Discovering Reliable Dependencies from Data: Hardness and Improved Algorithms[*]

Panagiotis Mandros[1] , **Mario Boley**[2] and **Jilles Vreeken**[3]
[1]Max Planck Institute for Informatics, Saarbrücken, Germany
[2]Monash University, Melbourne, Australia
[3]CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
pmandros@mpi-inf.mpg.de, mario.boley@monash.edu, jv@cispa.saarland

Abstract

The reliable fraction of information is an attractive score for quantifying (functional) dependencies in high-dimensional data. In this paper, we systematically explore the algorithmic implications of using this measure for optimization. We show that the problem is NP-hard, justifying worst-case exponential-time as well as heuristic search methods. We then substantially improve the practical performance for both optimization styles by deriving a novel admissible bounding function that has an unbounded potential for additional pruning over the previously proposed one. Finally, we empirically investigate the approximation ratio of the greedy algorithm and show that it produces highly competitive results in a fraction of time needed for complete branch-and-bound style search.

1 Introduction

Given a data sample $\mathbf{D}_n = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ drawn from the joint distribution p of some input variables $\mathcal{I}$ and an output variable Y, it is a fundamental problem in data analysis to find variable subsets $\mathcal{X} \subseteq \mathcal{I}$ that jointly influence or (approximately) determine Y. This **functional dependency discovery** problem, i.e., to find

$$\arg\max\{Q(\mathcal{X}; Y) : \mathcal{X} \subseteq \mathcal{I}\} \qquad (1)$$

for some real-valued measure Q that assesses the dependence of Y on $\mathcal{X}$, is a classic topic in the database community [Ramakrishnan and Gehrke, 2000, Ch. 15], but also has many other applications including feature selection [Song *et al.*, 2012] and knowledge discovery [Ziarko, 2002]. For instance, finding such dependencies can help identify compact sets of descriptors that capture the underlying structure and actuating mechanisms of complex scientific domains (e.g., [Ghiringhelli *et al.*, 2015; Ouyang *et al.*, 2017]).

For categoric input and output variables, the measure Q can be chosen to be the **fraction of information** [Cavallo and Pittarelli, 1987; Giannella and Robertson, 2004; Reimherr and Nicolae, 2013] defined as

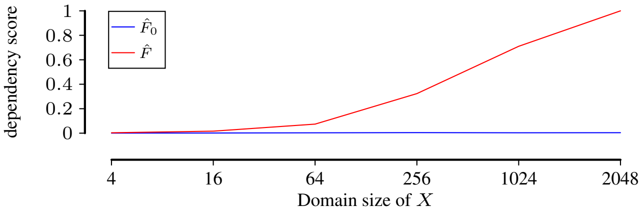


Figure 1: Dependency-by-chance. Estimated fraction of information for variables X of increasing domain size (4 to 2048) to independent Y (domain size 4) for fixed sample size (1000). Estimated dependency increases for naive estimator $\hat{F}$, while the corrected-for-chance estimator $\hat{F}_0$ accurately estimates population value $F(X; Y) = 0$.

$$F(\mathcal{X}; Y) = (H(Y) - H(Y \mid \mathcal{X}))/H(Y) \ ,$$

where $H(Y) = \sum_{y \in Y} p(y) \log p(y)$ denotes the **Shannon entropy**. This score represents the relative reduction of uncertainty about Y given $\mathcal{X}$. It takes on values between 0 and 1 corresponding to independence and exact functional dependency, respectively.

Estimating the score naively with empirical probabilities $\hat{p}$, however, leads to an overestimation of the actual dependence between $\mathcal{X}$ and Y, a behavior known as *dependency-by-chance* [Romano *et al.*, 2016]. In particular, since the bias is increasing with the domain size of variables [Roulston, 1999], it is unsuitable for dependence discovery where we have to soundly compare different variable sets of varying dimensionality and consequently of widely varying domain sizes (see Fig. 1). In some feature selection approaches (see, e.g., [Guyon and Elisseeff, 2003]) this problem is mitigated by only considering univariate and pairwise dependencies. Alternatively, some algorithms from the database literature, e.g., [Huhtala *et al.*, 1999; Kruse and Naumann, 2018], neglect this issue by assuming a closed-world, i.e., the unknown data generation process p is considered equal to the empirical $\hat{p}$ [Giannella and Robertson, 2004].

Both of these approaches are infeasible in the statistical setting with arbitrary sized variable sets that we are interested in. Instead, here, the fraction of information can be corrected by subtracting its estimated expected value under the hypothesis of independence. This gives rise to

[*]This is an extended abstract of work published as [Mandros *et al.*, 2018], that won the best paper award at the 2018 IEEE International Conference on Data Mining.

the **reliable fraction of information** [Mandros *et al.*, 2017; Vinh *et al.*, 2010] defined as

$$\hat{F}_0(\mathcal{X};Y) = \hat{F}(\mathcal{X};Y) - \hat{E}_0(\hat{F}(\mathcal{X};Y)) \ ,$$

where $\hat{E}_0(\hat{F}(\mathcal{X};Y)) = \sum_{\sigma \in S_n} \hat{F}(X;Y_\sigma)/n!$ is the expected value of $\hat{F}$ under the **permutation model** [Lancaster, 1969, p. 214], i.e., under the operation of permuting the empirical Y samples with a random permutation $\sigma \in S_n$. This estimator can be computed efficiently in $O(nk)$ for $\mathcal{X}$ with domain size k. Moreover, the maximization problem Eq. (1) can be solved effectively by a branch-and-bound scheme: the maximally attainable $\hat{F}_0$ for supersets of some solution $\mathcal{X}$ can be bounded by the function $\bar{f}_{\text{mon}}(\mathcal{X}) = 1 - \hat{E}_0(\hat{F}(\mathcal{X};Y))$, derived from the *monotonicity* of $\hat{E}_0(\hat{F}(\,\cdot\,;Y))$ [Mandros *et al.*, 2017].

This, however, is a rather simplistic bounding function that leaves room for substantial improvements. Moreover, it is unclear whether one has to rely on exponential-time worst-case branch-and-bound algorithms in the first place. Finally, the option of heuristic optimization has not yet been explored.

To this end, this paper provides the following contributions:

1. We show that the problem of maximizing the reliable fraction of information is NP-hard.
2. We then greatly improve the practical performance for both optimization styles by deriving a novel bounding function $\bar{f}_{\text{spc}}(\mathcal{X})$, which has an unbounded potential for additional pruning over the previously proposed one.
3. Finally, we report extensive empirical results evaluating the proposed bounding function and algorithms.

2 Reliable Dependency Discovery

Let us denote by $[n]$ the set of positive integers up to n. We assume a set of discrete random variables $\mathcal{A} = \mathcal{I} \cup \{Y\}$ is given along with an empirical sample $\mathbf{D}_n = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ of their joint distribution. For a variable X we denote its domain, called **categories** (or distinct values), by $V(X)$ but we also write $x \in X$ instead of $x \in V(X)$ whenever clear from the context. We identify a random variable X with the **labeling** $X\colon [n] \to V(X)$ it induces on the data sample, i.e., $X(i) = \mathbf{d}_i(X)$. Moreover, for a set $\mathcal{S} = \{S_1, \ldots, S_l\}$ of labelings over $[n]$, we define the corresponding vector-valued labeling by $\mathcal{S}(i) = (S_1(i), \ldots, S_l(i))$. With $X_\mathcal{Q}$ for a subset $\mathcal{Q} \subseteq [n]$, we denote the map X restricted to domain $\mathcal{Q}$.

We define $c_X : V(X) \to \mathbb{Z}_+$ to be the **empirical counts** of X, i.e., $c_X(x) = |\{i \in [n] : X(i) = x\}|$. We further denote with $\hat{p}_X : V(X) \to [0,1]$, where $\hat{p}_X(x) = c_X(x)/n$, the **empirical distribution** of X. Given another random variable Z, $\hat{p}_{Z\,|\,X=x} : V(Z) \to [0,1]$ is the **empirical conditional distribution** of Z given $X = x$, with $\hat{p}_{Z|X=x}(z) = c_{X \cup Z}(x,z)/c_X(x)$ for $z \in Z$. However, we use $\hat{p}(x)$ and $\hat{p}(z\,|\,x)$ respectively whenever clear from the context. These empirical probabilities give rise to the **empirical conditional entropy** $\hat{H}(Y\,|\,X) = \sum_{x \in X} \hat{p}(x)\hat{H}(Y\,|\,X = x)$, the **empirical mutual information** $\hat{I}(X;Y) = \hat{H}(Y) - \hat{H}(Y\,|\,X)$, and the **empirical fraction of information** $\hat{F}(X;Y) = \hat{I}(X;Y)/\hat{H}(Y)$. We abbreviate the **correction term** $\hat{E}_0(\hat{F}(X;Y))$ as $\hat{b}_0(X,Y,n)$ and the unnormalized version as $\hat{m}_0(X,Y,n) = \hat{b}_0(X,Y,n)\hat{H}(Y)$.

2.1 Specializations and Labeling Homomorphisms

Since we identified sets of random variables with their corresponding sample-index-to-value map, they are subject to the following general relations of maps with common domains.

Definition 1. *Let A and B be maps defined on a common domain N. We say that A is **equivalent** to B, denoted as $A \equiv B$, if for all $i, j \in N$ it holds that $A(i) = A(j)$ if and only if $B(i) = B(j)$. We say that B is a **specialization** of A, denoted as $A \preceq B$, if for all $i, j \in N$ with $A(i) \neq A(j)$ it holds that $B(i) \neq B(j)$.*

A special case of specializations is given by the subset relation of variable sets, e.g., if $\mathcal{X} \subseteq \mathcal{X}' \subseteq \mathcal{I}$ then $\mathcal{X} \preceq \mathcal{X}'$. The specialization relation implies some important properties for empirical probabilities and information-theoretic quantities.

Proposition 1. *Given variables X, Z and Y, with $X \preceq Z$, the following statements hold:*

a) *there is a projection $\pi : V(Z) \to V(X)$, s.t. for all $x \in V(X)$, it holds that $\hat{p}_X(x) = \sum_{z \in \pi^{-1}(x)} \hat{p}_Z(z)$,*

b) $\hat{H}(X) \leq \hat{H}(Z)$

c) $\hat{H}(Y\,|\,Z) \leq \hat{H}(Y\,|\,X)$,

d) $\hat{I}(X;Y) \leq \hat{I}(Z;Y)$,

In order to analyze monotonicity properties of the permutation model, the following additional definition is useful.

Definition 2. *We call a labeling X **homomorphic** to a labeling Z (w.r.t. the target variable Y), denoted as $X \precsim Z$, if there exists $\sigma \in S_n$ with $Y \equiv Y_\sigma$ such that $X \preceq Z_\sigma$.*

Importantly, the inequality of mutual information for specializations (Prop. 1d) carries over to homomorphic variables and in turn to their correction terms.

Proposition 2. *Given variables X, Z and Y, with $X \precsim Z$, the following statements hold:*

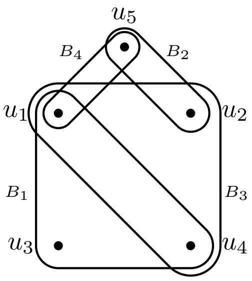
a) $\hat{I}(X;Y) \leq \hat{I}(Z;Y)$

b) $\hat{m}_o(X,Y,n) \leq \hat{m}_o(Z,Y,n)$

2.2 Search Algorithms

Effective algorithms for maximizing the reliable fraction of information over all subsets $\mathcal{X} \subseteq \mathcal{I}$ are enabled by the concept of bounding functions. A function $\bar{f}$ is called an **admissible bounding function** for an optimization function f if for all candidate solutions $\mathcal{X} \subseteq \mathcal{I}$, it holds that $\bar{f}(\mathcal{X}) \geq f(\mathcal{X}')$ for all $\mathcal{X}'$ with $\mathcal{X} \subseteq \mathcal{X}' \subseteq \mathcal{I}$. Such functions allow to prune all supersets $\mathcal{X}'$ of $\mathcal{X}$ whenever $\bar{f}(\mathcal{X}) \leq f(\mathcal{X}^*)$ for the current best solution $\mathcal{X}^*$ found during the optimization process.

Branch-and-bound, as the name suggests, combines this concept with a branching scheme that completely (and non-redundantly) enumerates the search space $2^{\mathcal{I}}$. In addition, branch-and-bound can trade optimality for efficiency with parameter $\alpha \in (0,1]$ governing the desired approximation guarantee. Here, we consider **optimized pruning for unordered search** (**Opus**), an advanced variant of



		$\mathbf{X_1}$	$\mathbf{X_2}$	X_3	X_4	Y
S_1	1	**1**	**a**	1	1	a
	2	**a**	**2**	2	a	a
	3	**3**	**a**	a	a	a
	4	**4**	**a**	4	a	a
	5	**a**	**5**	a	5	a
S_2	6	**a**	**a**	a	a	b
	7	**a**	**a**	a	a	b
	8	**a**	**a**	a	a	b
	9	**a**	**a**	a	a	b
	10	**a**	**a**	a	a	b
S_3	11	**b**	**c**	c	c	c
	12	**c**	**b**	c	c	c
	13	**c**	**c**	b	c	c
	14	**c**	**c**	c	b	c
	15	**c**	**c**	c	c	c

Figure 2: Base transformation example. A set cover instance $U = \{u_1, \dots, u_5\}$ and $\mathcal{B} = \{B_1, B_2, B_3, B_4\}$ (left). The resulting D_{15} using $\tau_1(U, \mathcal{B})$ (right) (bold indicates the set cover)

branch-and-bound that effectively propagates pruning information to siblings in the search tree [Webb, 1995]. A commonly used alternative to complete branch-and-bound search for the optimization of dependency measures is the standard **greedy algorithm** (see [Guyon and Elisseeff, 2003; Brown *et al.*, 2012]). This algorithm only refines the best candidate in a given iteration.

3 Hardness of Optimization

In this section, we show that the problem of maximizing $\hat{F}_0$ is NP-hard by providing a reduction from the well-known NP-hard **minimum set cover** problem: given a finite universe $U = \{u_1, \dots, u_n\}$ and collection of subsets $\mathcal{B} = \{B_1, \dots, B_m\} \subseteq 2^U$, find a set cover, i.e., a sub-collection $\mathcal{C} \subseteq \mathcal{B}$ with $\bigcup \mathcal{C} = U$, that is of minimal cardinality.

The reduction consists of two parts. First, we construct a base transformation $\tau_1(U, \mathcal{B}) = \mathbf{D}_l$ that maps a set cover instance to a dataset $\mathbf{D}_l$ such that set covers correspond to attribute sets with an empirical fraction of information score $\hat{F}$ of 1 and bias correction terms $\hat{b}_0$ that are a monotonically decreasing function of their cardinality (see Fig.2). In a second step, we calibrate the $\hat{b}_0$ terms such that, when considering the corrected score $\hat{F}_0$, they cannot change the order between attribute sets with different $\hat{F}$ values but only act as a tie-breaker between attribute sets of equal $\hat{F}$ value. This is achieved by copying the dataset $\mathbf{D}_l$ a suitable number of times k such that the correction terms are sufficiently small but the overall transformation, denoted $\tau_k(U, \mathcal{B}) = \mathbf{D}_{kl}$, is still polynomial.

4 Refined Bounding Function

The NP-hardness established in the previous section excludes (unless P=NP) the existence of a polynomial time algorithm for maximizing the reliable fraction of information, leaving therefore exact but exponential search and heuristics as the two options. For both, and particularly the former, reducing the search space can lead to more effective algorithms. For this purpose, we derive in this section a novel bounding function for $\hat{F}_0$ to be used for pruning. The ideal function would be

$$\bar{f}_{\text{ideal}}(\mathcal{X}) = \max\{\hat{F}_0(\mathcal{X}'; Y) \colon \mathcal{X} \subseteq \mathcal{X}' \subseteq \mathcal{I}\} \ .$$

Computing this function is equivalent to the original optimization problem and hence NP-hard. We can relax the maximum over all supersets to the maximum over all *specializations* of $\mathcal{X}$. That is, we define a bounding function $\bar{f}_{\text{spc}}(\mathcal{X})$ through

$$\begin{aligned}\bar{f}_{\text{spc}}(\mathcal{X}) &= \max\{\hat{F}_0(\mathcal{X}'; Y) \colon \mathcal{X} \preceq \mathcal{X}'\} \\ &\geq \max\{\hat{F}_0(\mathcal{X}'; Y) \colon \mathcal{X} \subseteq \mathcal{X}' \subseteq \mathcal{I}\} = \bar{f}_{\text{ideal}}(\mathcal{X}) \ .\end{aligned}$$

While this definition constitutes an admissible bounding function, it is unclear how it can be efficiently evaluated. Let us denote by R^+ the operation of joining a labeling R with the target attribute Y, i.e., $R^+ = \{R\} \cup \{Y\}$. This definition gives rise to a simple constructive form for computing $\bar{f}_{\text{spc}}$.

Theorem 3. *The function $\bar{f}_{spc}$ can be efficiently computed as $\bar{f}_{spc}(\mathcal{X}) = \hat{F}_0(\mathcal{X}^+; Y)$ in time $O(n|V(\mathcal{X})||V(Y)|)$.*

Intuitively, $\mathcal{X}^+$ constitutes the most efficient specialization of $\mathcal{X}$ in terms of growth in $\hat{F}$ and $\hat{b}_0$. In contrast, the bounding function $\bar{f}_{\text{mon}}(\mathcal{X}) = 1 - \hat{b}_0(\mathcal{X}, Y, n)$ of [Mandros *et al.*, 2017] assumes that full information about the target can be attained (i.e., $\hat{F} = 1$) without "paying" an increased $\hat{b}_0$ term. The following proposition shows this idea leads to an inferior bound.

Proposition 4. *Let $\mathcal{X} \subseteq \mathcal{I}$ and $\Delta = \bar{f}_{mon}(\mathcal{X}) - \bar{f}_{spc}(\mathcal{X})$. The following statements hold:*

a) $\Delta \geq 0$ for all datasets, i.e., $\bar{f}_{spc}(\mathcal{X}) \leq \bar{f}_{mon}(\mathcal{X})$

b) there are datasets $\mathbf{D}_{4l}$ for all $l \geq 1$ s.t. $\Delta \in \Omega(1 - \frac{1}{\log 2l})$

Thus, we have established that $\bar{f}_{\text{spc}}$ is not only tighter than $\bar{f}_{\text{mon}}$, but even that the difference can be arbitrary close to 1 (for an increasing domain size of Y). Put differently, their ratio, and thus the potential for additional pruning, is unbounded.

5 Evaluation

For ease of comparison to [Mandros *et al.*, 2017], we consider datasets from the KEEL data repository [Alcalà-Fdez *et al.*, 2011]. In particular, we use all classification datasets with $d \in [10, 90]$ and no missing values, resulting in 35 datasets with 52000 and 30 rows and columns on average, respectively. All implementations are available online[1].

We use two metrics for evaluation, the relative runtime difference and the relative difference in number of explored nodes. For methods A and B, the relative runtime difference on a particular dataset is computed as $\text{rrd}(A, B) = (\tau_A - \tau_B)/\max(\tau_A, \tau_B)$, where τ_A and τ_B are the run times for A and B respectively. The rrd score lies in $[-1, 1]$, where positive (negative) values indicate that B is proportionally faster (slower). For example, a rrd score of 0.5 corresponds to a factor of 2 speed-up, 0.66 to a factor of 3, 0.75 to 4 etc. The relative nodes explored difference rnd is defined similarly. For both scores, we consider $(-0.5, 0.5)$ to be a region of practical equivalence, i.e., a factor of 2 of improvement is required to consider a method "better".

[1] https://github.com/pmandros/fodiscovery

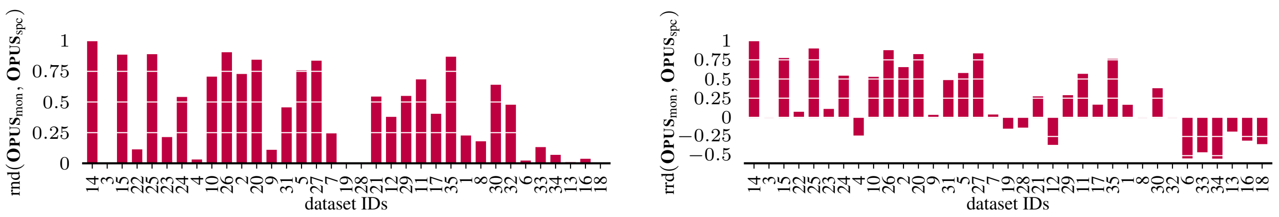


Figure 3: Evaluating $\bar{f}_{\text{spc}}$ for branch-and-bound optimization. Relative nodes explored difference (left) and relative runtime difference (right) between methods **OPUS**$_{\text{spc}}$ and **OPUS**$_{\text{mon}}$. Positive (negative) numbers indicate that **OPUS**$_{\text{spc}}$ (**OPUS**$_{\text{mon}}$) is proportionally "better". The datasets are sorted in decreasing number of attributes.

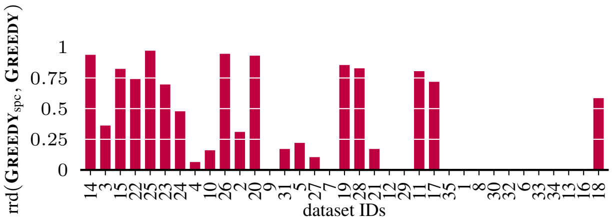


Figure 4: Evaluating $\bar{f}_{\text{spc}}$ for heuristic optimization. Relative time difference between methods **GREEDY**$_{\text{spc}}$ and **GREEDY**. Positive (negative) numbers indicate that **GREEDY**$_{\text{spc}}$ (**GREEDY**) is proportionally "better". Data are sorted in decreasing number of attributes.

5.1 Branch-and-bound

Here we investigate the effect of the refined bounding function by comparing **OPUS**$_{\text{spc}}$ and **OPUS**$_{\text{mon}}$. In Fig. 3 we present the comparison between **OPUS**$_{\text{spc}}$ and **OPUS**$_{\text{mon}}$. The left plot demonstrates that $\bar{f}_{\text{spc}}$ can lead to a considerable reduction of nodes explored over $\bar{f}_{\text{mon}}$. In particular, 15 cases have at least a factor of 2 reduction, 7 have 4, and there is one 1 with 760. For 20 cases there is no practical difference. The plot validates that the potential for additional pruning is indeed unbounded (Sec. 4). In terms of runtime efficiency (right plot), **OPUS**$_{\text{spc}}$ is "faster" in 70% of the datasets. In more detail, and considering practical improvements, 12 datasets have at least a factor of 2 speedup, 6 have 4, 1 has 266, while only 2 have a factor of 2 slowdown. Moreover, we observe from the plot (where datasets are sorted in decreasing number of attributes) a clear correlation between number of attributes and efficiency: the 6 out of 10 datasets with the slowdown are also the ones with the lowest number of features. Overall, $\bar{f}_{\text{spc}}$ leads to a more effective optimization with branch-and-bound, and particularly for the higher-dimensional cases.

5.2 Greedy

We begin the evaluation with the performance of $\bar{f}_{\text{spc}}$ for heuristic search. We present the relative runtime differences of **GREEDY** and **GREEDY**$_{\text{spc}}$ in Fig. 4. The plot shows that $\bar{f}_{\text{spc}}$ indeed improves the efficiency of the heuristic search, as we find that for 12 datasets there is a speedup of at least a factor of 2, and 8 of at least a factor of 4.

Next, we investigate the quality of the greedy results. In Fig. 5 we plot the differences between the $\hat{F}_0$ score of the results obtained by greedy and branch-and-bound on each dataset. Note that branch-and-bound uses the same α values as with the experiments in Sec 5.1, and that we only plot the non-zero differences in the two plots, left for $\alpha = 1$, i.e, optimal solutions, and right for $\alpha < 1$, i.e., approximate solutions with guarantees.

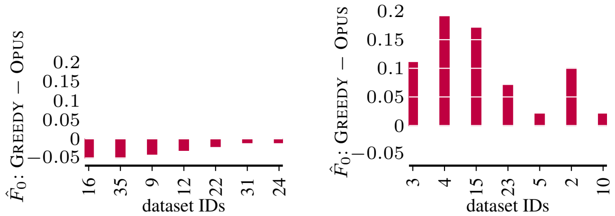


Figure 5: Evaluating the heuristic algorithm for result quality. Left: difference in $\hat{F}_0$ between methods **GREEDY**$_{\text{spc}}$ and **OPUS**$_{\text{spc}}$ (i.e., $\hat{F}_0(\mathcal{X}^*_{grd}; Y) - \hat{F}_0(\mathcal{X}^*_{bnb}; Y)$) for $\alpha = 1$. Negative values close to 0 indicate GREEDY retrieves nearly optimal solutions. Data are sorted in increasing quality difference. Right: difference for $\alpha < 1$. Positive values indicate that GREEDY retrieves better solutions when OPUS uses guarantees $\alpha < 1$. Data are sorted in increasing α values.

At a first glance, we observe that there is no difference in 21 out of 35 cases considered, 7 where greedy is better (this of course on the datasets where $\alpha < 1$), and 7 for branch-and-bound. Out of the 21 cases where the two algorithms have equal $\hat{F}_0$, 16 of them have $\alpha = 1$, i.e., the greedy algorithm is optimal roughly 45% of the time. Moreover, the cases where branch-and-bound is better is only by a small margin, 0.03 on average, while greedy "wins" by 0.1 on average. Another observation from the right plot of Fig. 5 is that the largest differences between the two algorithms is for the 3 datasets where the lowest α values where used, i.e., 0.05, 0.1, and 0.35.

6 Conclusion

We investigated the algorithmic aspect of discovering dependencies in data using the reliable fraction of information, where we proved the NP-hardness of the problem and derived a refined bounding function for more effective optimization. Moreover, we considered an improved branch-and-bound algorithm and explored the aspects of heuristic optimization. The experimental evaluation showed that the refined bounding function is very effective for both types of optimization, and that the greedy algorithm provides nearly optimal results.

References

[Alcalà-Fdez *et al.*, 2011] Jesús Alcalà-Fdez, Alberto Fernàndez, Juliàn Luengo, Joaquìn Derrac, and Salvador Garcìa. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287, 2011.

[Brown *et al.*, 2012] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *J. Mach. Learn. Res.*, 13:27–66, January 2012.

[Cavallo and Pittarelli, 1987] Roger Cavallo and Michael Pittarelli. The theory of probabilistic databases. In *Proceedings of the 13th International Conference on Very Large Data Bases (VLDB), Brighton, UK*, pages 71–81, 1987.

[Ghiringhelli *et al.*, 2015] Luca M Ghiringhelli, Jan Vybiral, Sergey V Levchenko, Claudia Draxl, and Matthias Scheffler. Big data of materials science: Critical role of the descriptor. *Physical review letters*, 114(10):105503, 2015.

[Giannella and Robertson, 2004] Chris Giannella and Edward L. Robertson. On approximation measures for functional dependencies. *Information Systems*, 29(6):483–507, 2004.

[Guyon and Elisseeff, 2003] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[Huhtala *et al.*, 1999] Ykä Huhtala, Juha Kärkkäinen, Pasi Porkka, and Hannu Toivonen. Tane: An efficient algorithm for discovering functional and approximate dependencies. *The Computer Journal*, 42(2):100–111, 1999.

[Kruse and Naumann, 2018] Sebastian Kruse and Felix Naumann. Efficient discovery of approximate dependencies. *Proc. VLDB Endow.*, 11(7):759–772, March 2018.

[Lancaster, 1969] H.O. Lancaster. *The chi-squared distribution*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley, 1969.

[Mandros *et al.*, 2017] Panagiotis Mandros, Mario Boley, and Jilles Vreeken. Discovering reliable approximate functional dependencies. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17. ACM, 2017.

[Mandros *et al.*, 2018] Panagiotis Mandros, Mario Boley, and Jilles Vreeken. Discovering reliable dependencies from data: Hardness and improved algorithms. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 317–326. IEEE, 2018.

[Ouyang *et al.*, 2017] Runhai Ouyang, Stefano Curtarolo, Emre Ahmetcik, Matthias Scheffler, and Luca Ghiringhelli. Sisso: a compressed-sensing method for systematically identifying efficient physical models of materials properties. (1710.03319), 2017.

[Ramakrishnan and Gehrke, 2000] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*. McGraw-Hill Higher Education, 2nd edition, 2000.

[Reimherr and Nicolae, 2013] Matthew Reimherr and Dan L Nicolae. On quantifying dependence: A framework for developing interpretable measures. *Statistical Science*, 28(1):116–130, 2013.

[Romano *et al.*, 2016] Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. A framework to adjust dependency measure estimates for chance. In *Proceedings of the SIAM International Conference on Data Mining (SDM), Miami, FL*. SIAM, 2016.

[Roulston, 1999] Mark S Roulston. Estimating the errors on measured entropy and mutual information. *Physica D: Nonlinear Phenomena*, 125(3):285–294, 1999.

[Song *et al.*, 2012] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *JMLR*, 13:1393–1434, 2012.

[Vinh *et al.*, 2010] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

[Webb, 1995] G. I. Webb. Opus: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research*, 3:431–465, 1995.

[Ziarko, 2002] Wojciech Ziarko. Rough set approaches for discovery of rules and attribute dependencies. *Handbook of data mining and knowledge discovery*, pages 328–338, 2002.