Knowledge and Information Systems manuscript No.
(will be inserted by the editor)

Discovering Dependencies with Reliable Mutual Information

Panagiotis Mandros · Mario Boley · Jilles Vreeken

Received: date / Accepted: date

Abstract Given a database and a target attribute of interest, how can we tell whether there exists a functional, or approximately functional dependency of the target on any other set of attributes in the data? How can we reliably, without bias to sample size or dimensionality, measure the strength of such a dependency? And, how can we efficiently discover the exact or approximate top-k dependencies? These are the questions we answer in this work.

To meaningfully measure the degree of dependency, we adopt an information-theoretic approach and propose a non-parametric and reliable estimator for mutual information that is well-suited for optimization in high-dimensional data. We then 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 derive two bounding functions for the proposed estimator, enabling for the first time the discovery of functional dependencies from data with guarantees of optimality. Empirical evaluation shows that the estimator has desirable statistical properties, the bounding functions lead to effective search algorithms, and

The research described in this paper builds upon and extends our work appearing in KDD'17 [8] and ICDM'18 [9].

Panagiotis Mandros (✉)
Max Planck Institute for Informatics and Saarland University
Saarland Informatics Campus, Saarbrücken, Germany
E-mail: pmandros@mpi-inf.mpg.de

Mario Boley
Monash University
Melbourne, Australia
E-mail: mario.boley@monash.edu

Jilles Vreeken
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
E-mail: jv@cispa.saarland

when combined, qualitative experiments show that the functional dependency framework indeed discovers meaningful dependencies.

Keywords information theory · knowledge discovery · approximate functional dependency · pattern mining · algorithms · branch-and-bound

1 Introduction

Given data over some input attributes $\mathcal{I} = \{X_1, \ldots, X_d\}$ and target attribute Y, finding those subsets $\mathcal{X} \subseteq \mathcal{I}$ that jointly (approximately) determine Y is a fundamental problem in knowledge discovery. Dependencies like these are essential for a variety of applications, including in many scientific domains where scientists are often concerned with finding compact sets of descriptors that capture the underlying process of phenomena they study [1; 2].

The **functional dependency discovery** problem can be formulated as finding the top-k subsets $\mathcal{X}_1^*, \ldots, \mathcal{X}_k^* \subseteq \mathcal{I}$ with

$$Q(\mathcal{X}_i^*; Y) = \max\{Q(\mathcal{X}; Y) \colon\ Q(\mathcal{X}_{i-1}^*; Y) \geq Q(\mathcal{X}; Y), \mathcal{X} \subseteq \mathcal{I}\}\ , \tag{1}$$

where Q is some real-valued measure that quantifies the degree of dependency of Y on $\mathcal{X}$. For knowledge discovery, it is essential that Q is both an interpretable and statistically robust measure, such that the analyst can understand the results and trust them to represent aspects of the underlying process. Moreover, solutions to Eq. (1) need to be exact, or at the very least come with approximation guarantees, as only with such guarantees we can verify the absence of dependencies in data.

For categorical input and output variables, the ideal choice for Q is the information-theoretic measure **fraction of information** [3; 4; 5]. It is defined as

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

where $H(Y) = -\sum_{y \in Y} p(y) \log(p(y))$ denotes the **Shannon entropy** and $H(Y \mid \mathcal{X}) = \sum_{\mathbf{x} \in \mathcal{X}} p(\mathbf{x}) H(Y \mid \mathcal{X} = \mathbf{x})$ the **conditional Shannon entropy**. The numerator is the well-known **mutual information** $I(\mathcal{X}; Y) = H(Y) - H(Y \mid \mathcal{X})$. The entropy measures the uncertainty about Y, while the conditional entropy measures the uncertainty after observing $\mathcal{X}$. The fraction of information then represents the proportional reduction of uncertainty about Y by knowing $\mathcal{X}$. It takes values between 0 and 1, and because these extremes have clear meaning, scores are easily interpretable; $F(\mathcal{X}; Y) = 0$ corresponds to statistical independence, whereas $F(\mathcal{X}; Y) = 1$ corresponds to functional dependency.

Estimating mutual information given empirical samples, however, is a non-trivial task. The standard plug-in estimator $\hat{I}$ defined using the empirical probability induced by the data samples tends to overestimate the actual dependency between $\mathcal{X}$ and Y—a behavior known as **dependency-by-chance** [6]. In the most extreme case, it can indicate a functional dependency even when $\mathcal{X}$ and Y are actually independent in the population (see Figure 1 for an example). Since the bias of $\hat{I}$ is a function of the domain sizes of variables [7], the empirical estimator is fully unsuited for use in dependency discovery: during search we have to soundly compare variable sets of varying dimensionality, and consequently of widely varying domain sizes. Moreover, there are to the best of our knowledge no efficient algorithms solving Eq. (1) with mutual information that provide exact and approximate results.

To obtain a statistically reliable estimator for high-dimensional mutual information, we propose a correction to the plug-in $\hat{I}$ by subtracting its expected value $\mathbb{E}_0[\hat{I}(\mathcal{X};Y)]$ under a suitable null hypothesis model. We choose the non-parametric **permutation model** [10], resulting in an expected value computed as the average over all possible sample permutations. The resulting estimator $\hat{I}_0(\mathcal{X};Y) = \hat{I}(\mathcal{X};Y) - \mathbb{E}_0[\hat{I}(\mathcal{X};Y)]$, which we term the **reliable mutual information**, not only accounts for dependency-by-chance, but is also efficiently computed. For the discovery part, we show that maximizing Eq. (1) with $\hat{I}_0$ is NP-hard. To enable efficient exact, approximate, and heuristic algorithms, we derive two bounding functions for $\hat{I}_0$ that can be used with branch-and-bound and heuristic search to greatly prune the search space.

In this article we build upon and extend our recent work published as Mandros et al. [8; 9]. In the first of these two papers we introduced the general problem, a corrected estimator, and a bounding function that allows branch-and-bound search for the strongest dependencies [8]. In the second paper, we focused more on the discovery aspect, proved NP-hardness and proposed a tighter bounding function, and investigated better algorithms for both exact and heuristic search [9]. In this paper, we present these results in a unified way that allows for more detail and insight in both the problem and proposed solutions. In particular, we provide a more comprehensive derivation of our estimator and formally make the link to non-parametric permutation tests. We prove that in our setting mutual information is not submodular, which excludes established approximation guarantees for submodular functions and heuristic optimization. Last, we provide extensive evaluation of our estimators to the state of the art, including one that is based on parametric statistics, and by performing bias, variance, and precision/recall experiments.

Overall, our contributions are the following. We derive a consistent and reliable estimator for mutual information that alleviates the dependency-by-chance problems arising from data sparsity (Section 2). This estimator is based on non-parametric statistics, and hence is well-suited for exploratory tasks. We accompany the estimator with a set of useful properties that can be used to develop efficient search algorithms. We then study the algorithmic aspects, and in particular, show that maximizing the reliable estimator is NP-hard (Section 3), and derive two effective bounding functions that search algorithms can use to greatly prune the search space (Section 4). To discover the top functional dependencies, we propose a branch-and-bound algorithm that comes with approximation guarantees, and in addition, a fast heuristic algorithm (Section 5). Last but not least, we perform an extensive evaluation for both the estimator and algorithms (Section 6). Although we formally show that the problem is not submodular, in practice the greedy algorithm performs

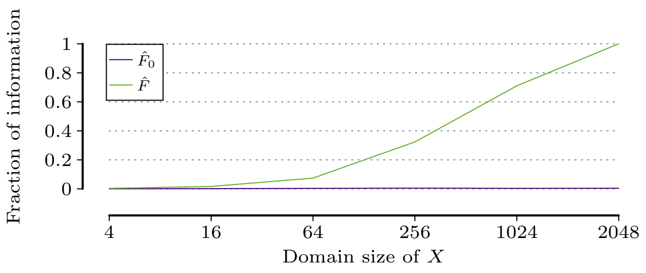


Fig. 1: **Dependency-by-chance.** Estimated fraction of information $F(X;Y)$ for Y independent of X. Variables X have increasing domain size (4 to 2048), and we consider a fixed sample size (1000). Estimated dependency increases for plug-in estimator $\hat{F}$, while the corrected-for-chance estimator $\hat{F}_0$ accurately estimates population value $F(X;Y) = 0$.

surprisingly well; it is almost always able to either retrieve the exact solution, or returns one that in terms of quality is extremely close to the optimum.

2 Reliable mutual information and properties

In this section we derive an estimator for mutual estimator that is well-suited for functional dependency discovery, and well as provide properties to be used for optimization. We start with preliminaries and notation.

Let us denote by $[n]$ the set of positive integers up to n. The symbols log and ln refer to the logarithms of base 2 and e, respectively. We assume a set of discrete random variables $\mathcal{I} = \{X_1, \ldots, X_d\}$ and Y is given along with an empirical sample $\mathbf{D}_n = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ of their joint distribution p. 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 $c_X\colon V(X) \to \mathbb{Z}_+$ 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, $\hat{p}_{Z \mid X=x}\colon V(Z) \to [0,1]$ 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 $\hat{p}(z \mid x)$ respectively whenever clear from the context. These empirical probabilities give rise to the **empirical conditional entropy** $\hat{H}(Y \mid X) = \sum_{x \in X} \hat{p}(x)\hat{H}(Y \mid X = x)$, the **empirical mutual information** $\hat{I}(X;Y) = \hat{H}(Y) - \hat{H}(Y \mid X)$, and the **empirical fraction of information** $\hat{F}(X;Y) = \hat{I}(X;Y)/\hat{H}(Y)$. These estimators are also known as **plug-in** estimators.

2.1 Reliable mutual information

Intuitively, the reason why $\hat{I}$ is unreliable as an estimator for I is that it does not take into account the confidence in the empirical estimates $\hat{H}(Y|\mathcal{X} = \mathbf{x})$ for subsets $\mathcal{X} \subseteq \mathcal{I}$. This is particularly profound for the extreme case where the empirical count $c_{\mathcal{X}}(\mathbf{x})$ is equal to 1. In this situation $c_{\mathcal{X} \cup Y}(\mathbf{x}, y) = 1$ exactly for one value of $y \in V(Y)$ and, hence, $\hat{H}(Y|\mathcal{X} = \mathbf{x})$ is trivially equal to 0 independent of the true distribution p. This case is likely to occur for many of the sampled values for $\mathcal{X}$ if the data size n is small compared to the observed domain of $\mathcal{X}$—even when $I(\mathcal{X}; Y) = 0$, which coincides with the highest error, because then $H(Y|\mathcal{X} = \mathbf{x}) = H(Y)$ while $\hat{H}(Y|\mathcal{X} = \mathbf{x}) = 0$.

The tendency for the plug-in estimator $\hat{I}$ to overestimate is more formally explained by the bias result of Roulston [7], where

$$\text{bias}(\hat{I}(\mathcal{X}; Y)) = \frac{|V(\mathcal{X} \cup \{Y\})| - |V(\mathcal{X})| - |V(Y)| + 1}{2n} \ .$$

We see that the bias is independent of the actual distribution p, and it depends solely on the observed domain sizes and the number of samples n. The bias is high when $\mathcal{X}$ and Y combined produce a large domain compared to their marginal domains and sample size n, and is at the highest when $\mathcal{X}$ and Y are independent in the underlying distribution p, i.e., when $I(\mathcal{X}; Y) = 0$.

These observations suggest a correction for the empirical estimator $\hat{I}$ by subtracting its expected value under a suitable null hypothesis model that takes into account data sparsity. A non-parametric choice for this model is the **permutation model** [10, p. 214], arriving at the expected value

$$\mathbb{E}_0[\hat{I}(\mathcal{X}; Y)] = \frac{1}{n!} \sum_{\sigma \in S_n} \hat{I}(X; Y_\sigma) \ , \tag{2}$$

where S_n denotes the **symmetric group** of $[n]$, i.e., the set of bijections from $[n]$ to $[n]$, and Y_σ denotes the composition of map Y with the permutation $\sigma \in S_n$, i.e., $Y_\sigma(\cdot) = Y(\sigma(\cdot))$. Essentially, Eq. (2) is the average empirical mutual information over all possible sample permutations with fixed marginal counts. With this, the **reliable mutual information** is defined as

$$\hat{I}_0(\mathcal{X}; Y) = \hat{I}(\mathcal{X}; Y) - \mathbb{E}_0[\hat{I}(\mathcal{X}; Y)] \ ,$$

and the **reliable fraction of information** as

$$\hat{F}_0(\mathcal{X}; Y) = \hat{I}_0(\mathcal{X}; Y) / \hat{H}(Y) \ .$$

This correction is related to permutation tests, where instead of finding the exact probability of observing more extreme datasets, the average value of the score over all possible datasets is computed. Intuitively, when it appears in a sample that $\hat{F}(\mathcal{X}; Y)$ is high for a $\mathcal{X} \subseteq \mathcal{I}$ with a large domain, many of the permutations will also show high dependency, and hence the bias is corrected accordingly. From here on we use these quantities interchangeably since $\hat{H}(Y)$ is just a constant normalization, and we abbreviate the **correction**

terms $\mathbb{E}_0[\hat{I}(X;Y)]$ as $m_0(X,Y,n)$ and the normalized version as $b_0(X,Y,n) = \mathbb{E}_0[\hat{F}(X;Y)] = m_0(X,Y,n)/\hat{H}(Y)$.

Regarding the evaluation of Eq. (2), a naive approach with $n!$ possible permutations is computationally infeasible. However, Vinh et al. [11] show that the complexity is dramatically reduced by reformulating it as a function of contingency table cell values and exploiting symmetries. Let the observed domains of $\mathcal{X}$ and Y be $V(\mathcal{X}) = \{\mathbf{x}_1, \dots, \mathbf{x}_R\}$ and $V(Y) = \{y_1, \dots, y_C\}$, respectively. We define shortcuts for the observed marginal counts $a_i = c(\mathcal{X} = \mathbf{x}_i)$ and $b_j = c(Y = y_j)$ as well as for the joint counts $c_{i,j} = c(\mathcal{X} = \mathbf{x}_i, Y = y_j)$. The **contingency table c** for $\mathcal{X}$ and Y is then the complete joint count configuration $\mathbf{c} = \{c_{i,j} : 1 \leq i \leq R, 1 \leq j \leq C\}$. The empirical mutual information for $\mathcal{X}$ and Y can then be computed as

$$\hat{I}(\mathcal{X}, Y) = \hat{I}(\mathbf{c}) = \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{c_{ij}}{n} \log \frac{c_{ij} n}{a_i b_j} \ .$$

Each $\sigma \in S_n$ results in a contingency table $\mathbf{c}^\sigma$. We denote with $\mathcal{T} = \{\mathbf{c}^\sigma : \sigma \in S_n\}$ the set of all such contingency tables. Crucially, all these tables have the same marginal counts a_i, b_j, $i \in [1,R], j \in [1,C]$. Hence, we can rewrite

$$m_0(\mathcal{X}, Y, n) = \sum_{\mathbf{c}^\sigma \in \mathcal{T}} \hat{p}_0(\mathbf{c}^\sigma) \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{c_{ij}^\sigma}{n} \log \frac{c_{ij}^\sigma n}{a_i b_j} \ ,$$

where $\hat{p}_0(\mathbf{c})$ is the probability of contingency table $\mathbf{c} \in \mathcal{T}$. This allows us to re-order the terms to have a per-cell contribution to m_0, rather than per-contingency-table $\mathbf{c} \in \mathcal{T}$, i.e.,

$$m_0(\mathcal{X}, Y, n) = \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{k=0}^{n} \hat{p}_0(c_{ij}^\sigma = k) \frac{k}{n} \log \frac{kn}{a_i b_j} \ .$$

Under the permutation model, the empirical counts c_{ij}^σ are distributed hypergeometrically, i.e.,

$$\hat{p}_0(c_{ij}^\sigma = k) = \binom{b_i}{k} \binom{n - b_i}{a_j - k} \Big/ \binom{n}{a_j} \ .$$

These probabilities can be computed efficiently in an incremental manner using the support of the hypergeometric distribution, i.e., k is non-zero for $k \in [\max(0, a_i + b_j - n), \min(a_i, b_j)]$, and the hypergeometric recurrence formula

$$\hat{p}_0(k+1) = \hat{p}_0(k) \frac{(a_i - k)(b_j - k)}{(k+1)(n - a_i - b_j + k + 1)} \ .$$

The complexity for m_0 is then $O(n \max\{|V(\mathcal{X})|, |V(Y)|\})$ [12]. Moreover, the computation can be done in parallel for each individual cell.

In addition to being computationally efficient, the resulting reliable dependency score $\hat{F}_0(\mathcal{X};Y) = \hat{F}(\mathcal{X};Y) - b_0(\mathcal{X}, Y, n)$ satisfies several other properties.

First of all, it is indeed a consistent estimator of F. In particular, Vinh et al. [13] show that $\lim_{n\to\infty} \hat{m}_0(\mathcal{X}, Y, n) = 0$, and together with the consistency of the plug-in $\hat{F}$ [14], we have that $\lim_{n\to\infty} \hat{F}_0(\mathcal{X}; Y) = F(\mathcal{X}; Y)$. Moreover, $\hat{F}_0(\mathcal{X}; Y)$ remains upper-bounded by 1, although this value is only attainable in the limit case $n \to \infty$ (for true functional dependencies). Most importantly, contrary to the naive estimator, we have that $\hat{F}_0$ approaches zero[1] as the empirical domain $V(\mathcal{X})$ increases relative to the data size n. We show this by proving the monotonicity of m_0 with respect to the subset relation.

Theorem 1 *Given two sets of variables $\mathcal{X}, \mathcal{X}'$ with $\mathcal{X} \subseteq \mathcal{X}' \subseteq \mathcal{I}$, then $m_0(\mathcal{X}, Y, n) \leq m_0(\mathcal{X}', Y, n)$, i.e., the expected value under the permutation model is monotonically increasing with respect to the subset relation.*

Proof Using the chain rule of information and that mutual information is non-negative [15, Chapter 2], we have that $I(\mathcal{X}; Y) \leq I(\mathcal{X}'; Y)$. Then for each $\sigma \in S_n$ it holds that $I(\mathcal{X}; Y_\sigma) \leq I(\mathcal{X}'; Y_\sigma)$, and hence $\sum_{\sigma \in S_n} \hat{I}(\mathcal{X}; Y_\sigma) \leq \sum_{\sigma \in S_n} \hat{I}(\mathcal{X}'; Y_\sigma)$, which concludes the proof. $\square$

Theorem 1 states that $m_0(\mathcal{X}, Y, n)$ can indeed penalize spurious dependencies that appear with high dimensional $\mathcal{X} \subseteq \mathcal{I}$, justifying therefore the adjective *reliable* for the two estimators. In the following section, we couple the above information-theoretic quantities with relations for empirical attributes.

2.2 Specializations and Labeling Homomorphisms

Since we identify 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, Y, with $X \preceq Z$, the following statements hold:*

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

[1] In fact, it is principally not lower bounded by 0 since m_0 can be larger than $\hat{I}$. These cases strongly indicate independence.

b) $\hat{H}(X) \leq \hat{H}(Z)$
c) $\hat{H}(Y \mid Z) \leq \hat{H}(Y \mid X)$
d) $\hat{I}(X;Y) \leq \hat{I}(Z;Y)$

Proof Let us denote with p and q the $\hat{p}_{X \cup Y}$ and $\hat{p}_{Z \cup Y}$ distributions respectively. Statement a) follows from the definition. For b), we define $h(x) = -p(x) \log p(x)$ for $x \in X$, and similarly $h(z)$ for $z \in Z$. We show that for all $x \in X$, $h(x) \leq \sum_{z \in \pi^{-1}(x)} h(z)$. The statement then follows from the definition of $\hat{H}$. We have

$$\begin{aligned} h(x) &= -p(x) \log p(x) \\ &= -\left(\sum_{z \in \pi^{-1}(x)} q(z) \right) \log \left(\sum_{z \in \pi^{-1}(x)} q(z) \right) \\ &= -\sum_{z \in \pi^{-1}(x)} \left(q(z) \log \left(\sum_{s \in \pi^{-1}(x)} q(s) \right) \right) \\ &\leq -\sum_{z \in \pi^{-1}(x)} q(z) \log q(z) = \sum_{z \in \pi^{-1}(x)} h(z) \ , \end{aligned}$$

where the inequality follows from the monotonicity of the log function (and the fact that $q(z)$ is positive for all $z \in Z$).
c) Let us first recall the log-sum inequality [15, p. 31]: for non-negative numbers $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$,

$$\sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} \geq \Big(\sum_{i=1}^{n} a_i \Big) \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \ , \tag{3}$$

with equality if and only if a_i/b_i constant. We have

$$\begin{aligned} \hat{H}(Y \mid Z) &= -\sum_{z \in Z, y \in Y} q(z,y) \log \frac{q(z,y)}{q(z)} \\ &\overset{(a)}{=} -\sum_{x \in X, y \in Y} \sum_{z \in \pi^{-1}(x)} q(z,y) \log \frac{q(z,y)}{q(z)} \\ &\overset{(3)}{\leq} -\sum_{x \in X, y \in Y} \Big(\sum_{z \in \pi^{-1}(x)} q(z,y) \Big) \frac{\sum\limits_{z \in \pi^{-1}(x)} q(z,y)}{\sum\limits_{z \in \pi^{-1}(x)} q(z)} \\ &= -\sum_{x \in X, y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)} = \hat{H}(Y \mid X) \ . \end{aligned}$$

d) We have $\hat{I}(Z;Y) = \hat{H}(Y) - \hat{H}(Y \mid Z) \leq \hat{H}(Y) - \hat{H}(Y \mid X) = \hat{I}(X;Y)$ following from (c). □

X_1	X_2	X_3	X_4	Y
a	a	a	b	a
a	b	b	a	b
b	c	b	b	b
b	c	c	c	b

Table 1: **Specialization and homomorphism examples**. We have $X_1 \preceq X_2$, $X_1 \precsim X_2$, $X_1 \precsim X_3$, $X_1 \precsim X_4$, $X_2 \precsim X_3$. Note that $X_3 \not\precsim X_4$ as there is no $\sigma \in S_4$ that satisfies specialization w.r.t. X_4 and $Y \equiv Y_\sigma$

To analyze the monotonicity properties of the permutation model, the following additional definition will be 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 $X \preceq Z_\sigma$.

See Tab. 1 for examples of both introduced relations. 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, Y, with $X \precsim Z$, the following statements hold:*

a) $\hat{I}(X;Y) \leq \hat{I}(Z;Y)$
b) $m_0(X,Y,n) \leq m_0(Z,Y,n)$

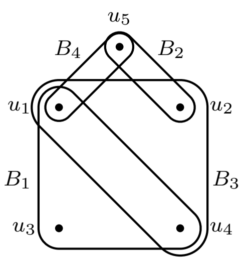
Proof Let $\sigma^* \in S_n$ be a permutation for which $Y \equiv Y_{\sigma^*}$ and $X \preceq Z_{\sigma^*}$. Property a) follows from

$$\hat{I}(Z;Y) = \hat{I}(Z_{\sigma^*};Y_{\sigma^*}) = \hat{I}(Z_{\sigma^*};Y) \geq \hat{I}(X;Y) \enspace ,$$

where the inequality holds from Prop. 1d). For b), note that for every $\sigma \in S_n$, it holds from Prop. 1d) that $\hat{I}(Z_{\sigma\circ\sigma^*};Y) \geq \hat{I}(X_\sigma;Y)$. Hence

$$\begin{aligned} m_0(Z,Y,n) =& \frac{1}{n!} \sum_{\sigma \in S_n} \hat{I}(Z_\sigma;Y) \\ =& \frac{1}{n!} \sum_{\sigma \in S_n} \hat{I}(Z_{\sigma\circ\sigma^*};Y) \\ \geq& \frac{1}{n!} \sum_{\sigma \in S_n} \hat{I}(X_\sigma;Y) = m_0(X,Y,n) \enspace . \end{aligned}$$

□



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

Fig. 2: **Base transformation example**. **Left:** a set cover instance $U = \{u_1, \ldots, u_5\}$ and $\mathcal{B} = \{\mathbf{B_1}, \mathbf{B_2}, B_3, B_4\}$. **Right:** the resulting $\mathbf{D}_{15}$ using $\tau_1(U, \mathcal{B})$ (bold indicates the set cover)

3 Hardness of optimization

In this section, we prove the NP-hardness of maximizing $\hat{F}_0$ (and hence $\hat{I}_0$) by providing a reduction from the well-known NP-hard **minimum set cover** problem: given a finite universe $U = \{u_1, \ldots, u_n\}$ and collection of subsets $\mathcal{B} = \{B_1, \ldots, B_m\} \subseteq 2^U$, find a **set cover**, i.e., a sub-collection $\mathcal{C} \subseteq \mathcal{B}$ with $\bigcup_{B \in \mathcal{C}} B = U$, that is of minimal cardinality [16, Chap. 16.1]. A **partial set cover** $\mathcal{C} \subseteq \mathcal{B}$ is one where $\bigcup_{B \in \mathcal{C}} B \neq U$.

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 the plug-in $\hat{F}$ is monotonically increasing with coverage, and in particular, set covers correspond to attribute sets with an empirical fraction of information score $\hat{F}$ of 1, and correction terms b_0 that are a monotonically increasing function of their cardinality. In a second step, we calibrate the b_0 terms such that all candidate set covers have a higher $\hat{F}_0$ value than partial set covers. The latter 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 of polynomial size. Combining these, we arrive at a polynomial time reduction, where maximizing $\hat{F}_0$ in $\mathbf{D}_{kl}$ corresponds to finding a minimal set cover for set cover instance $(U, \mathcal{B})$.

The **base transformation** $\tau_1(U, \mathcal{B}) = \mathbf{D}_l$ is defined as follows. The dataset $\mathbf{D}_l$ contains m descriptive attributes $\mathcal{I} = \{X_1, \ldots, X_m\}$ corresponding to the sets of the set cover instance, and a target variable Y. The sample size is $l = 2n + m + 1$ with a logical partition of the sample into the three regions $S_1 = [1, n]$, $S_2 = [n + 1, 2n]$, and $S_3 = [2n + 1, l]$. The target attribute Y assigns to data points one of three values corresponding to the three regions, i.e., $Y\colon [l] \rightarrow \{\mathrm{a},\mathrm{b},\mathrm{c}\}$ with

$$Y(j) = \begin{cases} \mathrm{a}, & j \in S_1 \\ \mathrm{b}, & j \in S_2 \\ \mathrm{c}, & j \in S_3 \end{cases} ,$$

and the descriptive attributes X_i assign up to $n+3$ distinct values dependending on the set of universe elements covered by set B_i, i.e., $X_i\colon [l] \rightarrow \{1,2,\ldots,n,\mathrm{a},\mathrm{b},\mathrm{c}\}$ with

$$X_i(j) = \begin{cases} j, & j \in S_1 \wedge u_j \in B_i \\ \mathrm{a}, & (j \in S_1 \wedge u_j \notin B_i) \vee j \in S_2 \\ \mathrm{b}, & j = 2n+i \\ \mathrm{c}, & j \in S_3 \setminus \{2n+i\} \end{cases} .$$

See Figure 2 for an illustration.

In a nutshell, the base transformation establishes a one-to-one correspondence between $\mathcal{C} \subseteq \mathcal{B}$ and variable sets $\mathcal{X} \subseteq \mathcal{I}$, which we denote with $\mathcal{I}(\mathcal{C})$. We note the following **two remarks**. Let us use $\mathbf{a}$ for $(\mathrm{a},\ldots,\mathrm{a})$, and $\bigcup\mathcal{C}$ as a short-cut for $\bigcup_{B\in\mathcal{C}} B$. We have that S_1 and S_2 couple the amount of uncovered elements $U \setminus \bigcup\mathcal{C}$ to the conditional entropy $\hat{H}(Y \mid \mathcal{I}(\mathcal{C}) = \mathbf{a})$ via

$$\hat{p}(Y = \mathrm{a} \mid \mathcal{I}(\mathcal{C}) = \mathbf{a}) = |U \setminus \bigcup\mathcal{C}|/(n + |U \setminus \bigcup\mathcal{C}|) .$$

In addition, part S_3 links the size of $\mathcal{C}$ to the number of distinct values of $\mathcal{I}(\mathcal{C})$ on S_3, i.e., $|\mathcal{C}| = V(\mathcal{I}(\mathcal{C})_{S_3}) - 1$. We now establish three central properties for the base transformation.

Lemma 1 *Let $\tau_1(U,\mathcal{B}) = \mathbf{D}_l$ be the transformation of a set cover instance $(U,\mathcal{B})$, and $\mathcal{C},\mathcal{C}' \subseteq \mathcal{B}$ two sets. The following statements hold.*

a) If $|\bigcup\mathcal{C}| \geq |\bigcup\mathcal{C}'|$, then $\hat{F}(\mathcal{I}(\mathcal{C});Y) \geq \hat{F}(\mathcal{I}(\mathcal{C}');Y)$, i.e., the plug-in $\hat{F}$ is monotonically increasing with coverage, and in particular, $\mathcal{C}$ is a set cover if and only if $\hat{F}(\mathcal{I}(\mathcal{C});Y) = 1$,

b) If C is a set cover and C' is not, *then $\hat{I}(\mathcal{I}(\mathcal{C});Y) - \hat{I}(\mathcal{I}(\mathcal{C}');Y) \geq 2/l$.*

c) If $\mathcal{C}$ and $\mathcal{C}'$ are both set covers, then $\mathcal{I}(\mathcal{C}) \precsim \mathcal{I}(\mathcal{C}')$ if and only if $|\mathcal{C}| \leq |\mathcal{C}'|$.

Proof Statement a) follows from the definition of τ_1.

To show b), since $\hat{F}(\mathcal{I}(\mathcal{C}');Y)$ and thus $\hat{I}(\mathcal{I}(\mathcal{C}');Y)$ are monotone in $|\bigcup\mathcal{C}'|$, it is sufficient to consider the case where $|U \setminus \bigcup\mathcal{C}'| = 1$, i.e., only one element $u \in U$ is uncovered. In this case we have

$$\hat{I}(\mathcal{I}(\mathcal{C});Y) - \hat{I}(\mathcal{I}(\mathcal{C}');Y) = \hat{H}(Y \mid \mathcal{I}(\mathcal{C}')) - \underbrace{\hat{H}(Y \mid \mathcal{I}(\mathcal{C}))}_{=0}$$

and, moreover, as required

$$\begin{aligned} \hat{H}(Y \mid \mathcal{I}(\mathcal{C}')) &= -\hat{p}(\mathbf{a},\mathrm{a}) \log \hat{p}(\mathrm{a} \mid \mathbf{a}) - \hat{p}(\mathbf{a},\mathrm{b}) \log \hat{p}(\mathrm{b} \mid \mathbf{a}) \\ &= -\frac{1}{l} \log\left(\frac{1}{n+1}\right) - \frac{n}{l} \log\left(\frac{n}{n+1}\right) \geq \frac{2}{l} . \end{aligned}$$

For c) observe that for variable set $\mathcal{X} = \mathcal{I}(\mathcal{C})$ corresponding to set cover $\mathcal{C}$, we have for all $i, j \in S_1$ that $\mathcal{X}(i) \neq \mathcal{X}(j)$. Thus, $\mathcal{X}_{S_1} \equiv \mathcal{X}'_{S_1}$ for variable set $\mathcal{X}' = \mathcal{I}(\mathcal{C}')$ corresponding to set cover $\mathcal{C}'$. Moreover, we trivially have $\mathcal{X}_{S_2} \equiv \mathcal{X}'_{S_2}$. Finally, let $Q, Q' \subseteq S_3$ denote the indices belonging to S_3 where $\mathcal{X}$ and $\mathcal{X}'$ take on values different from $(c, \ldots, c)$. Note that all values in these sets are unique. Furthermore, if $|\mathcal{C}| \leq |\mathcal{C}'|$ then $|Q| \leq |Q'|$ and in turn $|Q \setminus Q'| \leq |Q' \setminus Q|$. This means we can find a permutation $\sigma \in S_n$ such that for all $i \in Q \setminus Q'$ it holds that $\sigma(i) = j$ with $j \in Q' \setminus Q$ and $\sigma(i) = i$ for $i \notin Q \cap Q'$ (that is σ permutes all indices of non-$(c, \ldots, c)$ values of $\mathcal{C}$ in S_3 to indices of non-$(c, \ldots, c)$ values of $\mathcal{C}'$). For such a permutation it holds that $Y_\sigma \equiv Y$ and $\mathcal{X}_{S_3} \preceq \mathcal{X}'_{S_3\sigma}$. Therefore, $\mathcal{X} \precsim \mathcal{X}'$ as required. □

Now, although set covers $\mathcal{C} \subseteq \mathcal{B}$ correspond to variable sets $\mathcal{I}(\mathcal{X})$ with the maximal empirical fraction of information value of 1, due to the correction term, it can happen that $\hat{F}_0(\mathcal{I}(\mathcal{X}'); Y) \geq \hat{F}_0(\mathcal{I}(\mathcal{X}); Y)$ for a variable set $\mathcal{I}(\mathcal{X}')$ corresponding to a partial set cover. To prevent this, we make use of the following upper-bound of the expected mutual information under the permutation model.

Proposition 3 ([13], Thm. 7) *For a sample of size n of the joint distribution of variables A and B having $a, b \in \mathbb{Z}_+$ distinct values, respectively, we have*

$$m_0(A, B, n) \leq \log\left(\frac{n + ab - a - b}{n - 1}\right) .$$

Proposition 3 implies that we can arbitrarily shrink the correction terms if we increase the sample size but leave the number of distinct values constant. Thus, we define the **extended transformation** $\tau_i(U, \mathcal{B}) = \mathbf{D}_{il}$ through simply copying $\mathbf{D}_l$ a number of i times, i.e., by defining $\mathbf{d}_j = \mathbf{d}_{(j \mod l)}$ for $j \in [l+1, il]$. With this definition, we proceed with the NP-hardness result.

Theorem 2 *Given a sample of the joint distribution of variables $\mathcal{I}$ and Y, the problem of maximizing $\hat{F}_0(\,\cdot\,; Y)$ over all possible subsets $\mathcal{X} \subseteq \mathcal{I}$ is NP-hard.*

Proof First, let us assume that there exists a number $k \in O(l)$ such that w.r.t. transformation τ_k, all set covers $\mathcal{C} \subseteq \mathcal{B}$ and their corresponding variable sets $\mathcal{X} = \mathcal{I}(\mathcal{C})$ have correction terms with $m_0(\mathcal{X}, Y, kl) < 2/l$. Since all properties of Lemma 1 transfer from τ_1 to τ_k, this implies that for all variable sets $\mathcal{X}' = \mathcal{I}(\mathcal{C}')$ corresponding to partial set covers $\mathcal{C}' \subseteq \mathcal{B}$, it holds that

$$\begin{aligned}
\hat{F}_0(\mathcal{X}; Y) &= \hat{F}(\mathcal{X}; Y) - m_0(\mathcal{X}, Y, kl)/\hat{H}(Y) \\
&> \hat{F}(\mathcal{X}; Y) - 2/(l\hat{H}(Y)) \\
&\geq \hat{F}(\mathcal{X}; Y) - (\hat{I}(\mathcal{X}; Y) - \hat{I}(\mathcal{X}'; Y))/\hat{H}(Y) \\
&= \hat{F}(\mathcal{X}'; Y) \geq \hat{F}_0(\mathcal{X}'; Y) \ ,
\end{aligned}$$

where the greater-than follows from Lemma 1a) and 1b). Thus, all $\mathcal{X}$ corresponding to set covers have larger $\hat{F}_0$ than partial set covers. Moreover, we know that $\mathcal{C}$ must be a minimum set cover as required, because for a smaller set cover $\mathcal{C}'$, we would have $\mathcal{I}(\mathcal{C}') \precsim \mathcal{I}(\mathcal{C})$ by Lemma 1c), and thus $b_0(\mathcal{I}(\mathcal{C}'), Y, kl) \leq b_0(\mathcal{I}(\mathcal{C}), Y, kl)$ from Proposition 2b)—therefore, $\mathcal{I}(\mathcal{C})$ would not maximize $\hat{F}_0$.

Now, to find the number k that defines the final transformation τ_k, let $\mathbf{D}_{il} = \tau_i(U, \mathcal{B})$ and $\mathcal{C}$ be a set cover of $(U, \mathcal{B})$. Since $\mathcal{X} = \mathcal{I}(\mathcal{C})$ has at most l distinct values in $\mathbf{D}_{il}$ and Y exactly 3, from Proposition 3 and the monotonicity of ln, we have that

$$\ln(2) m_0(\mathcal{I}(\mathcal{C}), Y, n) \leq \ln\left(\frac{il + 3l}{il - 1}\right) \leq \ln\left(\frac{i+3}{i-1}\right) \leq \frac{4}{i-1} \ ,$$

where the last inequality follows from $\ln(x) \leq x - 1$. Thus, for $k > 2l/\ln 2 + 1 \in O(l)$ we have $m_0(\mathcal{X}, Y, kl) < 2/l$ as required. The proof is concluded by noting that the final transformation $\tau_k(U, \mathcal{B})$ is of size $O(l^2 m)$ (where $l = 2n + m + 1$), which is polynomial in the size of the set cover instance. □

4 Admissible bounding functions for effective search algorithms

The NP-hardness established in the previous section excludes the existence of a polynomial time algorithm for maximizing the reliable fraction of information (unless P=NP), 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 bounding functions (also called optimistic estimators) for the reliable fraction of information $\hat{F}_0$ to be used for pruning.

Recall that an **admissible bounding function** $\bar{f}$ is an upper bound to the optimization function value f of all supersets of a candidate solution $\mathcal{X} \subseteq \mathcal{I}$. The value $\bar{f}(\mathcal{X})$ is called the **potential** of node $\mathcal{X}$, and it must hold that $\bar{f}(\mathcal{X}) \geq f(\mathcal{X}')$ for all $\mathcal{X}'$ with $\mathcal{X} \subseteq \mathcal{X}' \subseteq \mathcal{I}$. With this property, all supersets $\mathcal{X}'$ of $\mathcal{X}$ can be pruned if $\bar{f}(\mathcal{X}) \leq f(\mathcal{X}^*)$, where $\mathcal{X}^*$ is the best candidate solution found during search. Therefore, for optimal pruning, the bounding function has to be as tight as possible. At the same time, it needs to be efficiently computable. For example, while the **ideal bounding function** for the reliable fraction of information would be

$$\bar{f}_{\text{ideal}}(\mathcal{X}) = \max\{\hat{F}_0(\mathcal{X}'; Y) \colon \mathcal{X} \subseteq \mathcal{X}' \subseteq \mathcal{I}\} \ , \tag{4}$$

solving Eq. (4) is equivalent to the original optimization problem and hence NP-hard.

A first attempt for an efficient bounding function involves the upper bound of the fraction of information (i.e., $F = 1$) and the monotonicity of the b_0 term with respect to the subset relation (Theorem 1). In particular, for all $\mathcal{X} \subseteq \mathcal{X}' \subseteq \mathcal{I}$, it follows that

$$\begin{aligned} \hat{F}_0(\mathcal{X}'; Y) =& \frac{\hat{H}(Y) - \hat{H}(Y \mid \mathcal{X}')}{\hat{H}(Y)} - b_0(\mathcal{X}', Y, n) \\ \leq& 1 - b_0(\mathcal{X}, Y, n) \ . \end{aligned}$$

Hence, we define

$$\bar{f}_{\text{mon}}(\mathcal{X}) = 1 - b_0(\mathcal{X}, Y, n) \tag{5}$$

to be the **monotonicity-based** admissible bounding function. Equation (5) is in practice inexpensive, as one can cache the $b_0(\mathcal{X}, Y, n)$ term while computing $\hat{F}_0(\mathcal{X}; Y)$ during search. However, it is potentially loose as it assumes that full information about the target can be attained, without the "penalty" of an increased b_0 term.

An alternative idea leading to a more principled admissible bounding function, is to relax the maximum over all supersets to the maximum over all *specializations* of $\mathcal{X}$. We define the **specialization-based** bounding function $\bar{f}_{\text{spc}}(\mathcal{X})$ through

$$\begin{aligned}\bar{f}_{\text{spc}}(\mathcal{X}) =& \max\{\hat{F}_0(\mathcal{X}'; Y) \colon\ \mathcal{X} \preceq \mathcal{X}'\} \qquad (6)\\ \geq& \max\{\hat{F}_0(\mathcal{X}'; Y) \colon\ \mathcal{X} \subseteq \mathcal{X}' \subseteq \mathcal{I}\} = \bar{f}_{\text{ideal}}(\mathcal{X})\ .\end{aligned}$$

While Eq. (6) constitutes an admissible bounding function, it is unclear how it can be efficiently evaluated. To do so, let us denote by R^+ the operation of joining a labeling R with the target attribute Y, i.e., $R^+ = \{R, Y\}$ (see Table 2 for an example). 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 $O(n|V(\mathcal{X})||V(Y)|)$.*

Proof We start by showing that the $(\cdot)^+$ operation causes a positive gain in $\hat{F}_0$, i.e., for an arbitrary labeling R it holds that $\hat{F}_0(R^+; Y) \geq \hat{F}_0(R; Y)$. It is sufficient to show that $\hat{I}_0(R^+; Y) \geq \hat{I}_0(R; Y)$. We have

$$\begin{aligned}\hat{I}_0(R^+; Y) =& \left(\hat{H}(Y) + \hat{H}(R^+) - \hat{H}(R^+, Y)\right)\\ &- \frac{1}{n!}\left(\sum_{\sigma \in S_n} (\hat{H}(Y_\sigma) + \hat{H}(R^+) - \hat{H}(R^+, Y_\sigma)\right)\\ =& \frac{1}{n!}\sum_{\sigma \in S_n} \hat{H}(R^+, Y_\sigma) - \hat{H}(R^+, Y)\\ \geq& \frac{1}{n!}\sum_{\sigma \in S_n} \hat{H}(R, Y_\sigma) - \hat{H}(R, Y) = \hat{I}_0(R; Y)\ ,\end{aligned}$$

since $\hat{H}(R^+, Y) = \hat{H}(R \cup Y, Y) = \hat{H}(R, Y)$, and from Proposition 1b), for every $\sigma \in S_n$, $\hat{H}(R^+, Y_\sigma) \geq \hat{H}(R, Y_\sigma)$.

To conclude, let $\mathcal{Z}$ be an arbitrary specialization of $\mathcal{X}$. We have by definition of $\mathcal{Z}$ and $\mathcal{Z}^+$, that $\mathcal{X}^+ \preceq \mathcal{Z}^+$. Moreover, $\hat{F}(\,\cdot\,; Y) = \hat{F}(\{\,\cdot\,\} \cup \{Y\}; Y) = 1$.

Thus

$$\begin{aligned}\hat{F}_0(\mathcal{X}^+;Y) =&\hat{F}(\mathcal{X}^+;Y) - b_0(\mathcal{X}^+,Y,n)\\ =&1 - b_0(\mathcal{X}^+,Y,n)\\ \geq&1 - b_0(\mathcal{Z}^+,Y,n)\\ =&\hat{F}_0(\mathcal{Z}^+;Y) \geq \hat{F}_0(\mathcal{Z};Y) \ ,\end{aligned}$$

as required. Here, the first inequality follows from Proposition 1b), the second from the positive gain of $\mathcal{Z}^+$ over $\mathcal{Z}$.

Regarding the complexity, recall that $b_0(\mathcal{X},Y,n)$ can be computed in time $O(n \max\{|V(\mathcal{X})|,|V(Y)|\})$. The complexity follows from $|V(\mathcal{X}^+)| \leq |V(\mathcal{X})||V(Y)|$. □

Intuitively, $\mathcal{X}^+$ constitutes the most efficient specialization of $\mathcal{X}$ in terms of growth in $\hat{F}$ and b_0 (which is not necessarily attainable by a subset of input variables). Note that the $\mathcal{X}^+$ operation is not computed explicitly since it is obtained as the non-zero cell counts of the joint contingency table for $\mathcal{X}$ and Y (which has to be computed for $\hat{F}_0(\mathcal{X};Y)$ anyway). The following proposition shows that this idea indeed leads to a superior bound compared to $\bar{f}_{\text{mon}}$.

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

a) $\varDelta \geq 0$ 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. $\varDelta \in \varOmega(1 - \frac{1}{\log 2l})$

Proof a)

$$\begin{aligned}\bar{f}_{\text{spc}}(\mathcal{X}) =&1 - b_0(\mathcal{X}^+,Y,n)\\ \leq&1 - b_0(\mathcal{X},Y,n) = \bar{f}_{\text{mon}}(\mathcal{X}) \ ,\end{aligned}$$

where the inequality holds from Proposition 1b) and $\mathcal{X} \preceq \mathcal{X}^+$.

b) For $l \geq 1$ we construct a dataset $\mathbf{D}_{4l}$ with two variables $X\colon [4l] \to \{\text{a},\text{b}\}$ and $Y\colon [4l] \to [2l]$, with

$$X(i) = \begin{cases}\text{a}, & i \mod 2 = 1\\ \text{b}, & i \mod 2 = 0\end{cases}$$

and $Y(i) = \lceil i/2 \rceil$ respectively (see Table 2). We have

$$\begin{aligned}\varDelta &= 1 - b_0(X,Y,4l) - 1 + \underbrace{b_0(X^+,Y,4l)}_{=\hat{H}(Y \mid X^+_\sigma)/\hat{H}(Y)=0}\\ &= \frac{1}{n!}\sum_{\sigma \in S_n} \hat{H}(Y \mid X_\sigma)/\hat{H}(Y)\\ &\geq \min_{\sigma \in S_n} \hat{H}(Y \mid X_\sigma)/\hat{H}(Y) \ .\end{aligned}$$

X	Y	X^+	X_{σ^*}
a	1	(a,1)	a
b	1	(b,1)	a
a	2	(a,2)	a
b	2	(b,2)	a
		⋮	

X	Y	X^+	X_{σ^*}
		⋮	
a	2l-1	(a,2l-1)	b
b	2l-1	(b,2l-1)	b
a	2l	(a,2l)	b
b	2l	(b,2l)	b

Table 2: **Construction showing the advantage of bound $\bar{f}_{\mathbf{spc}}$ versus $\bar{f}_{\mathbf{mon}}$.** We have $\bar{f}_{\text{spc}}(X) = 1 - b_0(X^+, Y, n) = 0$ while $\bar{f}_{\text{mon}}(X) = 1 - b_0(X, Y, n) \geq 1 - 1/\log(n/2)$, i.e., all specializations of X that contain full information about Y are injective (key) maps (see Proposition 4).

One can show that the minimum of the last step is attained by the permutation $\sigma^* \in S_n$ with

$$\sigma^*(i) = \begin{cases} 2i - 1, & i \in [1, 2l] \\ 4l - 2(4l - i), & i \in [2l + 1, 4l] \end{cases} ,$$

which corresponds to sorting the a and b values of X (see Table 2). For this permutation the normalized conditional entropy evaluates to $1 - 1/\log(2l)$ as required. □

Thus, we have established that $\bar{f}_{\text{spc}}$ is tighter than $\bar{f}_{\text{mon}}$, and even that the difference can be arbitrary close to 1. Put differently, their ratio, and thus the potential for additional pruning, is unbounded.

Computationally, $\bar{f}_{\text{spc}}(\mathcal{X})$ is more expensive than $\bar{f}_{\text{mon}}(\mathcal{X})$ by a factor of $|V(Y)|$. In order to partially alleviate this increase, note that one can first check the pruning condition w.r.t. $\bar{f}_{\text{mon}}$ and only compute $\bar{f}_{\text{spc}}$ if that first check fails. That is, whenever $\bar{f}_{\text{mon}}(\mathcal{X})$ is sufficient to prune a candidate $\mathcal{X}$ we can still do so with the same computational complexity. However, the additional evaluation of $\bar{f}_{\text{spc}}(\mathcal{X})$ can be a disadvantage in case it still does not allow to prune. This trade-off is evaluated in Sec. 6.3.

5 Algorithms

In this section we provide two search algorithms, one exponential and one heuristic, for maximizing the reliable fraction of information. Both make use of the bounding functions proposed. For simplicity, we solve the top-1 problem, but both algorithms can be trivially extended to a top-k formulation.

5.1 Exponential search

Branch-and-bound, as the name suggests, consists of two main ingredients, a strategy to explore the search space and a bound for the optimization

Algorithm 1 OPUS: Given a set of input variables $\mathcal{I}$, function f, bounding function $\bar{f}$, and $\alpha \in (0,1]$, the algorithm returns the $\mathcal{X}^* \subseteq \mathcal{I}$ satisfying $f(\mathcal{X}^*) \geq \alpha \max\{f(\mathcal{X}')\colon \mathcal{X}' \subseteq \mathcal{I}\}$

```
1: function OPUS($\mathbf{Q}, \mathcal{S}$)
2:     if $\mathbf{Q}$ is empty then
3:         return $\mathcal{S}$
4:     else
5:         $(\mathcal{X}, \mathcal{Z}) = pop(\mathbf{Q})$
6:         $\mathbf{R} = \{(\mathcal{X} \cup \{Z\}, Z)\colon Z \in \mathcal{Z}\}$
7:         $\mathcal{X}^* = \arg\max\{f(\mathcal{X}')\colon \mathcal{X}' \in \mathbf{R} \cup \{\mathcal{S}\}\}$
8:         $\mathbf{R}' = \{(\mathcal{X}', Z) \in \mathbf{R}\colon \alpha \bar{f}(\mathcal{X}') > f(\mathcal{X}^*)\}$
9:         $\mathcal{Z}' = \{Z\colon (\mathcal{X}', Z) \in \mathbf{R}'\}$
10:        $[(\mathcal{X}_1, Z_1), \ldots, (\mathcal{X}_k, Z_k)] = sort(\mathbf{R}')$
11:        $\mathbf{Q}' = \mathbf{Q} \cup \{(\mathcal{X}_i, \mathcal{Z}' \setminus \{Z_1, \ldots, Z_i\})\colon i \in [k])\}$
12:        return OPUS($\mathbf{Q}', \mathcal{X}^*$)
13: $\mathcal{X}^* = $ OPUS($\{(\emptyset, \mathcal{I})\}, \emptyset$)
```

function at hand (see, e.g., [17, Chap. 12.4]). Besides being very effective in practice for hard problems, this style of optimization also provides the option of relaxing the required result guarantee to that of an α-approximation for accuracy parameter $\alpha \in (0,1]$. Hence, using α-values of less than 1 allows to trade accuracy for computation time in a principled manner. Here, we consider **optimized pruning for unordered search** (**OPUS**), an advanced variant of branch-and-bound that effectively propagates pruning information to siblings in the search tree [18]. Algorithm 1 shows the details of this approach.

In addition to keeping track of the best solution $\mathcal{X}^*$ explored so far, the algorithm maintains a priority queue $\mathbf{Q}$ of pairs $(\mathcal{X}, \mathcal{Z})$, where $\mathcal{X} \subseteq \mathcal{I}$ is a candidate solution and $\mathcal{Z} \subseteq \mathcal{I}$ constitutes the variables that can still be used to refine $\mathcal{X}$, e.g., $\mathcal{X}' = \mathcal{X} \cup \{Z\}$ for a $Z \in \mathcal{Z}$. The top element is the one with the smallest cardinality and the highest $\bar{f}$ value (a combination of breadth-first and best-first order). Starting with $\mathbf{Q} = \{(\emptyset, \mathcal{I})\}$, $\mathcal{X}^* = \emptyset$, and a desired approximation guarantee $\alpha \in (0,1]$, in every iteration OPUS creates all refinements of the top element of $\mathbf{Q}$ and updates $\mathcal{X}^*$ accordingly (lines 5-7). Next the refinements are pruned using $\bar{f}$ and α (line 8). Following, the pruned list is sorted according to decreasing potential (a "trick" to propagate the most refinement elements to the least promising candidates [18]), the possible refinement elements $\mathcal{Z}'$ are non-redundantly propagated to the refinements of the top element, and finally the priority queue is updated with the new candidates (lines 9-11).

5.2 Heuristic search

A commonly used alternative to exponential search for optimizing dependency measures is the standard **greedy algorithm** (see [19; 20]). This algorithm only refines the best candidate in a given iteration. Moreover, bounding functions can be incorporated as an early termination criterion. For the reliable fraction

Algorithm 2 GRD: Given a set of input variables $\mathcal{I}$, function f, and bounding function $\bar{f}$, the algorithm returns the $\mathcal{X}^* \subseteq \mathcal{I}$ approximating $f(\mathcal{X}^*) = \max\{f(\mathcal{X}')\colon \mathcal{X}' \subseteq \mathcal{I}\}$

```
1: function GRD($\mathcal{C}, \mathcal{S}$)
2:     if $\mathcal{I} \setminus \mathcal{C}$ is empty or $\bar{f}(\mathcal{C}) \leq f(\mathcal{S})$ then
3:         return $\mathcal{S}$
4:     else
5:         $\mathbf{R} = \{\mathcal{C} \cup \{Z\}\colon Z \in \mathcal{I} \setminus \mathcal{C}\}$
6:         $\mathcal{C}^* = \arg\max\{f(\mathcal{X}')\colon \mathcal{X}' \in \mathbf{R}\}$
7:         $\mathcal{X}^* = \arg\max\{f(\mathcal{X}')\colon \mathcal{X}' \in \{\mathcal{S}, \mathcal{C}^*\}\}$
8:         return GRD($\mathcal{C}^*, \mathcal{X}^*$)
9: $\mathcal{X}^* =$ GRD($\emptyset, \emptyset$)
```

of information in particular, there is potential to prune many of the higher levels of the search space. The algorithm is presented in Algorithm 2.

The algorithm keeps track of the best solution $\mathcal{X}^*$ explored, as well as the best candidate for refinement $\mathcal{C}^*$. Starting with $\mathcal{X}^* = \emptyset$ and $\mathcal{C}^* = \emptyset$, the algorithm in each iteration (i.e., search space level) checks whether $\mathcal{C}^*$ can be refined further, i.e., if $\mathcal{I} \setminus \mathcal{C}^*$ is not empty, or if $\mathcal{C}^*$ has potential (the early termination criterion). If not, the algorithm terminates returning $\mathcal{X}^*$ (lines 2-3). Otherwise $\mathcal{C}^*$ is refined to all possible refinements, and the best one is selected as a candidate to update $\mathcal{X}^*$ (lines 5-7).

Concerning the approximation ratio of the greedy algorithm, there exists a large amount of research focused on submodular and/or monotone functions, e.g., [21; 22; 23]. Recall that for a set $\mathcal{I} = \{X_1, \ldots, X_d\}$, a function $f\colon 2^{\mathcal{I}} \to \mathbb{R}$ is called submodular if for every $\mathcal{X} \subseteq \mathcal{X}' \subseteq \mathcal{I}$ and $X_i \in \mathcal{I} \setminus \mathcal{X}'$, it holds that

$$f(\mathcal{X}' \cup \{X_i\}) - f(\mathcal{X}') \leq f(\mathcal{X} \cup \{X_i\}) - f(\mathcal{X}) \ ,$$

i.e., it satisfies the diminishing returns property. The following proposition establishes that I, $\hat{I}$, and $\hat{I}_0$, are all violating this property.

Proposition 5 *Given $\mathcal{I} = \{X_1, \ldots, X_d\}$ and target variable Y, the mutual information I, the plug-in $\hat{I}$, and corrected $\hat{I}_0$ are not submodular.*

Proof We prove it via an intuitive counter example. Let us consider the data of Table 3 and the corresponding induced empirical distribution $\hat{p}$. Here B and C are connected to Y via a XOR function, where Y is marginally independent of B and C, but functionally dependent on $\{B, C\}$. For sets $\{A\}$, $\{A, B\}$, and element C, we have that

$$\begin{aligned}\hat{I}(\{A,B,C\};Y) - \hat{I}(\{A,B\};Y) =& 0.5\\ &> \hat{I}(\{A,C\};Y) - \hat{I}(\{A\};Y) = 0.19 \ ,\end{aligned}$$

i.e., there is a violation of the diminishing returns property, and hence $\hat{I}$ is not submodular. By considering a distribution p for which we have $p = \hat{p}$, it is straightforward to show that I is also not submodular.

A	B	C	Y
a	a	a	a
a	a	b	b
a	b	b	a
b	b	a	b

Table 3: **Example data for non-submodularity of $I, \hat{I}, \hat{I}_0$.**

Regarding $\hat{I}_0$, we have that

$$\begin{aligned}\hat{I}_0(\{A,B,C\};Y) - \hat{I}_0(\{A,B\};Y) =& 0.17 \\ & > \hat{I}_0(\{A,C\};Y) - \hat{I}_0(\{A\};Y) = -0.17 \ ,\end{aligned}$$

and hence $\hat{I}_0$ is not submodular. Also note that while both $\hat{I}$ and I are monotone functions with respect to the subset relation (Theorem 1), $\hat{I}_0$ is not. □

While approximation results for submodular and/or monotone functions are not applicable to $\hat{I}_0$, we empirically evaluate the quality of solutions in Sec. 6.3.2.

6 Evaluation

In this section, we investigate the empirical performance of discovering dependencies with the reliable fraction of information $\hat{F}_0$, including the estimated bias and variance of $\hat{F}_0$ as an estimator, the consistency of correctly retrieving the top minimal dependency on synthetic data, the performance of the bounding functions for both branch-and-bound and greedy search, and two concrete examples of functional dependencies in real-world datasets.

6.1 Empirical bias and standard deviation

Here, we evaluate the **estimated bias and variance** of $\hat{F}_0$ for various degrees of dependency. We do so by creating synthetic data from various models for which we know the true F. Let us denote by $\mathcal{P}$ the set of all joint probability mass functions over two random variables X and Y with $|V(X)| = |V(Y)| = 3$, and by $\mathcal{P}_{[a,b]}$ all such probability mass functions for which we have a score of $F_p(X;Y) \in [a,b]$. We consider four different dependency score regions: "weak" $\mathcal{P}_{[0,0.25)}$, "low" $\mathcal{P}_{[0.25,0.5)}$, "high" $\mathcal{P}_{[0.5,0.75)}$, and "strong" $\mathcal{P}_{[0.75,1]}$.

Let $\tau(\mathbf{D}_n)$ be the result of estimator τ computed on data $\mathbf{D}_n$. We denote with $b_n(p,\tau)$ and $std_n(p,\tau)$ the bias and standard deviation of τ when fixing the underlying pmf to $p \in \mathcal{P}$, i.e., $b_n(p,\tau) = \mathbb{E}_{\mathbf{D}_n \sim p}[\tau(\mathbf{D}_n)] - F_p(X;Y)$ and $std_n(p,\tau) = \sqrt{\mathbb{E}_{\mathbf{D}_n \sim p}[(\tau(\mathbf{D}_n) - \mathbb{E}_{\mathbf{D}_n \sim p}[\tau(\mathbf{D}_n)])^2]}$. We sample uniformly 100 pmfs $p^{(1)}, \ldots, p^{(100)}$, 25 from each dependency region. For every $p^{(i)}$ we calculate the true $F_{p^{(i)}}$ value and compute the expectation terms by sampling per pmf $p^{(i)}$ a total of 1000 datasets $D_n \sim p^{(i)}$ of size n. We average over $\mathcal{P}_{[a,b]}$ regions

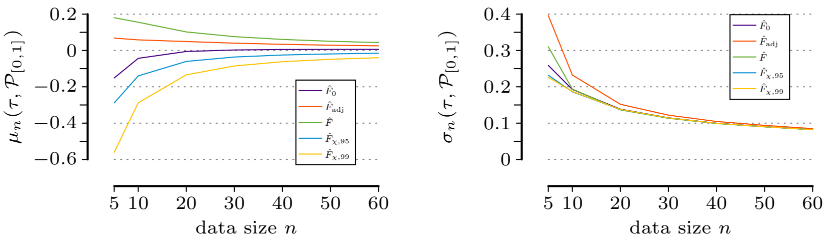


Fig. 3: **Empirical bias and standard deviation of estimators averaged over $p \in \mathcal{P}_{[0,1]}$.** Average bias $\mu_n(\tau, \mathcal{P}_{[0,1]})$ (**left**) and standard deviation $\sigma_n(\tau, \mathcal{P}_{[0,1]})$ (**right**) of estimators $\tau \in \{\hat{F}_0, \hat{F}_{\text{adj}}, \hat{F}, \hat{F}_{\chi,95}, \hat{F}_{\chi,99}\}$ for all 100 sampled pmfs $p^{(i)} \in \mathcal{P}_{[0,1]}$ across different data sizes n.

and end up with estimates $\mu_n(\tau, \mathcal{P}_{[a,b]})$ and $\sigma_n(\tau, \mathcal{P}_{[a,b]})$ for the average bias and standard deviation of estimator τ and sample size n.

In addition to the plug-in $\hat{F}$, we consider **two additional estimators**. The first is based on the same correction principle but with a parametric model and asymptotic values, and particular the χ^2 distribution, proposed by Vinh et al. [24]. This corrected estimator, which we denote as $\hat{F}_{\chi,\alpha}$, is defined as

$$\hat{F}_{\chi,\alpha}(\mathcal{X}; Y) = \frac{\hat{I}(\mathcal{X}, Y) - \frac{1}{2n}\chi_{\alpha,l(\mathcal{X},Y)}}{\hat{H}(Y)} \ ,$$

where $\chi_{\alpha,l(\mathcal{X},Y)}$ is the critical value corresponding to a significance level $1-\alpha$ and degrees of freedom $l(\mathcal{X}, Y) = (\prod_{X\in\mathcal{X}} V(X) - 1)(V(Y) - 1)$. Here, α can be thought as a parameter regulating the amount of penalty. The second follows an alternative correction resulting from the application of the quantification adjustment framework proposed by Romano et al. [6]. We denote this estimator by $\hat{F}_{\text{adj}}$, which is defined as

$$\hat{F}_{\text{adj}}(\mathcal{X}; Y) = \frac{\hat{I}(\mathcal{X}, Y) - \mathbb{E}_0[\hat{I}(\mathcal{X}, Y)]}{\hat{H}(Y) - \mathbb{E}_0[\hat{I}(\mathcal{X}, Y)]} \ .$$

For this experiment we consider $\tau = \{\hat{F}_0, \hat{F}_{\text{adj}}, \hat{F}, \hat{F}_{\chi,95}, \hat{F}_{\chi,99}\}$ and $n \in \{5, 10, 20, 30, 40, 50, 60\}$.[2] We expect the small sample sizes for the small domain size $|V(X)| = 3$ to behave similar to larger data sizes combined with the potentially huge domains $V(\mathcal{X})$ for $\mathcal{X} \subseteq \mathcal{I}$ occurring during search.

We first focus on the general behavior of the bias and standard deviation for each estimator τ, and plot in Figure 3 the average bias $\mu_n(\tau, \mathcal{P}_{[0,1]})$ and standard deviation $\sigma_n(\tau, \mathcal{P}_{[0,1]})$ across different data sizes n. We observe that the corrected estimator $\hat{F}_0$ exchanges the positive bias of $\hat{F}$ for a smaller, negative bias, and has the tendency to underestimate the true dependency for small n, as desired. Additionally, it converges very fast to 0 with respect to

[2] the α values in $\hat{F}_{\chi,\alpha}$ are chosen according to Vinh et al. [24]

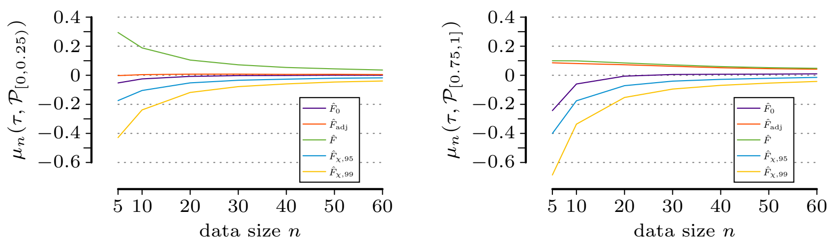


Fig. 4: **Bias of estimators averaged over $p \in \mathcal{P}_{[0,0.25)}$ and $p \in \mathcal{P}_{[0.75,1]}$.** Average bias $\mu_n(\tau, \mathcal{P}_{[0,0.25)})$ (**left**) and $\mu_n(\tau, \mathcal{P}_{[0.75,1]})$ (**right**) of estimators $\tau \in \{\hat{F}_0, \hat{F}_{\text{adj}}, \hat{F}, \hat{F}_{\chi,95}, \hat{F}_{\chi,99}\}$ across different data sizes n.

n. The $\hat{F}_{\text{adj}}$ has a very small positive bias, while the $\hat{F}_{\chi,\alpha}$ has a large negative bias and slow convergence that become more profound for increased α.

Regarding the standard deviation, the right plot show that the $\hat{F}_{\text{adj}}$ has by far the largest, which is to be expected as it also has the smallest bias. The plug-in $\hat{F}$ also has a large standard deviation that in combination with the relatively high bias, show that $\hat{F}$ is not a suitable estimator for functional dependency discovery. The $\hat{F}_{\chi,95}$, $\hat{F}_{\chi,99}$, and $\hat{F}_0$, have similar standard deviations, with $\hat{F}_0$ being slightly higher for $n = 5$. In general, estimators achieve better bias by trading variance, and from Figure 3 we see that in comparison to all estimators, $\hat{F}_0$ has the best bias for variance trade-off.

It is also interesting to consider the bias behavior not on average for $\mathcal{P}_{[0,1]}$, but specifically for weak and strong dependencies, i.e., the cases where F is closer to independence and functional dependency respectively, and plot in Figure 4 the average biases $\mu_n(\tau, \mathcal{P}_{[0,0.25)})$ (left) and $\mu_n(\tau, \mathcal{P}_{[0.75,1]})$ (right). Looking at the left plot we see that the reliable fraction of information $\hat{F}_0$ has a very small negative bias, and $\hat{F}$ has the largest positive bias and very slow convergence. Both $\hat{F}_{\chi,95}$ and $\hat{F}_{\chi,99}$ have a large negative bias, particularly $\hat{F}_{\chi,99}$, while $\hat{F}_{\text{adj}}$ is practically unbiased. Regarding strong dependencies, the right plot shows that both $\hat{F}, \hat{F}_{\text{adj}}$ have a small positive bias, while the rest have large negative biases for $n = 5$. For both $\hat{F}_{\chi,95}$ and $\hat{F}_{\chi,99}$ the bias is particularly high and does not converge fast to 0, unlike $\hat{F}_0$ that does after only $n = 10$ data samples. From a bias perspective, $\hat{F}_0$ shows the best reliable behavior, with small and "fast" negative bias across the whole range of dependencies.

With these observations, we can conclude that $\hat{F}_0$ is a suitable estimator for F, and particularly for exploratory tasks, as it does not require parameters and parametric assumptions in order to produce results. The $\hat{F}_{\text{adj}}$, although practically unbiased, has a very large standard deviation. The $\hat{F}_{\chi,\alpha}$ has the ability of regulating the amount of penalty with α, but that requires prior knowledge about the data. For example, $\alpha = 0.99$ and $\alpha = 0.95$ heavily penalize and can miss dependencies in higher levels. Smaller α values will cause $\hat{F}_{\chi,\alpha}$ to start behaving more like $\hat{F}$ and overestimate dependencies. The reliable $\hat{F}_0$ does that automatically with the data-dependent quantity $\mathbb{E}_0[\hat{I}]$.

$p(X_4 = \mathrm{a})$	$p(X_4 = \mathrm{b})$	$p(X_4 = \mathrm{c})$
0.5	0.3	0.2

X_4	$p(Y = \mathrm{a} \mid X_4)$	$p(Y = \mathrm{b} \mid X_4)$
a	0.7	0.3
b	0.2	0.8
c	0.9	0.1

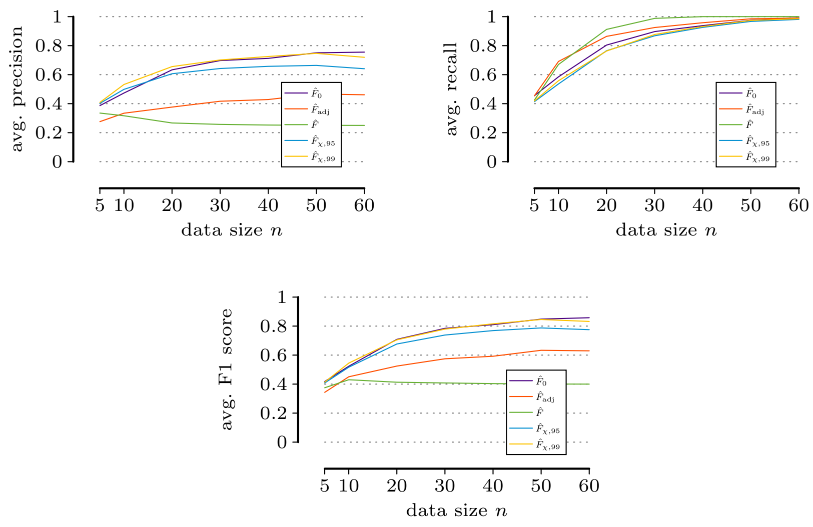


Fig. 5: **Precision, recall, and F1 score for retrieving the minimal top dependency. Top:** Probability tables for a Bayesian network with 5 variables X_1, X_2, X_3, X_4, Y, and only one edge $X \rightarrow Y$. The fraction of information for this dependency is 0.25. **Middle:** Precision and recall of estimators $\tau \in \{\hat{F}_0, \hat{F}_{\mathrm{adj}}, \hat{F}, \hat{F}_{\chi,95}, \hat{F}_{\chi,99}\}$ for retrieving X_4 as the top minimal dependency, averaged over 1000 sampled data for each sample size $n = \{5, 10, 20, 30, 40, 50, 60\}$. **Bottom:** The corresponding F1 score.

6.2 Precision, recall, and F1

Next we evaluate the performance of $\hat{F}_0$ in correctly retrieving the minimal top dependency on synthetic data.

We create a random variable network with input variables $\mathcal{I} = \{X_1, X_2, X_3, X_4\}$ of domain size 3 and a binary target variable Y. The only edge is $X_4 \rightarrow Y$ with the corresponding probability tables shown in Figure 5. Variables $X \in \mathcal{I}$ are uniformly distributed. The minimal top dependency in this network has score $F(X_4; Y) = 0.25$, with all other subsets of $\mathcal{I}$, excluding the supersets of X_4, having a score of 0. We are interested in the problem of retrieving $\mathcal{X}^* = \{X_4\}$ from sampled data as the top dependency. That is, we want the solutions sets to contain X_4, and at the same time be as small in cardinality as possible. An appropriate metric to quantify this is the F1 score, which is a weighted combination of both precision and recall. For example, the top result $\mathcal{X}^* = \{X_1, X_4\}$ of estimator τ has a recall of 1, precision 0.5, and recall 0.66.

Like before, we consider $\tau \in \{\hat{F}_0, \hat{F}_{\text{adj}}, \hat{F}, \hat{F}_{\chi,95}, \hat{F}_{\chi,99}\}$ and samples sizes $n = \{5, 10, 20, 30, 40, 50, 60\}$, and for each n we sample 1000 datasets according to the network. Since the number of attributes is small, we use level-wise exhaustive search. We randomize the order of which candidates are explored in each level to remove any bias introduced from the deterministic order[3]. We plot the average precision, recall, and F1 score, over 1000 data for each estimator and n in Figure 5.

We see that the $\hat{F}_0, \hat{F}_{\chi,95}$, and $\hat{F}_{\chi,99}$, have much better F1 curves than $\hat{F}$ and $\hat{F}_{\text{adj}}$, with those of $\hat{F}_0$ and $\hat{F}_{\chi,99}$ being the best. The plug-in estimator $\hat{F}$ almost always retrieves $\{X_1, X_2, X_3, X_4\}$ as a solution for $n \geq 20$, and hence has very high recall but very small precision. For $n = 5, 10$, the estimate is already 1 before the last level of the search, and hence $\hat{F}$ returns proper subsets of $\mathcal{I}$ resulting in slightly higher precision. In other words, $\hat{F}$ returns arbitrary solutions. The adjusted $\hat{F}_{\text{adj}}$ performs much better than $\hat{F}$, but the large variance does not allow it to compete in terms of precision with the corrected estimators, and hence has much lower F1 across all n.

We observe again that $\hat{F}_0$ shows good performance. In fact, it has a similar F1 curve to that of $\hat{F}_{\chi,99}$ that corresponds to a significance level of 1%. At the same time, Figure 3 suggests that smaller α values for $\hat{F}_{\chi,\alpha}$ can lead to better bias and variance trade-off, but that would harm the F1 score as we can see for $\hat{F}_{\chi,95}$ at 5%. The reliable $\hat{F}_0$ achieves both high F1 and good bias-variance trade-off, without the need of any parameter, and hence is much more suitable for exploratory tasks.

6.3 Optimization performance

We next investigate the optimization performance of the algorithms and bounding functions proposed on real-word data. Our code is available online[4]

We consider datasets from the KEEL data repository [25]. 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 metric attributes are discretized in 5 equal-frequency bins. The datasets are summarized in Table 4. The runtimes are averaged over 3 runs.

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

$$\text{rrd}(A, B) = \frac{(\tau_A - \tau_B)}{\max(\tau_A, \tau_B)} \ ,$$

[3] For example, let us assume that the top score for the first level of the search space, i.e., all singletons, is < 1. The first candidate from the second level is $\{X_1, X_2\}$. It might be that for an estimator τ that $\tau(\{X_1, X_2\}; Y) = \tau(\{X_3, X_4\}; Y) = 1$, and hence the algorithm will return as a solution $\{X_1, X_2\}$ and not $\{X_3, X_4\}$ that contains X_4, resulting in 0 precision and recall instead of 0.5 and 1 respectively. Randomization alleviates this issue.

[4] `https://github.com/pmandros/fodiscovery`

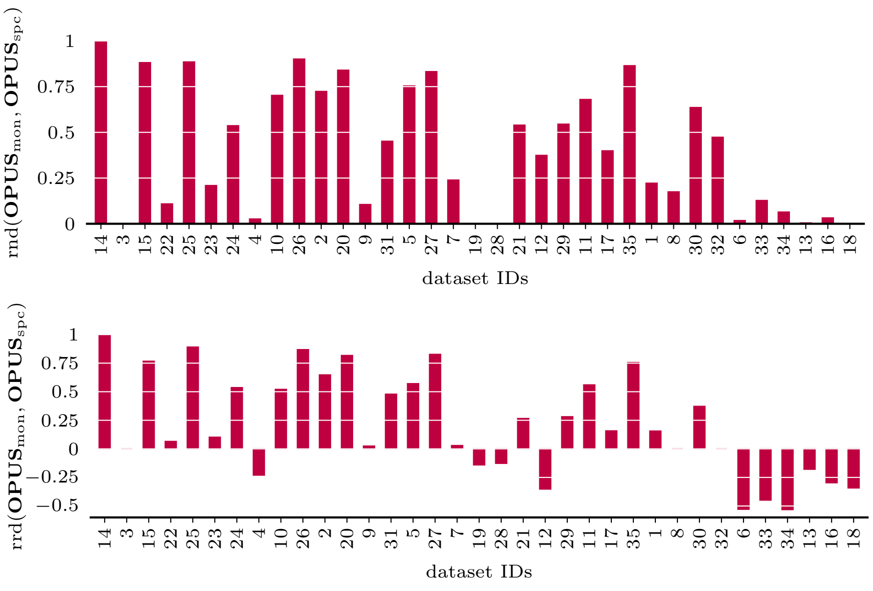


Fig. 6: **Evaluating the branch-and-bound optimization.** Relative nodes explored difference (top) and relative runtime difference (bottom) between methods $\mathbf{OPUS}_{\text{spc}}$ and $\mathbf{OPUS}_{\text{mon}}$. Positive (negative) numbers indicate that $\mathbf{OPUS}_{\text{spc}}$ ($\mathbf{OPUS}_{\text{mon}}$) is proportionally "better". The datasets are sorted in decreasing number of attributes.

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".

6.3.1 Branch-and-bound

We first investigate the performance of the exponential algorithm by comparing $\mathbf{OPUS}_{\text{spc}}$ and $\mathbf{OPUS}_{\text{mon}}$, i.e., Algorithm 1 with $\bar{f}_{\text{spc}}$ and $\bar{f}_{\text{mon}}$ as bounding functions respectively. For a fair comparison, we set a common α value for both methods on each dataset by determining the largest α value in increments of 0.05 such that they terminate in less than 90 minutes. The results can be found in Table 4.

In Figure 6 we present the comparison between $\mathbf{OPUS}_{\text{spc}}$ and $\mathbf{OPUS}_{\text{mon}}$. The top 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

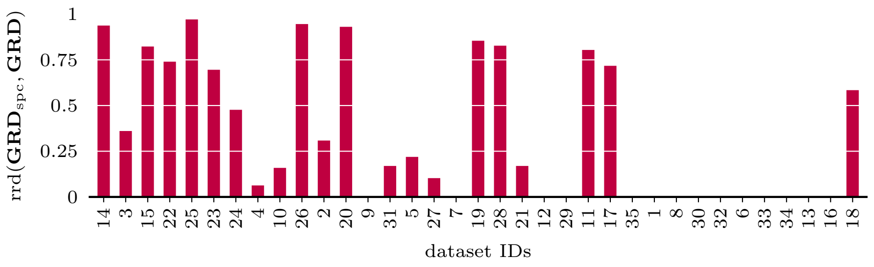


Fig. 7: **Evaluating $\bar{f}_{\mathrm{spc}}$ for heuristic optimization.** Relative time difference between methods $\mathbf{GRD}_{\mathrm{spc}}$ and **GRD**. Positive (negative) numbers indicate that $\mathbf{GRD}_{\mathrm{spc}}$ (**GRD**) is proportionally "better". The datasets are sorted in decreasing number of attributes.

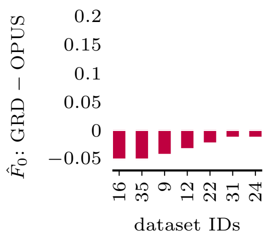
practical difference. The plot validates that the potential for additional pruning is indeed unbounded (Sec. 4). In terms of runtime efficiency (bottom plot), $\mathbf{OPUS}_{\mathrm{spc}}$ is "faster" in 70% of the datasets. In more detail, and considering practical improvements, 12 datasets have at 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 (since 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. We observe in general that both bounding functions, and particularly the $\bar{f}_{\mathrm{spc}}$, make the branch-and-bound search very effective in practice, requiring a couple of minutes on average for termination with good approximation guarantees.

In Table 4 we also show the maximum depth and solution depth for $\mathbf{OPUS}_{\mathrm{spc}}$, i.e., how far in the search space the algorithm had to go and in which level the solution was found. We see that indeed the $\hat{F}_0$ retrieves solutions small in cardinality, 3.6 on average, which is a reasonable number for the size of the data considered. The $\bar{f}_{\mathrm{spc}}$ on the other hand, with 5.9 maximum depth level on average, prunes many of the higher levels of the search space, which explains to a large extend its effectiveness.

6.3.2 Greedy

We now proceed with the evaluation for the heuristic search. We present the relative runtime differences of **GRD** and $\mathbf{GRD}_{\mathrm{spc}}$, i.e., Algorithm 2 with and without $\bar{f}_{\mathrm{spc}}$, in Figure 7 (results in Table 4). While the greedy algorithm is fast, the plot shows that $\bar{f}_{\mathrm{spc}}$ 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. Note that this is possible as we have access to the branch-and-bound results. In Figure 8 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



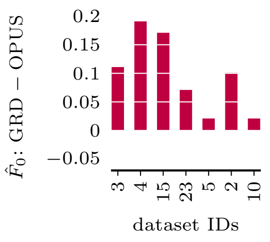


Fig. 8: **Evaluating the heuristic algorithm for result quality. Left**: difference in $\hat{F}_0$ between methods $\mathbf{GRD}_{\text{spc}}$ and $\mathbf{OPUS}_{\text{spc}}$ (i.e., $\hat{F}_0(\mathcal{X}^*_{grd}; Y) - \hat{F}_0(\mathcal{X}^*_{bnb}; Y)$ where $\mathcal{X}^*_{grd}$ and $\mathcal{X}^*_{bnb}$ are the solutions of Algorithm 2 and 1 respectively) for $\alpha = 1$. Since $\alpha = 1$, the negative values close to 0 indicate that Algorithm 2 retrieves nearly optimal solutions. Data are sorted in increasing quality difference. **Right**: difference for $\alpha < 1$. Positive values indicate that Algorithm 2 retrieves better solutions when Algorithm 1 uses guarantees $\alpha < 1$. Data are sorted in increasing α values.

same α values as with the experiments in Sec 6.3.1, 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.

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 Figure 8 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.

In Figure 9 we consider the relative runtime difference between greedy and branch-and-bound, i.e., $\mathbf{GRD}_{\text{spc}}$ and $\mathbf{OPUS}_{\text{spc}}$. As expected, the greedy algorithm is significantly faster in the majority of cases. There are, however, 4 cases where branch-and-bound terminates much faster, which also happen to coincide with more aggressive α values for branch-and-bound.

6.4 Case studies

We close this section with examples of concrete dependencies discovered in two different applications: determining the winner of a tic-tac-toe configuration and predicting the preferred crystal structure of octet binary semi-conductors. Both settings are examples of problems where elementary input features are available, but to correctly represent the input/output relation either non-linear models have to be used or—if interpretable models are sought—complex auxiliary features have to be constructed from the given elementary features.

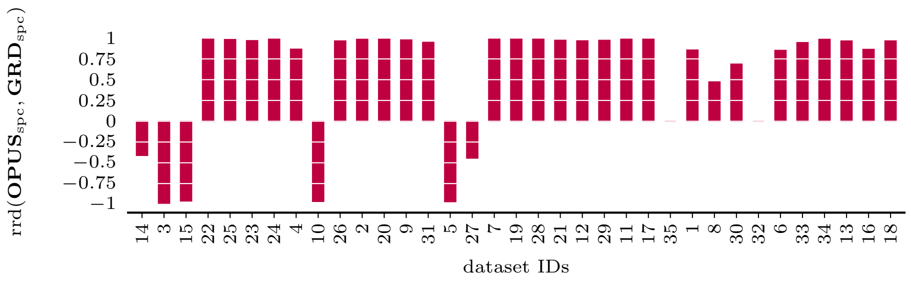


Fig. 9: **Evaluating the heuristic algorithm in terms of running time.** Relative time difference between methods **GRD**$_{\text{spc}}$ and **OPUS**$_{\text{spc}}$. Positive (negative) numbers indicate that **GRD**$_{\text{spc}}$ (**OPUS**$_{\text{spc}}$) is proportionally "better". Datasets are ordered in decreasing number of attributes.

X_1	X_2	X_3
X_4	X_5	X_6
X_7	X_8	X_9

3	2	3
2	4	2
3	2	3

Fig. 10: **Tic-tac-toe example. Left:** Tic-tac-toe board with input variables in corresponding board positions, and variables contained in top dependency marked in red. **Right:** Number of winning combinations each position is involved in.

The game of tic-tac-toe [26] is one of the earliest examples of this complex feature construction problem. Tic-tac-toe is a game of two players where each player picks a symbol from $\{x, o\}$ and, taking turns, marks his symbol in an unoccupied cell of a 3×3 game board. A player wins the game if he marks 3 consecutive cells in a row, column, or diagonal. A game can end in draw, if the board configuration does not allow for any winning move. The dataset consists of 958 end game winning configurations (i.e., there are no draws). The 9 input variables $\mathcal{I} = \{X_1, \ldots, X_9\}$ represent the cells of the board, and can have 3 values $\{x, o, b\}$, where b denotes an empty cell (see Figure 10). The output variable Y with $V(Y) = \{\text{win}, \text{loss}\}$ is the outcome of the game for player x.

Searching for dependencies reveals as top pattern with empirical fraction of information $\hat{F} = 0.61$ and corrected score $\hat{F}_0 = 0.45$ the variable set

$$\mathcal{X} = \{X_1, X_3, X_5, X_7, X_9\}$$

i.e., the four corner cells and the middle one. This is a sensible discovery as these cells correspond exactly to those involved in the highest number of winning combinations (see Figure 10). Knowing the state of these cells

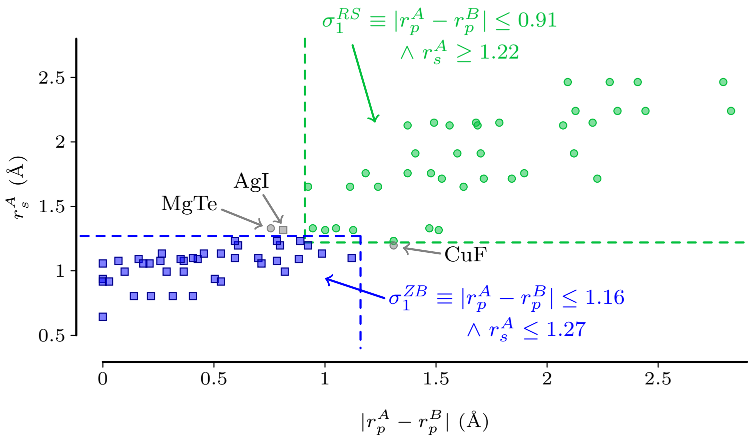


Fig. 11: **Materials Science example.** Binary semiconductors that crystalize as zinkblende (boxes) and rocksalt (circles). Blue and green materials are correctly classified by subgroup-based prediction model—the involved rules (annotated) use elements of the top dependency discovered. (source: [1])

provides, therefore, a high amount of information about the outcome of the game. Moreover, removing a variable results in a loss of a considerable amount of information, while adding a variable would provide more information, but also redundancy. That is, the increase of fraction of information would not be higher than the increase of $\hat{b}_0$.

Our second example is a classical problem from Materials Science [27], which has meanwhile become a canonical example for the challenge of the automatic discovery of interpretable and "physically meaningful" prediction models of material properties [2; 1]. The task is to predict the symmetry or crystal structure in which a given binary compound semi-conductor material will crystalize. That is, each of the 82 material involved consist of two atom types (A and B) and the output variable $Y = \{\text{rocksalt}, \text{zincblende}\}$ describes the crystal structure it prefers energy-wise. The input variables are 14 electro-chemical features of the two atom types considered in isolation: the radii of the three different electron orbitals shapes s, p, and d of atom type A denoted as $r_s(A), r_p(A), r_d(A)$, as well as four important energy quantities that determine its chemical properties (electron affinity, ionization potential, HOMO and LUMO energy levels); the same variables are defined for component B.

For this dataset the top dependency with $\hat{F}_0 = 0.707$ and uncorrected empirical fraction of information $\hat{F} = 0.735$ is

$$\mathcal{X} = \{r_s(A), r_p(A)\}$$

i.e., the atomical s and p radii of component A. Again, this is a sensible finding, since these two variables constitute two out of three variables contained in the best structure prediction model that can be identified using the non-linear subgroup discovery approach [1]. Also both features are involved in the best linear LASSO model based on systematically constructed non-linear combinations of the elementary input variables [2]. The fact that not all variables of those models are identified can likely be explained by the facts that (a) the continuous input variables had to be discretized and (b) the dataset is extremely small with only 82 entries, which renders the discovery of reliable patterns with more than two variables very challenging.

7 Discussion and Conclusions

We considered the problem of measuring and efficiently discovering functional dependencies from data. We adopted an information theoretic approach, and proposed a consistent estimator for mutual information suitable for optimization in high-dimensional data. We proved NP-hardness, and derived two bounding functions for the estimator to be used for pruning. With these, we can efficiently discover the optimal, or α-approximate top-k dependencies for the first time with branch-and-bound. The experimental evaluation shows that the estimator is well-suited for measuring dependencies in exploratory tasks, the bounding functions are very effective for both exhaustive and heuristic algorithms, and that the greedy algorithm provides solutions that are nearly optimal.

While the given reduction from set cover can be extended to show that, unless P=NP, no fully polynomial time approximation scheme exists, the possibility for weaker approximation guarantees remains. In particular, the strong empirical performance of the greedy algorithm hints that $\hat{F}_0$ might have a certain structure favored by the greedy algorithm, e.g., some weaker form of submodularity. For instance, it seems worthwhile to explore ideas from Horel and Singer [28] where a monotone function is e-approximately submodular if it can be bounded by a submodular function within $1 \pm e$. Another idea is that of restricted submodularity for monotone functions [29], where a function is submodular over a subset of the search space. It might be that the greedy algorithm only considers candidates where $\hat{F}_0$ is submodular.

For future work, the proposed bounding functions are likely to be applicable to a larger selection of corrected-for-chance dependency measures, and a general framework for maximizing reliable measures could be established. Additionally, it is also of interest to discover functional dependencies from continuous real-valued data. As entropy has been defined for such data, e.g. differential and cumulative entropy [30], it is possible to instantiate fraction of information scores. The question is, whether we can efficiently correct these scores for chance, and whether optimistic estimators exist that allow for effective search.

References

1. B. R. Goldsmith, M. Boley, J. Vreeken, M. Scheffler, and L. M. Ghiringhelli, "Uncovering structure-property relationships of materials by subgroup discovery," *New Journal of Physics*, vol. 19, 2017.

2. L. M. Ghiringhelli, J. Vybiral, S. V. Levchenko, C. Draxl, and M. Scheffler, "Big data of materials science: Critical role of the descriptor," *Physical review letters*, vol. 114, no. 10, p. 105503, 2015.
3. R. Cavallo and M. Pittarelli, "The theory of probabilistic databases," in *Proceedings of the 13th International Conference on Very Large Data Bases (VLDB), Brighton, UK*, pp. 71–81, 1987.
4. C. Giannella and E. L. Robertson, "On approximation measures for functional dependencies," *Information Systems*, vol. 29, no. 6, pp. 483–507, 2004.
5. M. Reimherr and D. L. Nicolae, "On quantifying dependence: A framework for developing interpretable measures," *Statistical Science*, vol. 28, no. 1, pp. 116–130, 2013.
6. S. Romano, N. X. Vinh, J. Bailey, and K. 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.
7. M. S. Roulston, "Estimating the errors on measured entropy and mutual information," *Physica D: Nonlinear Phenomena*, vol. 125, no. 3, pp. 285–294, 1999.
8. P. Mandros, M. Boley, and J. 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.
9. P. Mandros, M. Boley, and J. Vreeken, "Discovering reliable dependencies from data: Hardness and improved algorithms," in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 317–326, IEEE, 2018.
10. H. Lancaster, *The chi-squared distribution*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics, Wiley, 1969.
11. N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: is a correction for chance necessary?," in *Proceedings of the 26th International Conference on International Conference on Machine Learning*, pp. 1073–1080, ACM, 2009.
12. S. Romano, J. Bailey, N. X. Vinh, and K. Verspoor, "Standardized mutual information for clustering comparisons: One step further in adjustment for chance.," in *Proceedings of the 31st International Conference on Machine Learning (ICML), Beijing, China*, pp. 1143–1151, 2014.
13. N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *Journal of Machine Learning Research*, vol. 11, no. Oct, pp. 2837–2854, 2010.
14. A. Antos and I. Kontoyiannis, "Convergence properties of functional estimates for discrete distributions," *Random Structures & Algorithms*, vol. 19, no. 3-4, pp. 163–193, 2001.
15. T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley-Interscience, 1991.
16. B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*. Springer Publishing Company, Incorporated, 5th ed., 2012.
17. K. Mehlhorn and P. Sanders, *Algorithms and data structures: The basic toolbox*. Springer Science & Business Media, 2008.
18. G. I. Webb, "Opus: An efficient admissible algorithm for unordered search," *Journal of Artificial Intelligence Research*, vol. 3, pp. 431–465, 1995.
19. I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.
20. G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 13, pp. 27–66, Jan. 2012.
21. U. Feige, V. S. Mirrokni, and J. Vondrak, "Maximizing non-monotone submodular functions," *SIAM Journal on Computing*, vol. 40, no. 4, pp. 1133–1153, 2011.
22. A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 1057–1064, 2011.
23. A. A. Bian, J. M. Buhmann, A. Krause, and S. Tschiatschek, "Guarantees for greedy maximization of non-submodular functions with applications," in *International Conference*

on Machine Learning (ICML), 2017.

24. N. X. Vinh, J. Chan, and J. Bailey, “Reconsidering mutual information based feature selection: A statistical significance view,” in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014.
25. J. Alcalà-Fdez, A. Fernàndez, J. Luengo, J. Derrac, and S. Garcìa, “Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework.,” *Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2011.
26. C. J. Matheus and L. A. Rendell, “Constructive induction on decision trees,” in *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI), Detroit, MI*, p. 645, 1989.
27. J. A. Van Vechten, “Quantum dielectric theory of electronegativity in covalent systems. i. electronic dielectric constant,” *Physical Review*, vol. 182, no. 3, p. 891, 1969.
28. T. Horel and Y. Singer, “Maximization of approximately submodular functions,” in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 3045–3053, Curran Associates, Inc., 2016.
29. D.-Z. Du, R. L. Graham, P. M. Pardalos, P.-J. Wan, W. Wu, and W. Zhao, “Analysis of greedy approximations with nonsubmodular potential functions,” in *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’08, pp. 167–175, 2008.
30. M. Rao, Y. Chen, B. C. Vemuri, and F. Wang, “Cumulative residual entropy: A new measure of information,” *IEEE Transactions on Information Technology*, vol. 50, no. 6, pp. 1220–1228, 2004.

						time(s)				nodes explored		$\hat{F}_0$		depth	
ID	dataset	#rows	#attr.	#cl.	α	$\mathbf{OPUS}_{\mathrm{spc}}$	$\mathbf{OPUS}_{\mathrm{mon}}$	$\mathbf{GRD}_{\mathrm{spc}}$	**GRD**	$\mathbf{OPUS}_{\mathrm{spc}}$	$\mathbf{OPUS}_{\mathrm{mon}}$	$\mathbf{OPUS}_{\mathrm{spc}}$	$\mathbf{GRD}_{\mathrm{spc}}$	max	sol.
1	*australian*	690	14	2	1.00	7.0	8.3	1.0	1.0	4190	5388	0.54	0.54	8	4
2	*chess*	3196	36	2	0.75	192.1	545.9	2.5	3.6	69713	252766	0.77	0.87	5	5
3	*coil2000*	9822	85	2	0.05	1.0	1.0	189.1	294.4	86	86	0.06	0.17	1	1
4	*connect-4*	67557	42	3	0.10	1236.8	951.5	164.3	174.8	36183	37176	0.10	0.29	6	4
5	*fars*	100968	29	8	0.65	3.0	7.0	93.9	119.8	45	183	0.66	0.68	2	2
6	*flare*	1066	11	6	1.00	6.8	3.2	1.0	1.0	2011	2048	0.65	0.65	10	3
7	*german*	1000	20	2	1.00	931.5	960.1	1.0	1.0	216250	284397	0.21	0.21	11	6
8	*heart*	270	13	2	1.00	1.9	1.9	1.0	1.0	2275	2758	0.42	0.42	7	4
9	*ionosphere*	351	33	2	1.00	46.4	47.6	1.0	1.0	48094	53784	0.62	0.58	5	3
10	*kddcup*	494020	41	23	0.90	18.1	37.8	520.2	616.4	69	232	0.97	0.99	2	2
11	*letter*	20000	16	26	1.00	659.5	1501.0	3.8	19.1	4894	15300	0.60	0.60	6	5
12	*lymphography*	148	18	4	1.00	31.2	20.2	1.0	1.0	23971	38319	0.48	0.45	10	5
13	*magic*	19020	10	2	1.00	38.5	31.6	1.3	1.3	1012	1017	0.43	0.43	8	5
14	*move-libras*	360	90	15	0.50	1.0	266.6	1.7	25.9	213	163630	0.32	0.32	3	2
15	*optdigits*	5620	64	10	0.35	1.0	4.3	25.1	139.3	105	888	0.36	0.53	2	2
16	*pageblocks*	5472	10	5	1.00	7.4	5.2	1.0	1.0	831	859	0.65	0.60	8	4
17	*penbased*	10992	16	10	1.00	233.6	277.5	1.6	5.6	8099	13486	0.75	0.75	7	4
18	*poker*	1025010	10	10	1.00	2594.7	1705.2	86.0	205.3	908	908	0.57	0.57	7	5
19	*ring*	7400	20	2	1.00	1393.9	1197.3	1.1	7.4	60460	60460	0.29	0.29	6	4
20	*satimage*	6435	36	7	0.80	173.8	954.4	2.0	27.6	24622	154287	0.74	0.74	4	4
21	*segment*	2310	19	7	1.00	39.1	53.3	1.0	1.2	8815	19159	0.84	0.84	9	3
22	*sonar*	208	60	2	1.00	403.5	431.9	1.0	3.8	472554	530478	0.34	0.32	5	3
23	*spambase*	4597	57	2	0.55	515.6	574.6	15.4	50.1	167695	212147	0.54	0.60	7	4
24	*spectfheart*	267	44	2	1.00	171.1	369.3	1.0	1.9	155364	335365	0.23	0.22	5	3
25	*splice*	3190	60	3	0.65	92.3	851.0	1.5	46.9	36052	315125	0.65	0.65	4	4
26	*texture*	5500	40	11	0.80	62.9	480.3	2.1	36.6	10835	109878	0.76	0.76	5	4
27	*thyroid*	7200	21	3	0.50	1.0	5.8	1.8	2.0	247	1475	0.50	0.50	3	3
28	*twonorm*	7400	20	2	1.00	1332.2	1162.4	1.3	7.4	60460	60460	0.42	0.42	6	4
29	*vehicle*	846	18	4	1.00	38.2	53.2	1.0	1.0	10670	23462	0.48	0.48	8	3
30	*vowel*	990	13	11	1.00	3.2	5.1	1.0	1.0	590	1622	0.45	0.45	5	3
31	*wdbc*	569	30	2	1.00	19.9	38.2	1.0	1.2	18564	33862	0.76	0.75	7	3
32	*wine*	178	13	3	1.00	1.0	1.0	1.0	1.0	199	378	0.71	0.71	3	2
33	*wine-red*	1599	11	11	1.00	18.7	10.3	1.0	1.0	1481	1698	0.20	0.20	7	3
34	*wine-white*	4898	11	11	1.00	77.4	36.2	1.0	1.0	1881	2011	0.19	0.19	8	5
35	*zoo*	101	15	7	1.00	1.0	4.1	1.0	1.0	773	5724	0.80	0.75	7	5
avg.						296	360	32	51	41434	78309	0.51	0.53	5.9	3.6

Table 4: Datasets and results from Section 6