, can

Figure 4: Search results are presented as (non contiguous) islands on the grid, where different islands capture different semantics of keywords. For example, a) search result of the word "memory" reveals three islands related to computer memory, brain memory and the limbic system. Analogously b) search for the word "forest" revealed an island about deforestation and one about biodiversity in forests. By interacting with these islands the user can filter out unwanted results, or discover new things.

so that documents' words map into overlapping windows, and so through the positioning and intersection of many related documents the words should end up being arranged in a fine-grained manner so as to reflect their higher-order co-occurrence statistics.
To the best of our knowledge there is currently no other technique that attempts to perform similar optimization, so we compare here with an approach based on previous techniques that achieved visually most similar arrangements, at least at a first glance (see Fig.2).

Some previous embedding techniques proposed word embedding based on pairwise distances, or joint embedding of words and the documents based on document-document and document-word distances [11, 4]. The problem with these approaches is that each word is assigned to a single location but certain highly informative words still assume multiple meanings in different contexts. For example, the word "memory" in the corpus of Science magazine papers can be found in articles on neuroscience, but also in immunology (immune memory of the adaptive immune system), device memory, as well as in quantum mechanics and occasional computer science papers. This would make such a word a nexus

Corpus	# Docs	# Words	Tokens	Notes
Science Magazine	36K	24K	2.0M	Papers and Reports
Allrecipes	43K	4K	10M	
Arxiv	25K	31K	2.3M	Computer Science
IMDB	18K	25K	0.9M	Popular movies

Table 1: Statistics of the four corpora considered

of several different clusters, making the browsing confusing in that area. Things are worse given that there are in fact many such words, and the attempts of embedding into 2D in this way usually collapse. Another promising approach is to simply focus on document embedding and then show representative words from nearby documents in the plane [3]. The problem here is that most embedding methods create a lot of empty space among clusters, which leads to dramatic under use of screen real estate (see Fig.5a).

Nevertheless, we embark on this approach to build a reasonable baseline for our method by further deforming the neighborhood relationships are maintained but the grid is denser (otherwise, this method would suffer on diversity measures described above). This baseline is further aided by making an effort to avoid local word repetitions which further reduce the information content of the grid and thus the diversity measure above. Fig. 2a shows the best so obtained embedding for allrecipes.com data, containing 43k recipes. Although at a first glance the two visualizations share a lot of common qualities, the sampling experiments show a dramatic difference in favor of CG on four different datasets, all approximately 50K in size: Science Magazine articles from the last 10 years, all of arxiv CS articles, allrecipes.com, and the most popular movies from IMDB. Details of each dataset are reported in table 1.

As shown in Fig.6 the more traditional distance embedding + keyword spraying approach matches, more or less, the quality of CG when we sample for word pairs (k=2). However, as this approach, or any other in the literature does not attempt to directly capture higher level statistics of word usage, even though the general clusters look meaningful at a first glance that capture grow structure of the data, the fine grained local structure of CGs much better captures higher order correlation, with this advantage typically growing with k. One outlier seems to be the most diverse Science dataset with the richest vocabulary. The curves in Fig. 6 are pretty close, but Fig. 7a which shows the diversity of the indexed information explains the difference. Fig. 7b, shows the gradient of the last curve of Fig. 7a.

An embedding of words that creates the same trivial combinations of words in many areas of the grid (e.g. {salt, paper, sprinkle} would boost the fraction of dataset covered by this triple. However, the number of new documents would then not grow. In case of counting grid, not only are the k-tuples meaningful, but they are diverse and with repeated jumping over the grid more and more content is being retrieved, which is the combination we want in a user interface meant for summarizing, browsing and retrieval.

4.2 High speed multifarious search and the extent of collateral information gleaned

As we discussed in the introduction, the main motivation in research on visualizing datasets by mapping documents and/or displaying word clouds is in the potential ease in understanding the extent of the dataset, locating topics of interest quickly when these interests are not well defined, as well as accidental discovery of interesting and useful information [1] that is somewhat related to the original goals of the information seeking process.

Here we test the ability of users to rapidly gain insight both into specific and broad topics which are either directly or indirectly related to a mix of topics of interest, as well the collateral information gleaned in the process.

The traditional search paradigm would force us to try to look for this research linearly, focusing on one are at a time, getting new ideas for search only once we read the discovered papers. The counting grid visualizations with orders of magnitude more words than usual tag clouds and at a same time much denser and better organized embedding of relevant documents may (and did) enable us do some of these investigation rapidly and in parallel, jumping from topic to topic as the links are revealed. We assume, of course, that such multifarious search, where a variety of topics are of interest, some at a high level, and others needing to be explored in depth is often attempted by Internet users in a variety of tasks, and we focus here again on the allrecipes dataset.

We created a questionnaire with 60 questions of various specificity about the contents of the dataset by repeatedly sampling recipes form the dataset and formulating questions at different level of description depth, like “Are there Indian dishes here?”, “Are there crepe recipes in this dataset?”, “Are there savory recipes?”, “Do any recipes use zucchini?” etc. Then we added several control questions for which we knew that they referred to items not covered by the dataset, like “Wine reviews” or “Cheese platters” or “Cooking book reviews”. We expect that the users’ performance on this task should be predictive of the experience they would have with our tool in many real world scenarios.

We compared the CG interface with the allrecipes web’s own professionally and community-curated easy-to-use and powerful interface, which includes a modern search engine, various categorizations of the data, user-supplied votes and labels, etc.

We recruited seven subjects for the study and told them that they would be asked to answer a series of questions, including a list of ten that we read to them, and that they had 3 minutes to find out as many answers as they can using one combination of a dataset and an interface at a time.

We told the users that they would do this for two such combinations and we convinced them that the two combinations differ both in the extent of the data and the user interface, and that other than the initial 10 questions, the questionnaires would also be randomly related. However, to avoid issues with comparisons of different questionnaires and datasets on a small sample of users, we in fact varied only the interface, and used the same dataset asked the same questions, but placed the competing interface second in the study, where it would presumably have an advantage over our method if the users would never the less be inclined to look for answers to the questions beyond the initial ten questions. We hoped that the limited amount of time provided for the task would minimize that advantage anyhow, as our preliminary tests on the authors and pre-test subjects indicated that more than five minutes were need to perform all the searching necessary to cover a large fraction of questions if the user is searching based on their memory of the entire questionnaire. In addition, we found that no subject was able to find information relevant to more than about half of the questions asked, indicating again that there was not enough time for the traditional interface to gain significant unfair advantage over our interface. No single question was answered correctly by all users, except for the control questions, for which the real answer was no (There were no wine reviews in the data, etc.), indicating that they

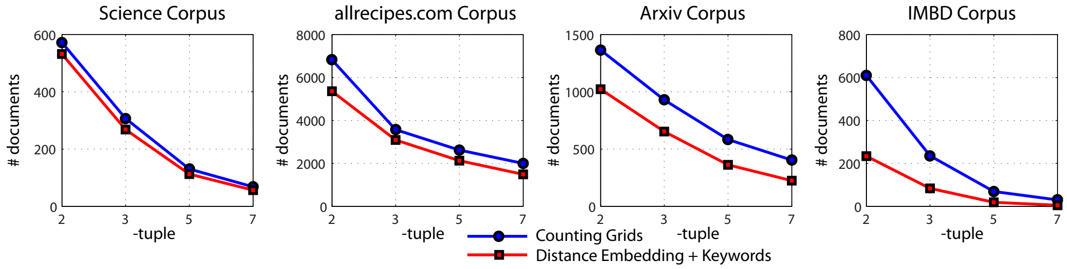


Figure 6: Consistency results

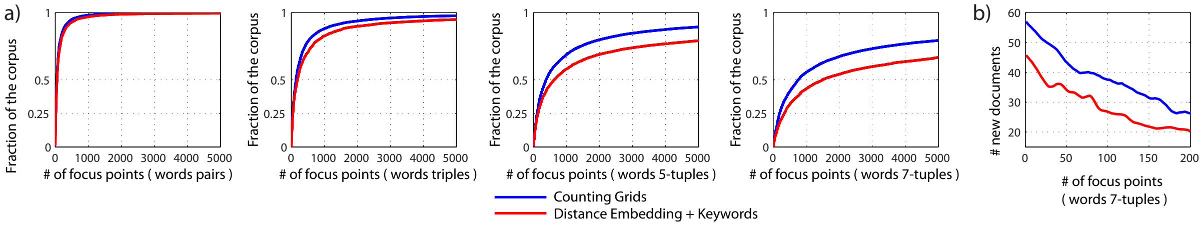


Figure 7: Diversity results

were giving us honest representations of what they remembered. All seven users performed better using counting grids (p<0.01), with the average gain in the number of questions answered of 60% over the allrecipes.com interface, despite the potential advantages that the latter may have had due to order of testing. The ability to glean collateral information beyond the 10 questions to which the users were primed was certainly biased by users' food preferences or familiarity with cooking styles. Only our one Chinese test subject detected traces of the Chinese cuisine in the counting grid based on the combination of ingredients much more typical of the Chinese cuisine; a quick click there indeed revealed Chinese dishes he had in mind. The types of meat and vegetables the users found or did not find in the dataset typically correlated with their preference for these foods.
However, for all users in this small study, the intersection of their preferences with the questions asked was enough to provide enough answers in order to see the difference between the two interfaces. Interestingly, the percentage of answered questions varied more widely after using the allrecipes.com standard interface (as low as 22% and as high as 46%) than for CG interface (42% - 51%), which provides another indication of the interaction between the users' own memory and the CG. Using the standard search interface the users could not remember or think to explore further items of low interest to them, even after seeing recipes that could provoke further investigation. But the word associations in the CG interface seemed to more readily enter their visual field and remind them of the task defined by the initial questions. Results are summarized in Fig. 8.

In post-test interviews, all users indicated preference for the CG interface for the task of rapidly discovering lots of information as well as for organizing the data. They could simply "see much more in parallel" in the CG interface, and could often recognize recipes just based on the words in the grid and without opening any of the documents mapped in the area. They also indicated that they had a better understanding what data was exposed by the CG interface, while the boundaries of allrecipes.com interface seemed uncharted and the data thus appeared potentially vast (even though the number of recipes was approximately the same). When asked for a subjective estimate of how much information they encountered while using the CG compared to the standard interface, they reported factors of 2-5, which are either inflated subjective estimates (compared to the measured factors which varied between 1.2 and 1.9), or they indicate that the users saw much more of the content related to their food interests in addition to the content to which we primed them to look for. The latter possibility would be in line with previously observed difficulties in measuring the diversity of information the user accesses during data exploration.

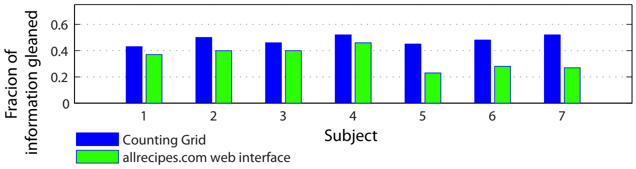


Figure 8: Information gleaning experiment

5. CONCLUSION

To the best of our knowledge, the counting grid visualization we presented here is the first system that directly optimizes for simultaneous presentation of word co-occurrence statistics of various orders well beyond the usual pairwise embedding. This is accomplished through a dense word and document embedding that facilitates a visual browsing and search paradigm that can more naturally rely on the cognitive processes we employ when we scan visual scenes as well as the ones that guide visual recognition of words in skim reading. We have shown that the CG representation tends to display words that go together in almost any location in the visual field, and that by sampling different local combinations of words we tend to identify a larger fraction of the dataset and in a more diverse manner across locations than we can achieve using standard embedding methods to display large number of words from embedded documents. We also find that this increased semantic order does indeed facilitate faster visual processing of the word map, as well as faster memorization of the word distributions in word spotting experiments. In addition to data organization, the CG visualization also facilitates interesting patterns of partial document consumption. By spotting several related words, the user is reminded of the knowledge they already have, and may not even need to open relevant documents. As described in the grocery shopping case study, in such cases the effect is akin to parallel skim-reading of hundreds of documents that contain the word combination to narrow down on a known common theme and extrapolate (remember) the rest of the document to the extent needed by the user. In addition, surprising combination of words in the area the user is interested in can lead to serendipitous discovery of new documents to be studied in detail.

From the perspective of word/tag cloud usability research perhaps the most exciting result comes from our preliminary experiments on multifarious search and serendipitous data exposure that show that thousands rather than dozens of words on the screen can still be consumed by the user and that the extent of the data explored this way is high enough that the differences can be quantified in user studies.

However, despite encouraging preliminary results, a lot about counting grid representations and interface design remains to be studied. We found that the quality of the embedding of high order statistics matters, yet we know from our experiments that the current algorithms are prone to local minima. Thus it remains to be seen if the document packing can be done more optimally in the maximum likelihood sense and if such improved grids would provide even better local word combinations that would be even easier to browse/search. We have experimented with a wide variety of medium-sized datasets containing tens of thousands of documents. It remains to be seen what the best way would be to scale this experience to very large datasets. Interface refinements can play a big role, too. For example, in our three-tiered approach to visual searching over the grid – visual scanning, filtering by word seen in the grid, or filtering by a typed word not yet spotted – the last modality tended to be avoided by users to unreasonable levels because it was perceived to be at odds with the smoother experience of combining visual scanning with mouse/touch actions.

6. REFERENCES

[1] P. André, M. C. Schraefel, J. Teevan, and S. T. Dumais. Discovery is never by chance: Designing for (un)serendipity. In *In C and C Š09*. ACM, 2009.

[2] C. A. Becker. Semantic context effects in visual word recognition: An analysis of semantic strategies. *Memory and Cognition*, 8(6):493–512, 1980.

[3] B. Fortuna, M. Grobelnik, and D. Mladenić. Visualization of text document corpus. *Special Issue: Hot Topics in European Agent Research I Guest Editors: Andrea Omicini*, 29:497–502, 2005.

[4] A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean embedding of co-occurrence data. In *Advances in Neural Information Processing Systems 17*, pages 497–504. MIT Press, 2005.

[5] H. Intraub. The representation of visual scenes. *Trends in Cognitive Sciences*, 1(6):217–222, Sept. 1997.

[6] N. Jojic and A. Perina. Multidimensional counting grids: Inferring word order from disordered bags of words. In *UAI*, pages 547–556, 2011.

[7] M. A. Just and P. A. Carpenter. A theory of reading: from eye fixations to comprehension. *Psychological review*, 87(4):329, 1980.

[8] A. Paivio. *Mental Representations: A Dual Coding Approach (Oxford Psychology Series, 9)*. Oxford University Press, 1990.

[9] M. F. Porter. Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[10] K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, pages 372–422, 1998.

[11] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000.

[12] J. Schrammel, M. Leitner, and M. Tscheligi. Semantically structured tag clouds: an empirical evaluation of clustered presentation approaches. In *Proceedings of CHI '09*, 2009.

[13] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proceedings on Information Visualization*, pages 51–58, 1995.

Visualizing uncertainty in spatio-temporal data

Ayush Shrestha
Department of Computer Science
Georgia State University
Atlanta, GA
ashrestha2@cs.gsu.edu

Ying Zhu
Department of Computer Science
Georgia State University
Atlanta, GA
yzhu@cs.gsu.edu

Ben Miller
Department of Communication
Georgia State University
Atlanta, GA
miller@gsu.edu

ABSTRACT

Analyzing the relationship between location and time in a spatio-temporal data is not trivial. It is even more challenging if the data contains uncertainty. In this paper, we present a new method that visualizes spatio-temporal data with uncertainty. This method is an extension of our 2D visualization technique called Storygraph, and it handles two types of data uncertainty: (1) the spatial and temporal uncertainty about an event; (2) the spatial and temporal uncertainty between two events. We applied this method to a case study that involves data extracted from witness testimonies and field reports containing uncertainties inherent to natural language.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Evaluation/methodology*

1. INTRODUCTION

The introduction of geo-location sensors in mobile devices and other commodity hardware has greatly aided in spatio-temporal data collection. As a result, novel and effective methods are needed to help analyze these great amounts of spatio-temporal data. Traditional methods like maps fail to show the temporal sequence of the *events*. An event in this paper refers to a row in the dataset having distinct time and location. If two events occur at the same location at different times, the markers will overlap, resulting in a single marker. Time series charts are helpful for presenting temporal information but difficult for analyzing spatial information. Other methods such as small multiples, animations, and 3D maps have significant drawbacks.

In our previous work, we introduced a technique called Storygraph [1] to address these issues. Storygraph is a 2D technique that visualizes both spatial and temporal components in an integrated graph. Our case studies demonstrated the benefits of this method on datasets containing precise geolocations and time such as military war logs [1] and software commit histories [2]. However, when applying our method to spatio-temporal data extracted from witness testimonies and field reports, we encountered problems of uncertainty in space and time. For example, our study of 511 interviews with first responders during the attack on World Trade Center (WTC) on September 11, 2001 showed that the narratives of these interviewees, who were trained to report incidences, still contained a fair amount of uncertainty in their descriptions of locations and times.

To address these issues, we developed a new version of Storygraph visualize uncertainty. In our revision, we begin by categorizing uncertainty into two categories: (1) event uncertainty and (2) between-event uncertainty. We designed our method to distinguish and visualize these two types of uncertainty. Event uncertainty is the spatio-temporal uncertainty about the event itself, including events with poorly specified spatial and/or temporal attributes. Between-event uncertainty is the uncertainty between two precisely recorded events, which we call them *key events*. This concept is in part influenced by Hagerstrand's Time Geography [3][4][5]. After specifying the key events, the between-event uncertainties are visualized as space-time prisms between the key events. Through this process, our visualization technique can be used to study the interactions between people (or characters) in both space and time.

The rest of the paper is organized as follows: Section 2 discusses related work in spatio-temporal and uncertainty visualization. Section 3 presents the mathematical model of Storygraph. Section 4, describes the classification of uncertainty. Section 5 discusses how uncertain events are visualized in Storygraph. Section 6 discusses how between-event uncertainty is visualized. Section 7 presents a case study featuring fire fighter interviews from WTC corpus. Section 8 concludes by summarizing our work and discussing future works.

2. RELATED WORK

Maps and time series charts are the most common visualization techniques to present spatial and temporal data sets. Other techniques include [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19]. These techniques, however, do not deal with uncertainties in spatial or temporal dimensions even though data collected from real world often contains various levels of uncertainty because of unreliable memory, unreliable source, or the inherent ambiguity of natural language.

Much work has been done in visualizing uncertainty [20]. Here, we focus on closely related work in spatial temporal data visualization. The most common method is to overlay uncertainty information on top of a map. For example, Love et al. [21] used color coding, displacement mapping, and bar glyph on a 3D map to visualize uncertainty. Some authors also used color to visualize probabilities on a 2D map [22][23]. Zuk et al. [24] used transparency, wire frame, or location shift to present uncertainties on 3D models. Some scientific visualization methods deal with location uncertainty by plotting multiple versions of the simulations or observations, which creates a spaghetti-like drawing of data points. Other methods use contour lines or sound to indicate uncertainty. However, most of the previous works are about visualizing uncertainty data associated with location and time rather than uncertainty in location and time themselves. For example, color coding, displacement mapping, and bar glyph on a map cannot show the area of possible (but uncertain) locations. Wire frame and transparency indicate the existence of uncertainty but not the possible range of uncertain locations or times. Pebesma et al. [23] used animation to show variability in time; but, with animation, users only see one image at a time, and it's difficult to conduct data analysis on a timeline [25][26]. Most importantly, previous methods have shown difficulty integrating spatial and temporal uncertainty in one view.

The main difference between our method and previous works is that, in our method, uncertainty information is not displayed on a map but on the more abstract Storygraph. The benefit is that it can visualize both spatial and temporal uncertainty in a single 2D view. Our method can clearly differentiate between uncertainty in location (spatial uncertainty), time (temporal uncertainty) as well as a combination of the two (spatio-temporal uncertainty). In other methods, such differences are not clearly distinguishable. Our method also visualizes between-event uncertainty, which is mostly ignored by other methods. Our between-event uncertainty visualization is influenced in part by Hagerstrand's Time Geography [3][4][5], a 3D map based visualization.

3. STORYGRAPH

Storygraph is a visualization technique that presents an integrated 2D view for spatio-temporal data [1]. It is a three-axis coordinate system with two parallel vertical axes for latitude and longitude and an orthogonal horizontal axis for time. Figure 1 illustrates the basic ideas of Storygraph.

The top sub-figure in Figure 1 shows 6 accidents marked on a map. Two accidents have been reported at each location at different times of the year. However, as shown in this figure, plotting these data points on a map results in overlapping markers. For the remaining non-overlapping markers, maps fail to show the temporal distance between these events. The sub-figure at the bottom shows the same events presented in Storygraph. Here, events are plotted on the location lines with no overlapping. In addition, Storygraph presents the temporal distance between the events. Figure 2 shows a Storygraph generated from the World Trade Center (WTC) corpus generated by our program. Few patterns that can be observed in this Storygraph are: (1) The points are clustered around location $(40.70, -74.00)$, (2) At times $t1$ - $t4$ and later on around $15:12$, there are events simultaneously taking place at many different locations.

Interpreting spatial information on Storygraph is not as intuitive as that on a map; however, analyzing temporal information on Storygraph is quite intuitive. The following analysis discusses the process of interpreting the spatial information on Storygraph.

Based on [1], let α_{max} and α_{min} be the maximum and minimum latitude, and β_{max} and β_{min} be the maximum and minimum longitude. Likewise, let T_{max} and T_{min} be the maximum and the minimum timestamps.

The mapping function $f(\alpha, \beta, t) \rightarrow (x_{storygrah}, y_{storygraph})$ of event $E(l_\alpha, l_\beta, t)$ is given by:

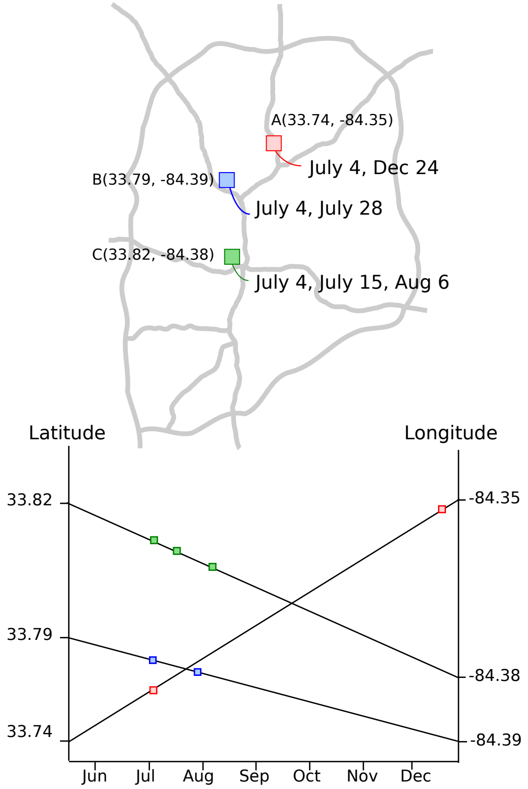


Figure 1: Example of Storygraph constructed from hypothetical accidents. Top: Outline map showing the major highways in Atlanta and hypothetical accident taking place at the junctions A, B, and C on the dates shown. Bottom: Same information plotted on Storygraph (not drawn to scale for illustrative purposes). Each location is represented as a line joining the latitude and longitude in the vertical axes. An event occurring at that location is represented by a point on the line. This representation allows users to see the temporal context of the events together with spatial context (i.e. when did most accidents take place? July-August in the figure above.)

$$y_{storygraph} = \frac{(\beta - \alpha)(x - T_{min})}{T_{max} - T_{min}} + \alpha \tag{1}$$

$$x_{storygraph} = t \tag{2}$$

Assuming $T_{min} = 0$ and $T_{max} = T$ without loss of generality, Equation 1 simplifies to

$$y = \frac{(\beta - \alpha)}{T} x + \alpha \tag{3}$$

Equation 3 is also the equation of the location line (Equation 1 rewritten in slope-intercept form).

In earlier sections, we discussed that a point on the Storygraph in the absence of location line can be mapped to range of locations in geographical space. Thus, the function f ceases to be one-to-one.

Lemma 3.1. *A point on a location line in Storygraph at*

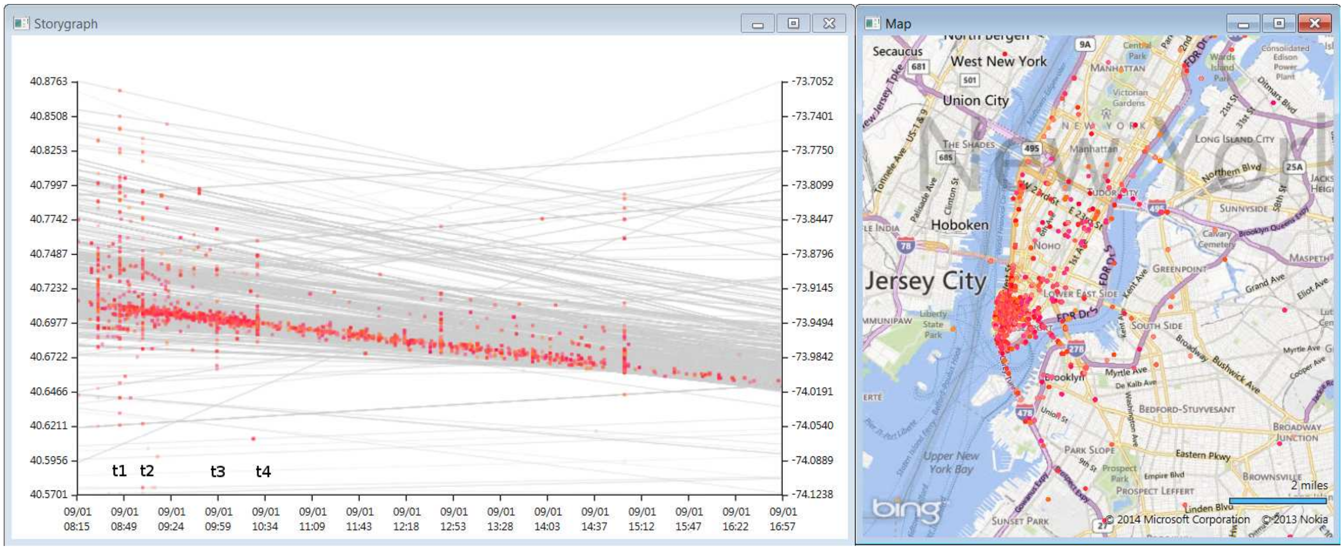


Figure 2: Left: Storygraph showing approximately 7000 events within 12 hours during 9/11 attack on WTC. Annotations $t1 - t4$ mark the key events: $t1(8:46)$, first plane crashes into the North Tower; $t2(9:03)$, second plane crashes into South Tower; $t3(9:59)$, South Tower collapses; $t4(10:28)$, North Tower collapses. At each of these times, events occurred simultaneously at multiple locations (marked by vertically aligned events). In addition, it can also be observed that the events clustered around the location $(40.70, -74.00)$. Right: Same set of events plotted on the map. Maps supplement Storygraphs as identifying locations on maps is relatively more intuitive.

time t corresponds to a precise point (geo-coordinate) on a map.

Proof. Setting $T = 0$ and $T = t$ Equation 3, we get the y_{lat} and y_{lng} of the Storygraph. Thus, geo-coordinates (α, β) can be obtained as

$$\alpha = y_{lat} \times \frac{\alpha_{max} - \alpha_{min}}{\alpha_{min} \times y_{max}} \tag{4}$$

$$\beta = y_{lng} \times \frac{\beta_{max} - \beta_{min}}{\beta_{min} \times y_{max}} \tag{5}$$

□

Lemma 3.2. *Without location lines, a point on a Storygraph at time t corresponds to a line segment on a map.*

Proof. We can rewrite equation (3) as

$$\beta = (1 - \frac{T}{x})\alpha + \frac{yT}{x} \tag{6}$$

Thus, a fixed point (x, y) on the Storygraph corresponds to many points (α, β) on the Cartesian map at time $t = x$: those $\alpha_{min} \leq \alpha \leq \alpha_{max}$ and $\beta_{min} \leq \beta \leq \beta_{max}$ satisfying (6). Plotting these values of (α, β) results in a line segment with non-positive slope since $x \leq T$ as illustrated in Figure 3. □

Lemma 3.3. *Without location lines, a vertical line segment at time t on a Storygraph corresponds to an area on a map.*

Proof. Consider a vertical line segment, with end coordinates (x, y_1) and (x, y_2), $y_1 \leq y_2$. Using 3.2, these extremes of the line segment in (6) we get two straight line equations

$$\beta = (1 - \frac{T}{x})\alpha + \frac{y_1 T}{x} \tag{7}$$

$$\beta = (1 - \frac{T}{x})\alpha + \frac{y_2 T}{x} \tag{8}$$

Hence the vertical line segment between (x, y_1) and (x, y_2) on the Storygraph corresponds to an area between two parallel lines (7) to (8) in the geographical space. As in Lemma 3.2, this area is also bounded by the maximum and minimum values of α and β – this results in a polygon as illustrated in Figure 4. □

Lemma 3.4. *Without location lines, a vertical line segment at t on the Storygraph corresponds to a projected area, $A_{Storygraph} \geq A_{actual}$ in geographical space at t.*

Proof. If the area on the plane is bounded by right rectangle, since $\forall\alpha : \alpha1 \leq \alpha \leq \alpha2$ and $\forall\beta : \beta1 \leq \beta \leq \beta2$, $A_{Storygraph} = A_{actual}$. For any other shape, the vertical line segment in the Storygraph represents a rectangular bounding box (from 3.3). Thus, $\exists\alpha : \alpha \in A_{Storygraph} - A_{actual}$. Hence, $A_{Storygraph} \geq A_{actual}$ □

Corollary 3.5. *Real-world area at time t maps to a vertical line segment in storygraph at time t.*

Proof. Inverse of Lemma 3.4, when the exact coordinates of all the four corners are known

we can state that an area A_{actual} in the geographical space gets mapped to a line segment in the Storygraph orthogonal to the time axis. The area formed by this line segment l bounded by coordinates (α_1, β_1, t) and (α_2, β_2, t) is given by $A_{Storygraph} = (\alpha_2 - \alpha_1)^2 + (\beta_2 - \beta_1)^2$. Thus, $A_{Storygraph} \geq A_{actual}$. □

Lemma 3.6. *Storygraph preserves spatial proximity for location lines but does not preserve spatio-temporal proximity for events.*

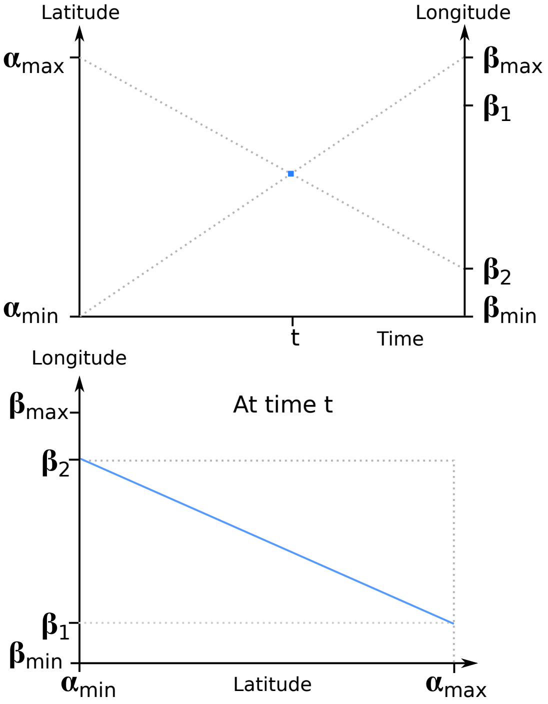


Figure 3: Top: A point in the Storygraph at time t and the corresponding location lines the point can belong to shaded. Bottom: The line segment generated in the Cartesian coordinate by mapping the point.

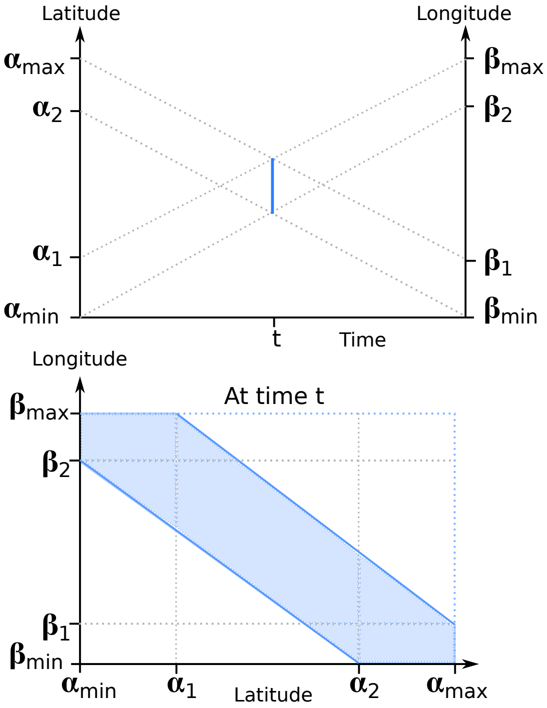


Figure 4: Top: A vertical line in Storygraph at time t and the corresponding location lines the line segment can belong to shaded. Bottom: The bounded region generated in the geographical space by mapping the line segment.

Proof. Two events close to each other in Storygraph may not be close to each other in geographical space. Consider two locations (α_1, β_1) and (α_2, β_2) in geographical space where $\alpha_1 \ll \alpha_2$. Since both of the axes are ordered in Storygraph, $\alpha_1 \ll \alpha_2$ holds true as well. □

4. CLASSIFICATION OF UNCERTAINTY

Different classifications of uncertainty have been proposed [27][28]; however, most of these classifications are about uncertainties introduced in scientific experiments or probabilistic models. In our case, uncertainties are introduced in narratives. Thus, we classify this kind of uncertainty into three categories:

Uncertainty about time and/or location of the event. This type of uncertainty is characterized by the presence of phrases denoting uncertainty before temporal or spatial description. An example is "I got there maybe around 11 am." The phrase 'maybe around' adds uncertainty to '11 am' in this example. Such uncertainties may also arise from ambiguity in language. For example, in "I was in Brooklyn when the plane hit the building," the word 'Brooklyn' does not give a precise location. We call these types of uncertainties as *event uncertainty* which can be further divided into three sub-categories:

- *Spatial uncertainty.* This category includes events that have precise time stamps but uncertain location.
- *Temporal uncertainty.* In addition to uncertain phrases (e.g. maybe, about), temporal uncertainty may come from the language itself. For example, in "I was at the station in the all day," the phrase "all day" without any modifier can refer to a wide range of time introducing uncertainty.
- *Spatio-temporal uncertainty.* This category includes events that have uncertainty in both time and location. For example, in "It was in the afternoon, I was heading south." The words 'afternoon' and 'south' are uncertain.

Uncertainty between two events. In "It was 8 in the morning I was at home. As soon as I heard about it, I reached the site at 10.", the first event ("at home") and the second event ("reached the site") are both certain. However, what happened between the two events is unknown. We call this type of uncertainty *between-even uncertainty*

Uncertainty about the even taking place. In the WTC corpus, we often encounter sentences like "I think Chief pulled me back". The word 'think' indicates an uncertainty about whether the event has ever happened. Detecting this type of uncertainty is difficult and beyond the scope of this paper. Instead, we focus only on visualizing event uncertainty and between-event uncertainty.

5. EVENT UNCERTAINTY

In this section, we discuss the extraction and visualization of event uncertainty. To extract event uncertainty, we compiled a list of English words that may indicate location uncertainty, such as "around," "near," "close to," "maybe," "perhaps," etc. We then gave each word an uncertainty

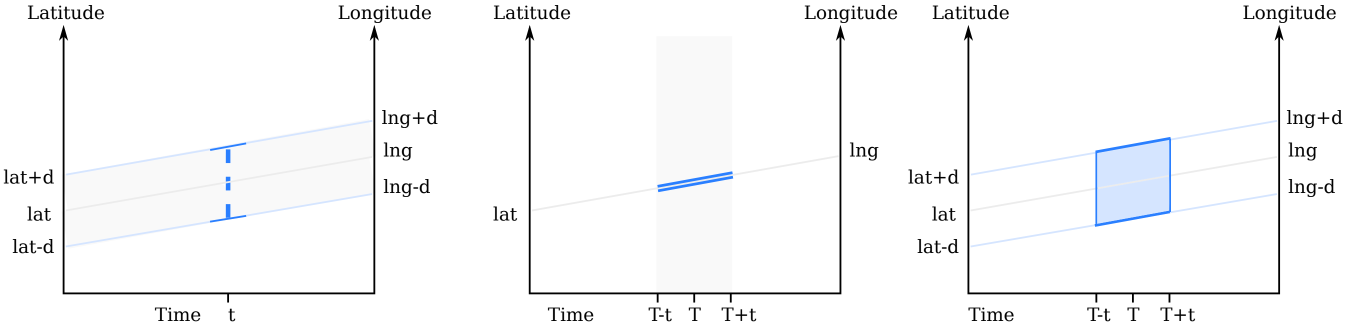


Figure 5: Three kinds of glyphs used to represent spatial, temporal and spatio-temporal uncertainty. Left: Dashed I-beam is used to represent spatial uncertainty. The slope of the top and bottom of the beam disambiguates the range of locations in the geographical space. Middle: parallel lines are used to denote the temporal uncertainty. Right: Box showing spatio-temporal uncertainty. The slope of the edges of the box maps to a fixed geographical area within a certain time.

score in the range of $1 - 100$ [29][30][31]. The same process was repeated for temporal information. We extracted the named entities from WTC corpus using Stanford NER [32] and time using SuTime [33]. TARSQI [34] was used to extract the temporal sequence of the events, and locations were geocoded using Google Maps API. The results were then verified and corrected.

In the WTC corpus, we observed all three types of event uncertainties: spatial, temporal, and spatio-temporal. Some key events with precise spatio-temporal information were used as anchor events. These include the first and second plane hitting the tower, and the plane crashing into the pentagon. These events were chosen as key events because all of the interviews described more local events in reference to these global events. Examples include “When the second plane hit the tower, I was running towards Vesey,” and “I was at the station when the news about the first explosion was on TV.” When considering these key events in the context of the first example, the time is certain but the location is uncertain. Additionally, in a sentence that references no key events like “When the EMS arrived at the scene, I began heading south”, both location and time would be considered uncertain.

For each event, latitude, longitude, date/time, color, spatial uncertainty, and temporal uncertainty were fed to the visualization program, which then visualized the uncertainty information along with other information.

Spatial Uncertainty. Spatial uncertainty is visualized as a vertical dashed I-beam. From Corollary 3.5, we know that an area on a map corresponds to a line in Storygraph. The length of the I-beam is proportional to the area of possible locations. More importantly, the top and the bottom of the beam disambiguate the range of locations in geographical space. This is shown by the left sub-figure in Figure 5.

Temporal Uncertainty. We use sloped double lines to represent temporal uncertainty. Each double line is drawn along the location line for the corresponding event, which can be seen in the middle sub-figure in Figure 5. A double line indicates that the event happens at a particular location within a certain time frame. In contrast, a single solid line along the location line means that the character stayed at the specified location for a period of time. Through these representations, the two cases are visually distinct.

Spatio-temporal Uncertainty. We use a semi-transparent box to visualize spatio-temporal uncertainty, which means both location and time are uncertain. The sloped top and bottom sides of the box indicate the range of locations while the vertical sides of the boxes shows the temporal bound. The box is drawn as semi-transparent to prevent glyph occlusion. This is shown by the right sub-figure in Figure 5.

Figure 6 shows this concept applied to the events extracted from WTC corpus.

6. BETWEEN-EVENT UNCERTAINTY

The purpose of visualizing between-event uncertainty is to display the space-time constraints between two key events. Any activity takes place within a certain span of time and a certain geographical region. Individuals participating in these activities have to trade time for space or vice versa. For example, during a workday lunch hour a person could walk to a nearby restaurant for a longer meal or drive to distant restaurant for a shorter meal. Visualizing between-event uncertainty can assist planning, scheduling, analyzing possible overlapping in people’s activities.

Our between-event visualization technique is partially based on Hagerstrand’s Time Geography, a conceptual framework which focuses on constraints and trade-offs in the allocation of time among activities in space [5]. However, Time Geography is a map based 3D visualization. Therefore it suffers from the typical problems associated with 3D visualizations, such as 3D occlusion and difficulty of navigation. Besides, space and time are not well integrated in Time Geography. Our work is an attempt to address these issues.

6.1 Space-time paths and space-time prisms

We adapted two important concepts from Time Geography: space-time paths and space-time prisms. Space-time path traces the movement of a character in space and time. Figure 7 shows an example of a space time path adapted from [5]. The base plane is the geographical space and the orthogonal axis is time. In this example, an individual travels from location 1 to 2, spends some time at 2 and then moves on to 3. The time and location of the starting or end point are known as *control points* or *key events.* The straight line segments connecting two control points are known as *path segments.* Path segments are represented by straight line segments for simplicity [35][36]. In our earlier work, we

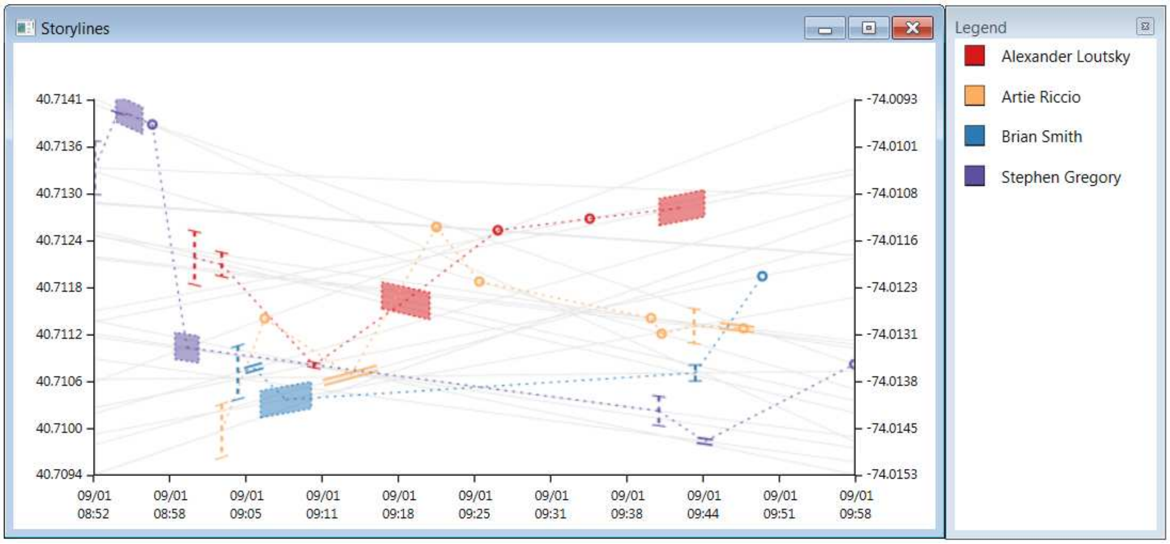


Figure 6: Storylines of four firefighters before the second tower collapsed along with event uncertainty. The dashed vertical I-beam shows the spatial uncertainty. The slope of the top and bottom portion of the beam shows the possible range of locations. The parallel lines show temporal uncertainty. The boxes represent spatio-temporal uncertainty and the circles show certain events.

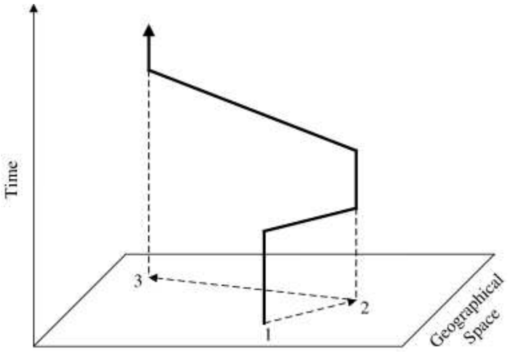


Figure 7: An example of space time path adapted from [5]. Space time paths trace the movement of an individual moving from one location to another. Space-time paths also show the amount of time spent at a location by the individual before moving to the next location.

adapted the concept of space-time paths in Storygraph using storylines[1]. Here, Storylines become space-time paths, connecting two consecutive key events via dotted line segment.

Space-time prisms extend space time paths to create a 3D space consisting of all the possible routes an individual can take while moving from one point to another. This space is known as the *potential path space*. The prism between $t1$ and $t2$ in Figure 8 demonstrates this concept. The slope of the edges of this prism is determined by the inverse of maximum velocity. That is, the possible paths are constrained by the maximum velocity of the individual, a fixed time frame, and fixed destinations. In our implementation, the maximum velocity is set by the user.

If an individual is at origin, o, at t_o and needs to reach destination, d, at t_d, the time budget is $T = t_d - t_o$. The path space from the origin under the time budget is shown by the red inverted dotted cone. This space shows all the possible paths and all the possible locations that can be reached within the time budget with maximum velocity v. Let this region be denoted by $R_o(T)$. Similarly, the blue dotted cone shows the path space towards d under the time budget. This 3D space gives all the locations from where d can be reached under time T. Let this region be $R_d(T)$. The intersection of these cones give the potential path space for individual traveling from o to d [37]. Hence,

$$R_{od}(T) = R_o(T) \cap R_d(T) \tag{9}$$

The projection of the space time prism on the geographical space, as shown by a gray circle in the figure, shows all the possible locations that the user can reach. This area is called the *potential path area*.

Given all the control points within a specific time window, τ, the construction of space-time prism requires the destination d to lie within the $R_o(T)$ and vice versa. Stating it formally,

$$\forall o, d \in \phi_\tau : (o \cap R_d(T) \neg \emptyset) \wedge (d \cap R_o(T) \neg \emptyset) \tag{10}$$

In Time Geography, space-time paths and space-time prisms are generally drawn inside a 3D space-time cube [38] (Figure 8). In our work, space-time paths and space-time prisms are drawn on Storygraph in a 2D view.

6.2 Visualizing between-event uncertainty

Storygraph draws space-time prisms based on Equations 9 and 10. From Corollary 3.5, we know that an area in the geographical space is mapped to a line in Storygraph. Thus starting from a location, $o(\alpha, \beta)$, at $t0$ and taking a snapshots of the potential path area at each time step we get a set of areas sequentially increasing at the rate of the velocity.

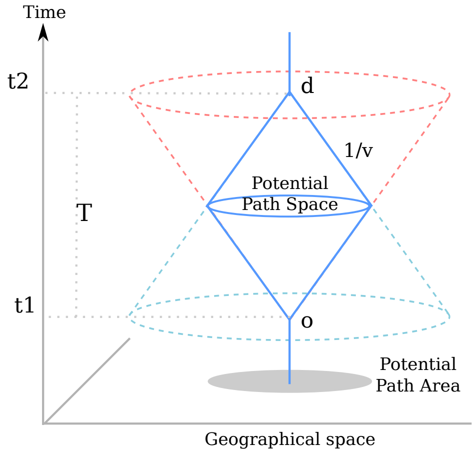


Figure 8: Space-time prism. In this figure, the individual is at location o at $t1$ needs to be at the same location d at $t2$ (o origin of travel). (S)he has the time budget of T. The red dotted cone shows the possible path space starting from o with the maximum velocity, v. Similarly the blue dotted cone shows the path space towards d. The intersection of these two cones gives the potential path space under the given time budget T. The potential path area is shown by the gray area on the geographic space.

The top sub-figure in Figure 9 shows an individual at point (α, β) at $t0$ and his/her possible path area after each time step $t1 - t5$. The figure simplifies the drawing of the potential path areas by representing them with squares rather than circles. The bounding of the actual potential path by squares introduces some uncertainty itself [20] but greatly simplifies the drawing and the calculations.

Hence, if the time step, $\Delta t \to 0$, then conical region $R_o(T)$ would be reduced to a triangular region in Storygraph. This region is shown by the area enveloped by two gray lines in the bottom sub-figure in Figure 9.

Figure 10 shows the result of mapping the space-time prism in Figure 8 in Storygraph. The mapping process resembles the drawing of the space-time prism inside the space-time cube. Given two control points, maximum velocity and a time budget, these parameters are plugged into Equation 10 to check whether the control points satisfy this criteria. If the criteria is satisfied, we compute the extents (lat_{max}, lng_{max}) and (lat_{min}, lng_{min}) of the $R_o(T)$ with the following sets of equations,

$$lat_{max} = max_{lat_r}[\sqrt{(lat_r - lat_o)^2 + (lng_r - lng_o)^2} = vT] \quad (11)$$

$$lng_{max} = max_{lng_r}[\sqrt{(lat_r - lat_o)^2 + (lng_r - lng_o)^2} = vT] \quad (12)$$

$$lat_{min} = min_{lat_r}[\sqrt{(lat_r - lat_o)^2 + (lng_r - lng_o)^2} = vT] \quad (13)$$

$$lng_{min} = min_{lng_r}[\sqrt{(lat_r - lat_o)^2 + (lng_r - lng_o)^2} = vT] \quad (14)$$

Similarly, the extents for the $R_d(T)$ is calculated. Finally, $R_{od}(T)$ is obtained from the intersection of these regions.

6.3 Intersections of prisms in Storygraph

Space-time paths and prims are both based on the movement data of characters. Given a dataset containing the movement data of two or more individuals, it is likely that the space-time prisms will overlap. However, since Storygraph does not preserve event proximity (Lemma 3.6), it is important to note that these overlaps may not necessarily mean that these prisms intersect in geographical space.

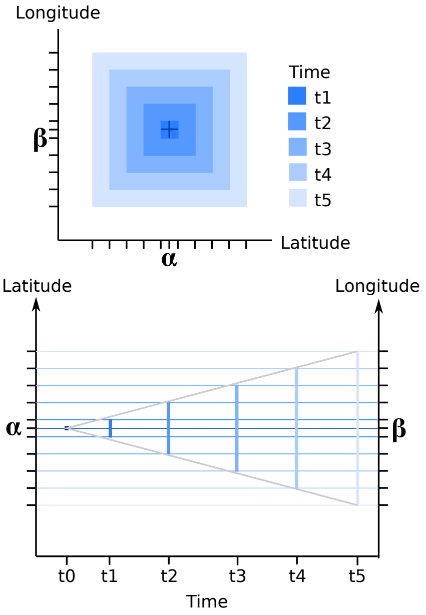


Figure 9: Above: Starting from the origin of travel, α, β, at $t0$, the potential path areas in each time step $t1 - t5$ (assuming a certain velocity and regular time intervals). Below: The same data plotted in Storygraph. Each potential path area is mapped to a line segments in Storygraph. For continuous time, this would result in an area enveloped by two gray lines. The slope of the gray lines is equal to the maximum velocity.

Hence, given a point p and a prism $R_a(T)$ in the Storygraph, we first establish the conditions for a valid point-prism intersection. Building on this, we then present the validity of intersection between two prisms.

Let $R_a(T) : R_a(T) = R_o(T) \cup R_d(T)$, be all the possible locations that the individual can travel within the time budget T with a maximum velocity v. Then following cases for point-prism intersection could arise:

1. The point is not inside the prism but the location line is inside $R_a(T)$. This case implies that the event occurred within the geographical bounds but the individual may not have been involved in the event due to the travel constraints.

2. The point is inside the prism but the location line is not inside $R_a(T)$. This implies that the event occurred within the time span, T, but at some other location $\notin R_a(T)$.

3. The point is inside the prism and location line is inside $R_a(T)$. This is the only case where the individual could have been involved in the event.

THEOREM 6.1. *For a valid point-prism intersection, the point should be inside the prism and the location line should lie inside $R_a(T)$.*

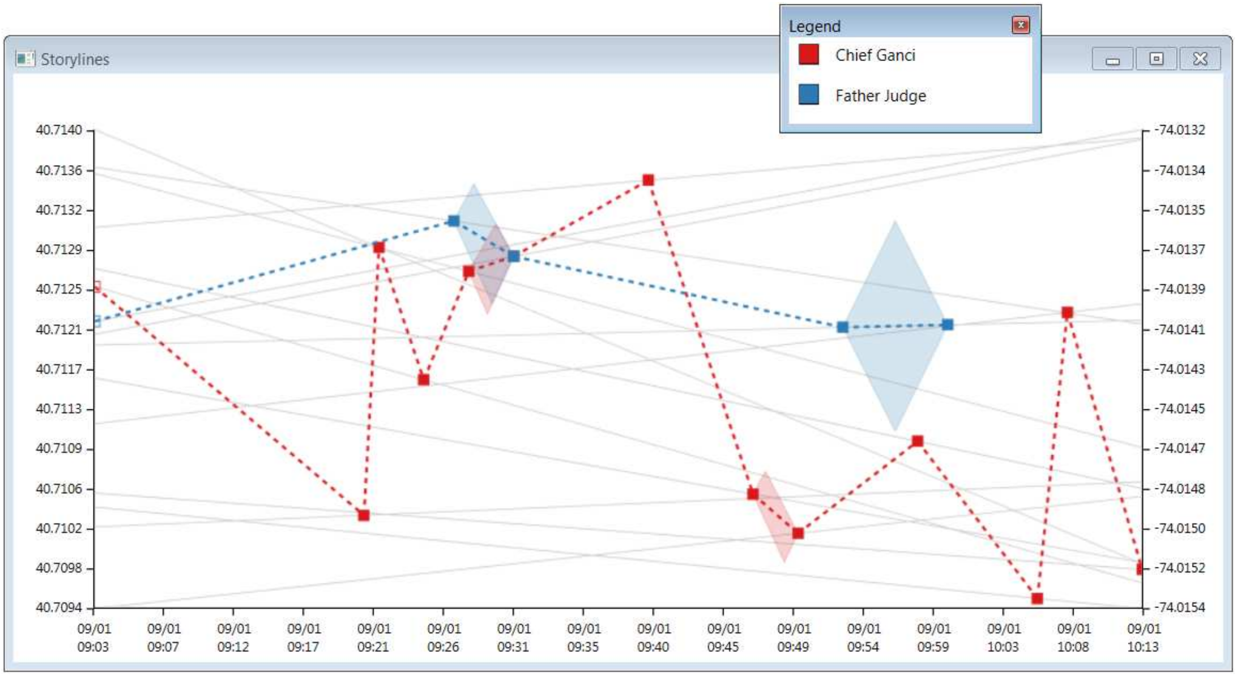


Figure 11: Storylines of WTC victims Chief Ganci and Father Judge. Father Judge was officially identified to be the first victim of the incident. Only a few key points have prisms between them. For other key points, the distance between them cannot be travelled within the given time at a velocity set by the user. This could either mean missing data points, change in velocity, or data reporting error.

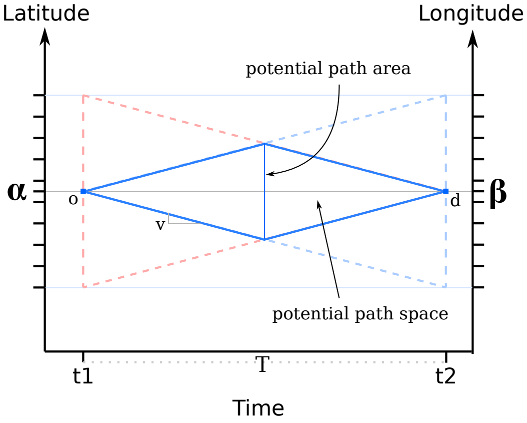


Figure 10: The space-time prism shown in Figure 8 drawn in Storygraph.

PROOF. Assume that this is not a valid intersection. It means that the point representing the event is either spatially or temporally incorrect. This is temporally incorrect because for a point to lie inside the prism, it has to occur within the time budget. This is spatially incorrect since $R_a(T)$ defines the maximum distance an individual can travel at within a time T. □

Hence, given two prisms, $P1$ and $P2$, the prism-prism intersection is only valid if there exists a point p on location line l such that $l \in R_a^{P1}(T) \wedge l \in R_a^{P2}(T) \wedge p \in P1 \cap P2$.

7. CASE STUDY: WTC 9/11

In the immediate aftermath of the attacks in New York on September 11, 2001, the NYC Fire Department convened a task force to interview first responders to the affected areas. These 511 interviews, conducted in the two months following the attacks, were later released by the New York Times. Each interview was conducted by staff from the New York Fire Department assigned to the task force and ran anywhere from 8-20 minutes with the aim to elicit from first responders their activities on September 11. The language of the reports is typical of event interviews and oral histories. Despite having a population with high area knowledge and normalized reporting practices, locations and times were predominately referred to referentially. Known individuals seen by the interviewee are named, but most are either not named or referred to solely by rank. The primary reason to visualize this data is to enable historians and investigators to identify accurate and inaccurate information and to allow for more ready recognition of corroborating evidence. When viewed as a corpus rather than separate interviews, it becomes possible to identify overlaps in the reported events of the witnesses. The challenge posed to this task by the referential language usage of the witnesses is pervasive in oral history and other investigatory work reliant on interviewing.

Event Uncertainty Visualization. Time, location, and characters (or people) in this corpus were extracted using Java code and the aforementioned natural language processing tools. Each event was given an uncertainty score using the method described in Section 5. We first drew a Storygraph without uncertainty information (Figure 2). In this figure, key events – such as when the first and second plane hit and when the towers collapsed – are shown by $t1 - t4$. There are many co-occurring events around 15 : 00 hrs, but the causes of these patterns are not yet clear.

Next, we plotted the storylines of four fire fighters before the South Tower collapsed with event uncertainty (Figure 6). It should be noted that two storylines crossing does not necessarily mean the two characters encounter each other; rather, it only means that two people were moving in directions diagonal to each other. One limitation of using uncertainty glyphs is that they might result in occlusion and ambiguity for large datasets. When the dataset is large, the bigger glyphs (e.g. the ones representing spatio-temporal uncertainty) could occlude the smaller ones.

Between-event Uncertainty Visualization. Figure 11 visualizes the between-event uncertainty for two victims: Father Judge and Chief Ganci. The space-time prisms in Storygraph enable users to see the possibilities of individuals encountering each other between key events. There are two patterns in this figure: (1) the prisms are only present between some key events, and (2) some prisms overlap. The first pattern indicates that locations of the two events are too far apart in that it would be impossible for a person to cover that distance at the maximum velocity. It does not necessarily mean that part of the story is false; rather, it may be the result of missing information between two events or uncertainty in the events themselves. From overlapping prisms (from Theorem 6.1), we can also deduce that Chief Ganci and Father Judge might have encountered each other within that time frame and region.

8. CONCLUSION

In this paper, we presented a new method for visualizing uncertainty in spatio-temporal data set. This method is an extension of our previous work Storygraph, a visualization technique for displaying spatio-temporal data sets in an integrated 2D view. Our method can visualize both temporal-spatial uncertainty about an event and the uncertainty between events. This extended method provides more accurate and faithful visualization of spatio-temporal data sets with inherent uncertainties. In addition, between-event uncertainty visualization can help users analyze the feasibility of spatio-temporal events and possible encounters between multiple characters. We demonstrated this method in a case study.

In the future we plan to conduct user studies to evaluate the effectiveness of this method and compare it with other methods. We also plan to investigate new methods to analyze and visualize uncertainty in the identification of people or groups.

9. ACKNOWLEDGEMENT

This project is supported in part by National Science Foundation Grant No. 1209172. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

10. REFERENCES

[1] Ayush Shrestha, Ying Zhu, Ben Miller, and Yi Zhao. Storygraph: Telling stories from spatio-temporal data. In *Advances in Visual Computing*, pages 693–702. Springer, 2013.

[2] A. Shrestha, Ying Zhu, and B. Miller. Visualizing time and geography of open source software with storygraph. In *Software Visualization (VISSOFT), 2013 First IEEE Working Conference on*, pages 1–4, Sept 2013.

[3] Torsten Hägerstrand. Reflections on “what about people in regional science?”. In *Papers of the Regional Science Association*, volume 66, pages 1–6. Springer, 1989.

[4] Nigel Thrift. An introduction to time-geography. Geo Abstracts, University of East Anglia, 1977.

[5] Harvey J Miller. A measurement theory for time geography. *Geographical analysis*, 37(1):17–45, 2005.

[6] Wolfgang Aigner, Silvia Miksch, Wolfgang Muller, Heidrun Schumann, and Christian Tominski. Visualizing time-oriented data - a systematic view. *Computers and Graphics*, 31(3):401 – 409, 2007.

[7] W. Aigner, S. Miksch, W. Muller, H. Schumann, and C. Tominski. Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):47 –60, jan.-feb. 2008.

[8] D. Fisher, A. Hoff, G. Robertson, and M. Hurst. Narratives: A visualization to track narrative events as they develop. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 115 –122, oct. 2008.

[9] K. Vrotsou, J. Johansson, and M. Cooper. ActiviTree: Interactive visual exploration of sequences in event-based data using graph similarity. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):945 –952, nov.-dec. 2009.

[10] W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927 –934, nov.-dec. 2010.

[11] T. Gschwandtner, W. Aigner, K. Kaiser, S. Miksch, and A. Seyfang. CareCruiser: Exploring and visualizing plans, events, and effects interactively. In *Proceedings of IEEE Pacific Visualization Symposium (PacificVis)*, pages 43 –50, march 2011.

[12] Zhao Geng, ZhenMin Peng, R.S. Laramee, R. Walker, and J.C. Roberts. Angular histograms: Frequency-based visualizations for large, high dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2572 –2580, dec. 2011.

[13] Jian Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory analysis of time-series with ChronoLenses. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2422 –2431, dec. 2011.

[14] M. Krstajic, E. Bertini, and D. Keim. CloudLines: Compact display of event episodes in multiple time-series. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2432 –2439, dec. 2011.

[15] R. Maciejewski, S. Rudolph, R. Hafen, A. Abusalah, M. Yakout, M. Ouzzani, W.S. Cleveland, S.J. Grannis, and D.S. Ebert. A visual analytics approach to understanding spatiotemporal hotspots. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):205 –220, march-april 2010.

[16] Ali Asgary, Alireza Ghaffari, and Jason Levy. Spatial

and temporal analyses of structural fire incidents and their causes: A case of Toronto, Canada. *Fire Safety Journal*, 45(1):44 – 57, 2010.

[17] Charlotte Plug, Jianhong (Cecilia) Xia, and Craig Caulfield. Spatial and temporal visualisation techniques for crash analysis. *Accident Analysis and Prevention*, 43(6):1937 – 1946, 2011.

[18] Gennady Andrienko, Natalia Andrienko, Martin Mladenov, Michael Mock, and Christian Politz. Identifying place histories from activity traces with an eye to parameter impact. *IEEE Transactions on Visualization and Computer Graphics*, 18(5):675–688, May 2012.

[19] SungYe Kim, R. Maciejewski, A. Malik, Yun Jang, D.S. Ebert, and T. Isenberg. Bristle maps: A multivariate abstraction technique for geovisualization. *Visualization and Computer Graphics, IEEE Transactions on*, 19(9):1438–1454, Sept 2013.

[20] K. Brodlie, R. A. Osorio, and A. Lopes. A review of uncertainty in data visualization. In *Expanding the Frontiers of Visual Analytics and Visualization*. Springer, London, 2012.

[21] Alison Love, Alex Pang, and David Kao. Visualizing spatial multivalue data. *IEEE Computer Graphics and Applications*, 25(3):69–79, 2005.

[22] Lydia Gerharz, Edzer Pebesma, and Harald Hecking. Visualizing uncertainty in spatio-temporal data. In *Spatial Accuracy 2010*, pages 169–172, 2010.

[23] Edzer J. Pebesma, Kor de Jonga, and David Briggs. Interactive visualization of uncertain spatial and spatio-temporal data under different scenarios: an air quality example. *International Journal of Geographical Information Science*, 21(5):515–527, 2007.

[24] T. Zuk, S. Carpendale, and W. D. Glanzman. Visualizing temporal uncertainty in 3d virtual reconstructions. In *Proceedings of the 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST 2005)*, pages 99–106, 2005.

[25] Mark Harrower. The cognitive limits of animated maps. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 42(4):349–357, 2007.

[26] Kirk Goldsberry and Sarah Battersby. Issues of change detection in animated choropleth maps. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 44(3):201–215, 2009.

[27] Kristin Potter, Paul Rosen, and Chris R Johnson. From quantification to visualization: A taxonomy of uncertainty visualization approaches. In *Uncertainty Quantification in Scientific Computing*, pages 226–249. Springer, 2012.

[28] Osman Abul, Francesco Bonchi, and Mirco Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 376–385. Ieee, 2008.

[29] Lee D Erman, Frederick Hayes-Roth, Victor R Lesser, and D Raj Reddy. The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys (CSUR)*, 12(2):213–253, 1980.

[30] Marie-Catherine de Marneffe, Christopher D. Manning, and Christopher Potts. Veridicality and utterance understanding. In *Proceedings of the 2011 IEEE Fifth International Conference on Semantic Computing*, ICSC '11, pages 430–437, Washington, DC, USA, 2011. IEEE Computer Society.

[31] Roser Sauri and James Pustejovsky. Factbank: a corpus annotated with event factuality. *Language Resources and Evaluation*, 43(3):227–268, 2009.

[32] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[33] Angel X Chang and Christopher Manning. Sutime: A library for recognizing and normalizing time expressions. In *LREC*, pages 3735–3740, 2012.

[34] Marc Verhagen, Inderjeet Mani, Roser Sauri, Robert Knippen, Seok Bae Jang, Jessica Littman, Anna Rumshisky, John Phillips, and James Pustejovsky. Automating temporal annotation with tarsqi. In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pages 81–84. Association for Computational Linguistics, 2005.

[35] Dieter Pfoser and Christian S Jensen. Capturing the uncertainty of moving-object representations. In *Advances in Spatial Databases*, pages 111–131. Springer, 1999.

[36] José Moreira, Cristina Ribeiro, and Jean-Marc Saglio. Representation and manipulation of moving points: an extended data model for location estimation. *Cartography and Geographic Information Science*, 26(2):109–124, 1999.

[37] Tijs Neutens, Nico Weghe, Frank Witlox, and Philippe Maeyer. A three-dimensional network-based spaceâĂŞtime prism. *Journal of Geographical Systems*, 10(1):89–107, 2008.

[38] Menno-Jan Kraak. The space-time cube revisited from a geovisualization perspective. In *Proc. 21st International Cartographic Conference*, pages 1988–1996, 2003.

NIA: System for News Impact Analytics

Mikalai Tsytsarau
University of Trento
tsytsarau@disi.unitn.eu

Themis Palpanas
Paris Descartes University
themis@mi.parisdescartes.fr

ABSTRACT

The analysis of news impact on people is relevant to a variety of applications, ranging from monitoring product and companies reputations, to stock market prediction. Therefore, it is important to understand the underlying mechanisms which affect the propagation of news and drive the evolution of sentiments in one way or another. In this demonstration paper, we describe NIA, a system that identifies and describes news events that caused changes of sentiments. NIA is based on a novel framework for a complex news event modeling, which is capable of detecting time and importance characteristics of events by only observing a time series of news articles publications, and then correlating this data with a time series of sentiment shifts. The operation of our system is summarized as follows. First, we apply a deconvolution to recover the time, longitude, importance and impact of news events. Second, we compute a sentiment time series, e.g., by monitoring sentiments for positive or negative bursts, and coherently analyze sentiment and news time series, automatically determining their time lag. Third, we evaluate the corresponding news articles for a time interval of interest and extract the essence of what happened. Finally, we present the selected news time series to the user, as well as several more correlated stories, which could have affected sentiments as well, proposing to interactively explore their connections.

1. INTRODUCTION

Today, sentiment analysis has become a platform that provides valuable information on people's opinions regarding different topics, and is widely used by businesses and social study institutions [6]. By aggregating sentiments, expressed in multiple texts, and assessing the result with statistical measurements, we can capture certain changes, or shifts, in global sentiment, which cannot be attributed to random variation [5]. Recent studies indicate that the observed sentiment changes can be the result of people reacting differently to external events [4, 7], opening this problem for the investigation.

In this demo, we aim at determining the impact of news events on sentiment changes. However, most of the news events are announced as atomic pieces of information and their importance is not readily intelligible from the text alone. To determine the importance and impact of news to people, it is crucial to consider the relevant publication dynamics of the whole crowd, rather than only from news agencies or news media. Whats more, it is important to analyze all types of sentiment shifts, which could be connected with news events. The problem is that relevant sentiment shifts can be particularly small and can occur before, during or after the event - all with varying delays, depending on the event type and publication pace of the media. This necessitates the sophisticated news and sentiment extraction, aggregation and tracking methods, as well as proper correlation measures between news and sentiments.

Such problems require processing significant amounts of data to produce a desired output, from sentiment extraction to event processing. However, the most challenging part of our problem is finding relevant pairs of news events and sentiment shifts, because there is usually no one-to-one correspondence between the event and sentiment shift types, and there can also exist multiple correlated topics, which contribute to sentiment deviations. At this step, the interaction with the user in order to pick up such cases can be very beneficial, since the system can quickly filter through the correlated topics, but only human can understand the semantic connection (and causality) between events and sentiments.

This demonstration features NIA [7] - a system for news and sentiment analytics, which monitors important news events, evaluates their dynamics, and captures the correlated sentiment changes. Our system aims to predict which event types are likely to cause the sentiment to change by analyzing news importance and dynamics and letting the user to explore the connections between time series of sentiment shifts and news events for correlated topics.

The NIA system relies on principled techniques and approaches to news and sentiment aggregation and analysis: (a) we employ automated parameter optimization for processing time series, detecting news events and measuring their characteristics; (b) sentiment noise and irregularity are reduced by regression smoothing, taking into account the diversity and significance of sentiments.

Motivating Scenarios and Examples:

Example 1: We want to detect changes in the opinion on a particular topic, when such changes are caused by news events. For instance, imagine the situation demonstrated in Figure 1, in which the sentiment expressed in Twitter for the Large Hadron Collider (LHC) has dropped from positive to negative just after the first experiments begun. In our example, we see that people started to talk negatively in the aftermath of the first experiments (marked *“collision"*), while the news about the record beam energy (marked *“record energy"*) pushed sentiments back to neutral. To understand the difference between these two events we need to navigate to a correlated news trend and analyze the volume of news around these sentiment changes. However, proper news event detection and processing require special methods, as shown in our next example.

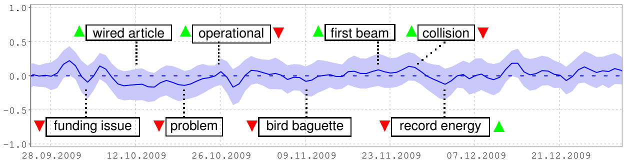


Figure 1: Sentiment shifts for the topic "LHC" from Twitter.

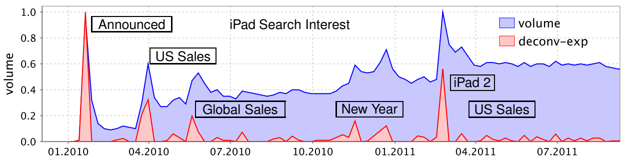


Figure 2: Events identified by deconvolution for “iPad”.

Example 2: Consider a *search interest* time series extracted from Google for the topic “iPad”, shown in Figure 2, blue. It features a growing number of search queries overlaid with a series of overlapping bursts of user interest, making it very hard to detect news events. For instance, the relative difference in interest between *“iPad 2"* and the following *“US Sales"* events is obstructed by the trend, which makes their volumes appear similar. The output time series of events (Figure 2, red), processed using our method, demonstrates a more vivid event separation, making them easily detectable. Moreover, it appears without the global trend, revealing true event importance and dynamics.

2. NEWS IMPACT ANALYTICS

Our system for news impact analytics, `NIA` [7], addresses the problems of detecting interesting changes of aggregated sentiment and connecting them to relevant news events which could have caused these situations. In this section we briefly introduce the main capabilities and design principles of `NIA`, proceeding with the description of its main features and the demonstration scenario in Section 3.

2.1 System Overview

Figure 3 outlines the composition of `NIA`. It consists of *Sentiment Analysis* and *News Event Analysis* layers, which analyze aggregated sentiment data and news volume as described below. The sentiment analysis layer takes care of aggregating sentiments for a topic and detecting interesting changes, which can be contradictions, outbursts of sentiments’ volume or other changes in sentiment happening over time. The news event analysis layer works with time series of publication volume (which can be news, blog posts, tweets) to detect various events that could have caused the observed shifts in sentiment. Events are detected with the help of deconvolution, through observing outbursts in news volume, and are annotated automatically by summarizing relevant news articles.

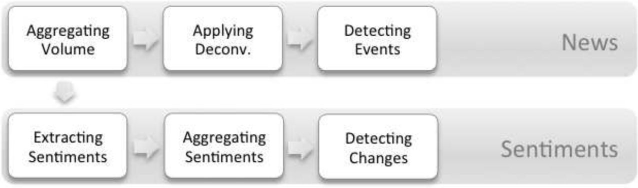


Figure 3: Compositional diagram of the system.

2.2 Sentiment Analysis

We determine topic T and sentiment S for each text and assign a continuous sentiment value S in the range [-1;1] that indicates the polarity of the opinion expressed regarding the topic. For the sentiment assignment step, we use the SentiStrength [4] tool, which recognizes opinion expressions, emoticons and works especially well for short texts, like tweets.

For analyzing news impact, we are interested in sentiment measures that are sensitive to particular kinds of sentiment changes, usually correlated with events. However, not many studies propose suitable measures for opinion shifts, which can be analyzed coherently with the news time series in order to extract correlations. The particular methods which can be adopted to our problem are *sentiment volume* [4] and *contradiction level* [8], discussed below.

Sentiment Volume is defined as the amount or the sum of sentiments of a particular polarity, expressed within a specified time interval [4]. It captures bursts of particular opinions, e.g., *positive*:

$$s(t) = \sum_{i=1}^{n} S_i^+(t), \quad \text{or} \quad s(t) = |S_i^+|(t)$$

Contradiction Level is another suitable measure for sentiment shifts, that can detect both changes of sentiment polarity as well as temporary shifts of sentiments [8]. The intuition behind this measure is that when the aggregated value for sentiments μ_S is close to zero, while the sentiment diversity (variance) σ_S^2 is high, then the contradiction should be high. Combining μ_S and σ_S^2 in a single formula, we propose the following measure for contradictions:

$$s(t) = \frac{\vartheta \cdot \sigma_S^2}{\vartheta + (\mu_S)^2} W(n),$$

where n is the number of sentiments, $\vartheta \neq 0$ is the normalizing constant, and W is a weight function that takes into account the significance of sentiment statistics involved in the calculation [8].

2.3 News Event Analysis

We consider that sentiment changes can be preceded with or followed by news events. A time lag between the two sequences can be determined by maximizing their cross-correlation coefficient. It can then be used to navigate to the relevant news event, given a time interval of sentiment shift, annotating it with the keyword description and importance dynamics.

Extracting News Time Series. An example time series of news volume is shown in Figure 2. It consists of a series of bursts of varying height and length, which can even be overlapping. Constructing the news volume time series $n(t)$ for a specific topic involves the analysis of documents in the collection $\mathscr{D}$ and estimation of topic’s popularity (frequency) among them. For example, we can count a number of documents D_i which have occurrences of the topics’ keywords T, or sum their TF-IDF scores:

$$n(t) = |\mathscr{D}^T|_t = \{D_i \in \mathscr{D} \mid T \in D_i\}; \quad n(t) = \sum_{D_i \in \mathscr{D}^T} \mathit{TF\text{-}IDF}(T, D_i)|_t$$

Detecting Impacting Events. As we already noted, not every kind of publications outbursts is caused by external news, and not every kind of news dynamics has an impact on sentiment, so we want to distinguish them at a fine level of detail, for instance, distinguishing between *linear*, *hyperbolic* or *exponential* response types, either symmetric or asymmetric, depending whether events are anticipated or not. Our system represents news publication volume as the result of the interplay between the original news’ importance $e(t)$ and media response $mrf(t)$, in a process known as convolution. In order to recover $e(t)$, we perform a deconvolution of news volume time series, using Fourier transformation, as described in [7]:

$$e(t) = \mathscr{F}^{-1}\{e(\omega)\} = \mathscr{F}^{-1}\{n(\omega)/mrf(\omega)\}$$

Unlike other models [2, 1, 3], describing publication dynamics by complex equations, deconvolution uncovers *succinct* and *meaningful* event parameters in the form of $e(t)$, such as: event’s interest *buildup and decay*, its *longitude* and *maximum importance* level.

Extracting Event Annotations. To automatically annotate news event, we compare TF-IDF scores of the news documents within a current time interval to the same scores over the entire collection of news, and extract top k terms, which became more popular in the event time interval $\mathscr{D}_e^T$:

$$T_{event} = \{T_j \mid max_k(\mathit{TF\text{-}IDF}(T_j, \mathscr{D}_e^T) - \mathit{TF\text{-}IDF}(T_j, \mathscr{D}^T))\}$$

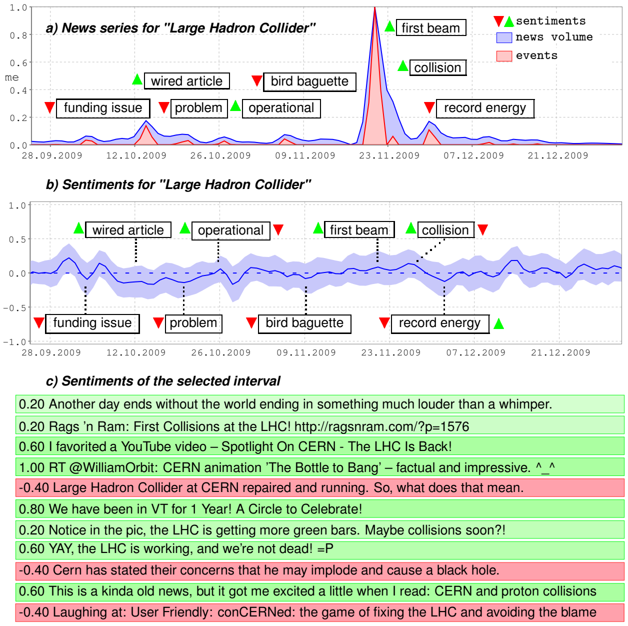
Correlating News and Sentiments.

We observe that sentiment and news time series require special correlation methods, that are different to conventional Pearson cross-correlation coefficient, which measures the linear dependency between variables. Such time series do not have a definite average level, around which the movement is happening. Instead, their values are outbursting from the minimum level at particular points in time. Therefore, we apply binary similarity measures, for example *cosine similarity* or *Jaccard coefficient*, measuring the intersection between sentiment and event bursts. In addition to counting the number of overlapping bursts, we can apply their weighting, for example based on magnitude.

3. DEMONSTRATION SCENARIO

Our system is capable to detect sentiment shifts in multiple time series and correlate them with news events in real time. In this demonstration, we intend to show the main features of our system on the real dataset from Twitter, by applying `NIA` on the stored data flow and giving users a possibility to visualize and explore news events, along with their sentiment changes, automatically extracted in real time. The important feature of our system is that it assigns sentiment changes to events on the same topic automatically based on their correlation, and also allows user to explore and suggest events from other related topics.

Demonstration Dataset. For our demonstration dataset, we selected 30 trending topics from Twitter, which featured the most prominent events for the period of half a year, from June 2009 till December 2009. The dataset contains approximately 7 million tweets in total and over 400 peaks during the events. We use 1-day aggregation for the time series of tweets volume and sentiments.


a) News series for "Large Hadron Collider"

b) Sentiments for "Large Hadron Collider"

c) Sentiments of the selected interval

0.20 Another day ends without the world ending in something much louder than a whimper.
0.20 Rags ’n Ram: First Collisions at the LHC! http://ragsnram.com/?p=1576
0.60 I favorited a YouTube video – Spotlight On CERN - The LHC Is Back!
1.00 RT @WilliamOrbit: CERN animation ’The Bottle to Bang’ – factual and impressive. ^_^
-0.40 Large Hadron Collider at CERN repaired and running. So, what does that mean.
0.80 We have been in VT for 1 Year! A Circle to Celebrate!
0.20 Notice in the pic, the LHC is getting more green bars. Maybe collisions soon?!
0.60 YAY, the LHC is working, and we’re not dead! =P
-0.40 Cern has stated their concerns that he may implode and cause a black hole.
0.60 This is a kinda old news, but it got me excited a little when I read: CERN and proton collisions
-0.40 Laughing at: User Friendly: conCERNed: the game of fixing the LHC and avoiding the blame

Figure 4: NIA demo workflow for the topic "LHC" from Twitter.

Demonstration Workflow. We intend to demonstrate an interactive application, shown in Figure 4, which allows users exploring news time series for each of the topics in our dataset, visualize the corresponding sentiments, drill down to the actual positive and negative posts, and see which other relevant news events could have affected sentiments, based on correlation analysis. Users can interact with the system by selecting and zooming time series, and also by adjusting various parameters, such as aggregation granularity, smoothing level and correlation thresholds, in real time.

Our demo starts by displaying to users a graph with the news volume, as seen in Figure 4(a). In this graph, `NIA` automatically extracts and annotates the relevant news events. Moreover, it marks the related sentiment shifts near text event annotations. The user can also visualize the entire time series of average sentiment, contradiction level, positive or negative sentiment volume, which in this case also become annotated with sentiment shifts and event labels, shown in Figure 4(b). In cases, when events cause transitions of sentiments (from positive to negative or vice verse), event annotations are marked with the two corresponding arrows, as seen in events marked as *“collision"* and *“record energy"*. Finally, by clicking on the interesting time interval, users are able to see a time series of posts, marked with positive (green) and negative (red) sentiment labels, as shown in Figure 4(c) for the event *“first beam"*.

4. CONCLUSIONS

Our system allows correlated analysis of sentiments and news, and raises new data analysis opportunities, useful for sociology and marketing researchers. Our evaluation reveals the existence of different parameters for various events, even for the same topic, all having different impacts on sentiments, suggesting that it is possible to predict sentiment changes. To achieve this, we need to take into account the type of response dynamics in addition to the event’s importance level, creating a more elaborate causality model. This task requires building a database of event and sentiment shift profiles, and exploration of events on related topics, in addition to events on the same topic, leading to the necessity for employing a sophisticated and interactive analytics platform, which helps users in their search for event causality. The purpose of our demo system is to facilitate the development of such a platform and explore possible ways of interactive news and sentiment analysis.

5. REFERENCES

[1] S. Asur, B. A. Huberman, G. Szabó, and C. Wang. Trends in social media: Persistence and decay. In *ICWSM*, 2011.

[2] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD*, 2009.

[3] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: Model and implications. In *KDD*, pages 6–14, 2012.

[4] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment in twitter events. *JASIST*, 62(2):406–418, 2011.

[5] M. Tsytsarau, S. Amer-Yahia, and T. Palpanas. Efficient sentiment correlation for large-scale demographics. In *SIGMOD*, pages 253–264, 2013.

[6] M. Tsytsarau and T. Palpanas. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery, Special Issue on 10 Years of Mining the Web*, pages 1–37, 2011.

[7] M. Tsytsarau, T. Palpanas, and M. Castellanos. Dynamics of news events and social media reaction. In *KDD*, 2014.

[8] M. Tsytsarau, T. Palpanas, and K. Denecke. Scalable detection of sentiment-based contradictions. In *DiversiWeb*, 2011.

Author Index