Knowledge and Information Systems
https://doi.org/10.1007/s10115-020-01494-9

REGULAR PAPER

Discovering dependencies with reliable mutual information

Panagiotis Mandros[1] · Mario Boley[2] · Jilles Vreeken[3]

Received: 11 March 2019 / Revised: 30 June 2020 / Accepted: 4 July 2020

Abstract
We consider the task of discovering functional dependencies in data for target attributes of interest. To solve it, we have to answer two questions: How do we quantify the dependency in a model-agnostic and interpretable way as well as reliably against sample size and dimensionality biases? How can we efficiently discover the exact or α-approximate top-k dependencies? We address the first question by adopting information-theoretic notions. Specifically, we consider the mutual information score, for which we propose a reliable estimator that enables robust optimization in high-dimensional data. To address the second question, we then systematically explore the algorithmic implications of using this measure for optimization. We show the problem is NP-hard and justify worst-case exponential-time as well as heuristic search methods. We propose two bounding functions for the estimator, which we use as pruning criteria in branch-and-bound search to efficiently mine dependencies with approximation guarantees. Empirical evaluation shows that the derived estimator has desirable statistical properties, the bounding functions lead to effective exact and greedy search algorithms, and when combined, qualitative experiments show the framework indeed discovers highly informative dependencies.

Keywords Information theory · Knowledge discovery · Approximate functional dependency · Pattern mining · Algorithms · Branch-and-bound

1 Introduction

Given data $\mathbf{D}$ from a joint distribution $p(\mathcal{I}, Y)$ over input variables $\mathcal{I} = \{X_1, \ldots, X_d\}$ and a target of interest Y, it is a fundamental problem in knowledge discovery to find subsets $\mathcal{X} \subseteq \mathcal{I}$

✉ Panagiotis Mandros
pmandros@mpi-inf.mpg.de

Mario Boley
mario.boley@monash.edu

Jilles Vreeken
jv@cispa.saarland

[1] Max Planck Institute for Informatics and Saarland University, Saarland Informatics Campus, Saarbrücken, Germany

[2] Monash University, Melbourne, Australia

[3] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Published online: 24 July 2020

that jointly influence or (approximately) determine Y. These dependencies are essential for a variety of applications. In scientific domains, for example, the analysis often involves identifying compact sets of descriptors that capture the underlying process of the various phenomena under investigation [1,2]. The task of **functional dependency discovery** can be formulated as finding the top-k attribute 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 function quantifying the functional dependency of Y on $\mathcal{X}$.

For an effective knowledge discovery procedure, Q should be able to identify any type of dependency, e.g., nonlinear, multivariate, without a priori assumptions on the underlying data generating process p [3]. Moreover, solutions to Eq. (1), besides being efficient, should be exact or come with approximation guarantees. These guarantees can, in particular, verify the absence of meaningful dependencies for Y and prompt the acquisition of new and potentially more relevant features [2]. These two requirements differentiate our task from similar and well-known applications such as key discovery for data management [4], feature selection in machine learning [5], and Markov blanket discovery in Bayesian networks [6]. The former operates under a closed world assumption and hence the discovered keys will not generalize to unseen data drawn from the same distribution p. Unlike feature selection where the end-user is a machine learning algorithm, we are interested in providing the analyst with sparse but exact solutions where all interactions are accounted for, rather than high-dimensional, greedy solutions for pairwise associations (see [7] for a survey on scores Q designed for feature selection). Lastly, Markov blanket algorithms[1] often operate under the assumption that p is faithfully represented by a DAG, implying a unique Markov blanket. Additionally, high-order dependencies (e.g., $Y = X \oplus Z$) are neglected due to the greedy search employed. The variant we consider does not impose a DAG structure for p, nor faithfulness, and is therefore better suited for exploratory analysis and high-order dependencies.

Given categorical data $\mathbf{D}$, the ideal choice for Q is the information-theoretic measure **fraction of information** [9–11], 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 **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 about Y after observing $\mathcal{X}$. The fraction of information then represents the proportional reduction in uncertainty about Y by knowing $\mathcal{X}$. Moreover, the extreme values $F(\mathcal{X}; Y) = 1$ and $F(\mathcal{X}; Y) = 0$ correspond to functional dependency and statistical independence, respectively.

Estimating the mutual information $I(\mathcal{X}; Y)$ naively with empirical probabilities, however, can lead to an overestimation of the true dependency between $\mathcal{X}$ and Y—a behavior known as *dependency-by-chance* [12]. While asymptotically efficient [13], the empirical estimator $\hat{I}(\mathcal{X}; Y)$ has an increasing bias with the domain size of variables [14], and hence, is unsuited for dependency discovery where we have to soundly compare different variable sets of varying dimensionality and consequently of widely varying domain sizes. It is even possible that a dependency is indicated, when $\mathcal{X}$ and Y are actually independent in p (see Fig. 1). As a result,

[1] Established Markov blanket algorithms, e.g., IAMB [8], or the more specialized PCMB [6], are primarily based on greedy search and independence tests, combined with mutual information to rank candidates. To find all possible Markov blankets, there are extensions based on the random greedy algorithm and repeated algorithm executions [6].

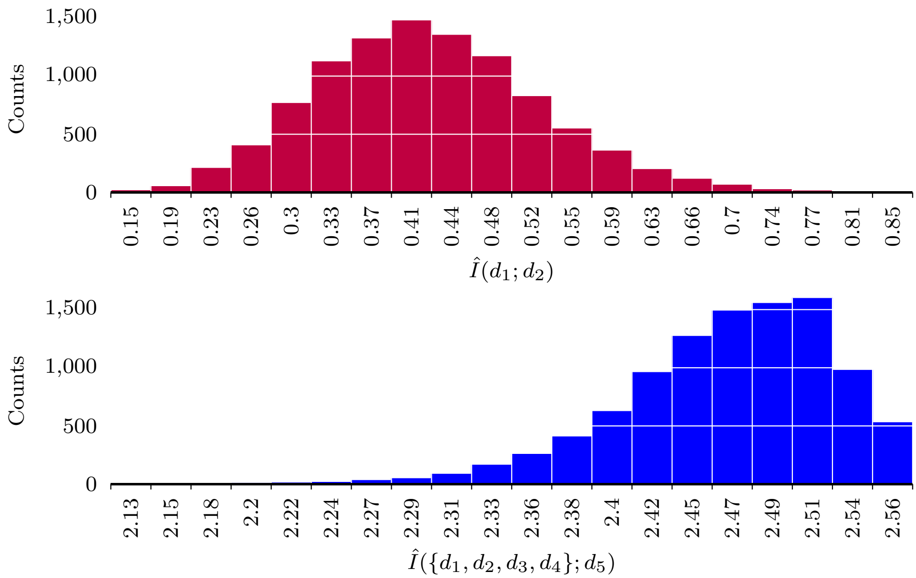


Fig. 1 Histogram of plug-in mutual information estimates $\hat{I}$ for independent dice rolls. Top: For a pair of dice d_1, d_2, we perform 50 independent rolls and compute $\hat{I}(d_1; d_2)$. We repeat this process with 10, 000 simulations and plot the histogram for 20 equal-frequency bins. Despite having a population value of $I(d_1; d_2) = 0$, the histogram has a right-tailed bell shape with expected value $\mathbb{E}[\hat{I}(d1; d2)] \approx 0.44$. Bottom: Same procedure but with 5 dice. Here the histogram has a left tail, with $\mathbb{E}[\hat{I}(\{d_1, d_2, d_3, d_4\}; d_5)] \approx 2.48$

$\hat{I}(\mathcal{I}; Y)$ is a trivial and uninformative maximizer for Eq. (1). To the best of our knowledge, there are no exact algorithms that efficiently solve Eq. (1) incorporating more refined mutual information estimators.

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 nonparametric *permutation model* [15, p. 214], under which the expected value is computed as the average $\hat{I}(\mathcal{X}; Y_\sigma)$ over all sample permutations σ. 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 greedy search to heavily prune the search space.

In this article we build upon and extend our recent work published as Mandros et al. [16,17]. In the former paper, we introduced the general problem, a corrected estimator, and a bounding function that allows branch-and-bound search for the strongest dependencies. In the second paper, we focused on the discovery aspect, proved NP-hardness, proposed a tighter bounding function, and investigated better algorithms for both exact and heuristic search. 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 make the link to nonparametric permutation tests. We show that in our setting mutual information is not submodular, which excludes approximation guarantees of greedy optimization for this function. Last, we extend the evaluation and include comparisons with a corrected estimator using a parametric null model, and by performing bias, variance, and precision/recall experiments.

Our overall contributions are the following. We derive a consistent and reliable estimator for mutual information to correct the positive bias, as well as accompany the estimator with a set of useful properties that can be used for optimization (Sect. 2). We then study the algorithmic aspects, show that maximizing the reliable estimator is NP-hard (Sect. 3), and derive two effective bounding functions that can be used by algorithms to prune the search space (Sect. 4). The first function is applicable to any estimator having a monotonically increasing (w.r.t. the superset relation) correction term, while the second is specifically tailored to our proposed reliable estimator and has an unbounded pruning potential over the first. We propose an admissible branch-and-bound algorithm to discover the α-approximate top dependencies for desired approximation guarantee $\alpha \in (0, 1]$, and in addition, a fast greedy algorithm (Sect. 5). Last, we perform an extensive evaluation for the estimator, pruning functions, and resulting discovery framework (Sect. 6). The experiments demonstrate an excellent performance for the greedy algorithm combined with $\hat{I}_0$ (a non-monotonic, non-submodular set function), with optimal or nearly optimal results in all 35 real-world datasets under investigation.

2 Reliable mutual information and properties

In this section we derive our estimator for mutual information, as well as 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, $\hat{p}_{Z\,|\,X=x}\colon\ V(Z) \to [0, 1]$ 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, because they arise from simply "plugging in" the empirical distribution $\hat{p}$ instead of p.

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 [14], where

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

We see that the bias is independent of the actual distribution p and it depends solely on the domain sizes and the number of samples n. The bias is high when the $\mathcal{X}$, Y, samples produce jointly 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 $p(\mathcal{X}, Y) = p(\mathcal{X})p(Y)$, and hence $I(\mathcal{X}; Y) = 0$.

These last observations suggest a correction for the empirical $\hat{I}(\mathcal{X}; Y)$ by subtracting its bias assuming independence for $\mathcal{X}$ and Y. A nonparametric choice for the null model is the **permutation model** [15, p. 214], arriving at the bias $\mathbb{E}[\hat{I}(\mathcal{X}; Y) - I(\mathcal{X}; Y) \mid I(\mathcal{X}; Y) = 0]$ expressed as 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

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

The reliable estimator $\hat{I}_0$ controls the number of false positives by being unbiased under the null hypothesis with fixed marginal counts. In relation to statistical hypothesis testing and permutation tests, here we subtract the expected value of the null distribution instead of finding the exact probability of the tail. Our approach is more flexible as it does not require a fixed confidence interval, but instead it adapts to the data and the different dimensionalities encountered during search[2]. Moreover, we will see below that computing the mean is much more efficient than enumerating all possible permuted datasets to obtain the exact probability (which is only applicable for small data). Intuitively, $\hat{I}_0$ works in the following way: when it appears in a sample that $\hat{I}(\mathcal{X}; Y)$ 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 correction is large as well. 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 $b_0(X, Y, n) = \mathbb{E}_0[\hat{F}(X; Y)] = m_0(X, Y, n)/\hat{H}(Y)$.

[2] Normally in such cases one would consider multiple hypothesis testing to control the family-wise error rate, e.g., Bonferroni correction. Our approach does not depend on the number of hypotheses, with the level of conservatism controlled by the domain size of variables. This means that it does not become unnecessarily strict for large hypothesis spaces, e.g., a high-dimensional dataset, and better adapts to the data at hand.

Regarding the evaluation of Eq. (2), a naive approach with $n!$ possible permutations is computationally infeasible. However, Vinh et al. [18] 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 nonzero 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 $O(n \max\{|V(\mathcal{X})|, |V(Y)|\})$ [19]. 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. [20] show that $\lim_{n \to \infty} m_0(\mathcal{X}, Y, n) = 0$, and together with the consistency of the plug-in $\hat{F}$ [13], 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[3] as the empirical domain

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

$V(\mathcal{X})$ 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 nonnegative [21, 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. □

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 $\pi : V(Z) \to V(X)$, 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) $\hat{H}(Y \mid Z) \leq \hat{H}(Y \mid X)$
(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

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

$$\leq - \sum_{z \in \pi^{-1}(x)} q(z) \log q(z) = \sum_{z \in \pi^{-1}(x)} h(z) \ ,$$

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 [21, p. 31]: for nonnegative 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

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

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). □

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 Table 1 for examples of both introduced relations. Importantly, the inequality of mutual information for specializations (Proposition 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

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

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

$$m_0(Z, Y, n) = \frac{1}{n!} \sum_{\sigma \in S_n} \hat{I}(Z_\sigma; Y)$$

Table 1 Specialization and homomorphism examples

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

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 $Y \equiv Y_\sigma$

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

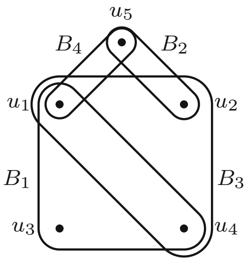
□

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 [22, 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 parts, i.e., $Y\colon\ [l] \to \{\text{a, b, c}\}$ with



		$\mathbf{X_1}$	$\mathbf{X_2}$	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

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)

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

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

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

See Fig. 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 a for (a, ..., 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

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

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 $(c, \dots, c)$. 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-$(c, \dots, c)$ values of $\mathcal{C}$ in S_3 to indices of non-$(c, \dots, c)$ 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 ([20], 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 1(a) and 1(b). 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 1(c), and thus $b_0(\mathcal{I}(\mathcal{C}'),Y,kl) \leq b_0(\mathcal{I}(\mathcal{C}),Y,kl)$ from Proposition 2(b)—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 $O(l^2m)$ (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 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. This optimistic estimator is both inexpensive[4], and applicable to any estimator that has a monotonically increasing correction term. 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

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

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

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

since $\hat{H}(R^+, Y) = \hat{H}(R \cup Y, Y) = \hat{H}(R, Y)$, and from Proposition 1(b), 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 1(b), the second from the positive gain of $\mathcal{Z}^+$ over $\mathcal{Z}$.

[4] one can cache the $b_0(\mathcal{X}, Y, n)$ term for Eq. (5) while computing $\hat{F}_0(\mathcal{X}; Y)$ for a $\mathcal{X} \subseteq \mathcal{I}$ during search

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 $|V(\mathcal{X}^+)| \leq |V(\mathcal{X})||V(Y)|$. □

In a nutshell, the $(\cdot)^+$ can only increase the $\hat{F}_0$ value, and $\mathcal{X}^+$ 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 nonzero 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 $\Delta = \bar{f}_{mon}(\mathcal{X}) - \bar{f}_{spc}(\mathcal{X})$. 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. $\Delta \in \Omega(1 - \frac{1}{\log 2l})$*

Proof (a)

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

where the inequality holds from Proposition 1(b) 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

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

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.

Regarding their applicability to other mutual information estimators, $\bar{f}_{\text{mon}}$ only requires monotonicity for the correction term, while $\bar{f}_{\text{spc}}$ additionally needs a positive gain w.r.t. to the $(\cdot)^+$ operation. The former is easier to satisfy. Computationally, $\bar{f}_{\text{spc}}(\mathcal{X})$ is more expensive than $\bar{f}_{\text{mon}}(\mathcal{X})$ by a factor of $|V(Y)|$. In practice one can combine both optimistic estimators

Table 2 Construction showing the advantage of bound $\bar{f}_{\text{spc}}$ versus $\bar{f}_{\text{mon}}$

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
⋮			
⋮			
a	2l-1	(a,2l-1)	b
b	2l-1	(b,2l-1)	b
a	2l	(a,2l)	b
b	2l	(b,2l)	b

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)

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 $f(\mathcal{X}^*) \geq \alpha \max\{f(\mathcal{X}') \colon \mathcal{X}' \subseteq \mathcal{I}\}$

```
function OPUS($\mathbf{Q}, \mathcal{S}$)
    if $\mathbf{Q}$ is empty then
        return $\mathcal{S}$
    else
        $(\mathcal{X}, \mathcal{Z}) = pop(\mathbf{Q})$
        $\mathbf{R} = \{(\mathcal{X} \cup \{Z\}, Z) \colon Z \in \mathcal{Z}\}$
        $\mathcal{X}^* = \arg\max\{f(\mathcal{X}') \colon \mathcal{X}' \in \mathbf{R} \cup \{\mathcal{S}\}\}$
        $\mathbf{R}' = \{(\mathcal{X}', Z) \in \mathbf{R} \colon \alpha \bar{f}(\mathcal{X}') > f(\mathcal{X}^*)\}$
        $\mathcal{Z}' = \{Z \colon (\mathcal{X}', Z) \in \mathbf{R}'\}$
        $[(\mathcal{X}_1, Z_1), \ldots, (\mathcal{X}_k, Z_k)] = sort(\mathbf{R}')$
        $\mathbf{Q}' = \mathbf{Q} \cup \{(\mathcal{X}_i, \mathcal{Z}' \setminus \{Z_1, \ldots, Z_i\}) \colon i \in [k])\}$
        return OPUS($\mathbf{Q}', \mathcal{X}^*$)
$\mathcal{X}^* = $ OPUS($\{(\emptyset, \mathcal{I})\}, \emptyset$)
```

in a *chain-like manner*: 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 Sect. 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.

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($\emptyset, \emptyset$)
```

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 function at hand (see, e.g., [23, 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 [24]. 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 [24]), 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 [5,7]). 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 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).

Table 3 Example data demonstrating non-submodularity of I, $\hat{I}$, $\hat{I}_0$ in our supervised scenario where the target Y is fixed (Proposition 5)

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

Concerning the approximation ratio of the greedy algorithm, there exists a large amount of research focused on submodular and/or monotone functions, e.g., [25–27]. Recall that for a set $\mathcal{I} = \{X_1, \ldots, X_d\}$, a function $f : 2^{\mathcal{I}} \to \mathbb{R}$ 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 $I(.; Y)$, the plug-in $\hat{I}(.; Y)$, and corrected $\hat{I}_0(.; Y)$ are not submodular w.r.t. the first argument.*

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 $p = \hat{p}$, it is straightforward to show that I is also not submodular.

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 Sect. 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, and the performance of the bounding functions for both branch-and-bound and greedy search. We additionally perform qualitative experiments with two case studies.

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 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. [28]. 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. [12]. 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 $n \in \{5, 10, 20, 30, 40, 50, 60\}$.[5] 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 Fig. 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 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

[5] The α values in $\hat{F}_{\chi,\alpha}$ are chosen according to Vinh et al. [28]

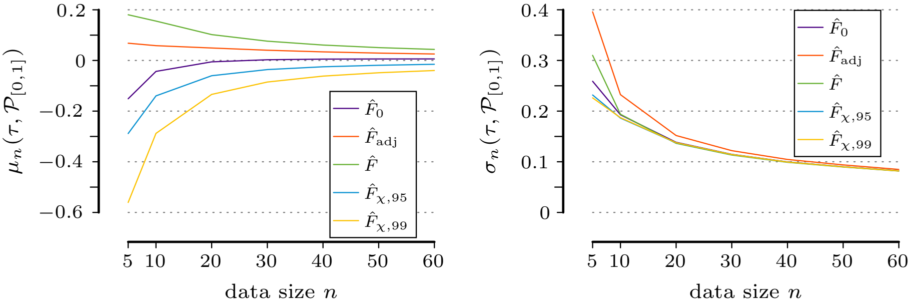


Fig. 3 Empirical bias and standard deviation of estimators averaged over $p \in \mathcal{P}_{[0,1]}$. Average bias $\mu_n(\tau, \mathcal{P}_{[0,1]})$ (left) and standard deviation $\sigma_n(\tau, \mathcal{P}_{[0,1]})$ (right) of estimators $\tau \in \{\hat{F}_0, \hat{F}_{\text{adj}}, \hat{F}, \hat{F}_{\chi,95},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

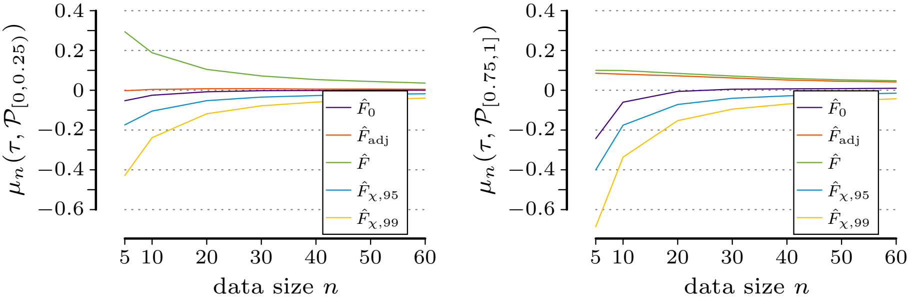


Fig. 4 Bias of estimators averaged over $p \in \mathcal{P}_{[0,0.25)}$ and $p \in \mathcal{P}_{[0.75,1]}$. Average bias $\mu_n(\tau, \mathcal{P}_{[0,0.25)})$ (left) and $\mu_n(\tau, \mathcal{P}_{[0.75,1]})$ (right) of estimators $\tau \in \{\hat{F}_0, \hat{F}_{\text{adj}}, \hat{F}, \hat{F}_{\chi,95},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

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 Fig. 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 Fig. 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

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

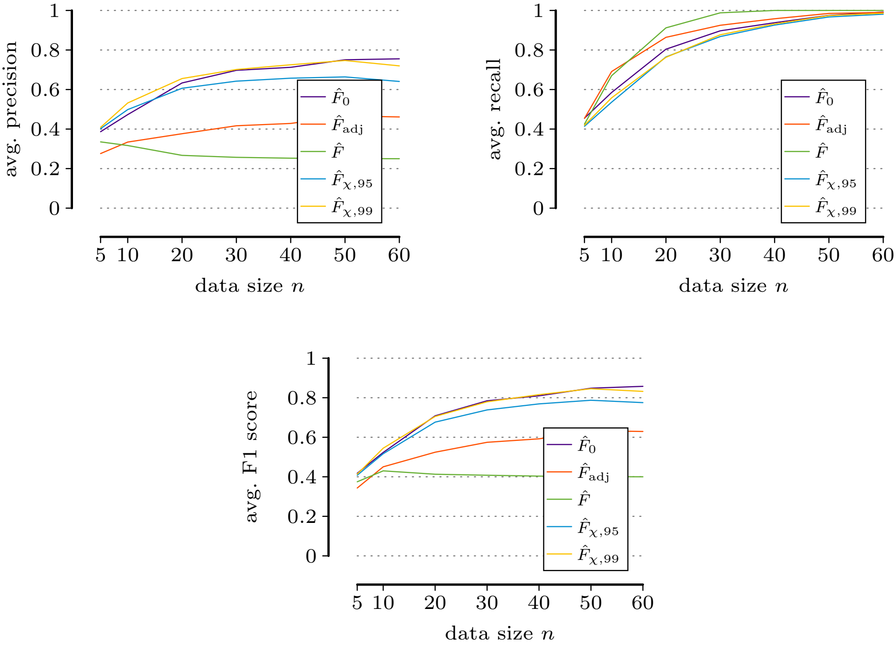


Fig. 5 Precision, recall, and F1 score for retrieving the minimal top dependency. Top: Probability tables for a Bayesian network with 5 variables X_1, X_2, X_3, X_4, Y, and only one edge $X \rightarrow Y$. The fraction of information for this dependency is 0.25. Middle: Precision and recall of estimators $\tau \in \{\hat{F}_0, \hat{F}_{\mathrm{adj}}, \hat{F}, \hat{F}_{\chi,95}, \hat{F}_{\chi,99}\}$ 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

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 $\mathbb{E}_0[\hat{I}]$.

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 Bayesian 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 Fig. 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[6]. We plot the average precision, recall, and F1 score, over 1000 data for each estimator and n in Fig. 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, Fig. 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-world data. Our code is available online.[7]

We consider datasets from the KEEL data repository [29]. 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

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

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,

[6] 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.

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

Table 4 Datasets and results from Sect. 6

ID	dataset	#rows	#attr.	#cl.	α	Time(s)				Nodes explored		$\hat{F}_0$		Depth	
						$\mathbf{OPUS}_{\text{spc}}$	$\mathbf{OPUS}_{\text{mon}}$	$\mathbf{GRD}_{\text{spc}}$	**GRD**	$\mathbf{OPUS}_{\text{spc}}$	$\mathbf{OPUS}_{\text{mon}}$	$\mathbf{OPUS}_{\text{spc}}$	$\mathbf{GRD}_{\text{spc}}$	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	69,713	252,766	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*	67,557	42	3	0.10	1236.8	951.5	164.3	174.8	36,183	37,176	0.10	0.29	6	4
5	*fars*	100,968	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	216,250	284,397	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	48,094	53,784	0.62	0.58	5	3
10	*kddcup*	494,020	41	23	0.90	18.1	37.8	520.2	616.4	69	232	0.97	0.99	2	2
11	*letter*	20,000	16	26	1.00	659.5	1501.0	3.8	19.1	4894	15,300	0.60	0.60	6	5
12	*lymphography*	148	18	4	1.00	31.2	20.2	1.0	1.0	23,971	38,319	0.48	0.45	10	5
13	*magic*	19,020	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	163,630	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*	10,992	16	10	1.00	233.6	277.5	1.6	5.6	8099	13,486	0.75	0.75	7	4

Table 4 continued

ID	dataset	#rows	#attr.	#cl.	α	Time(s)				Nodes explored		$\hat{F}_0$		Depth	
						OPUS$_{\text{spc}}$	**OPUS**$_{\text{mon}}$	**GRD**$_{\text{spc}}$	**GRD**	**OPUS**$_{\text{spc}}$	**OPUS**$_{\text{mon}}$	**OPUS**$_{\text{spc}}$	**GRD**$_{\text{spc}}$	Max	Sol.
18	*poker*	1,025,010	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	60,460	60,460	0.29	0.29	6	4
20	*satimage*	6435	36	7	0.80	173.8	954.4	2.0	27.6	24,622	154,287	0.74	0.74	4	4
21	*segment*	2310	19	7	1.00	39.1	53.3	1.0	1.2	8815	19,159	0.84	0.84	9	3
22	*sonar*	208	60	2	1.00	403.5	431.9	1.0	3.8	472,554	530,478	0.34	0.32	5	3
23	*spambase*	4597	57	2	0.55	515.6	574.6	15.4	50.1	167,695	212,147	0.54	0.60	7	4
24	*spectfheart*	267	44	2	1.00	171.1	369.3	1.0	1.9	155,364	335,365	0.23	0.22	5	3
25	*splice*	3190	60	3	0.65	92.3	851.0	1.5	46.9	36,052	315,125	0.65	0.65	4	4
26	*texture*	5500	40	11	0.80	62.9	480.3	2.1	36.6	10,835	109,878	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	60,460	60,460	0.42	0.42	6	4
29	*vehicle*	846	18	4	1.00	38.2	53.2	1.0	1.0	10,670	23,462	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	18,564	33,862	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	78,309	0.51	0.53	5.9	3.6

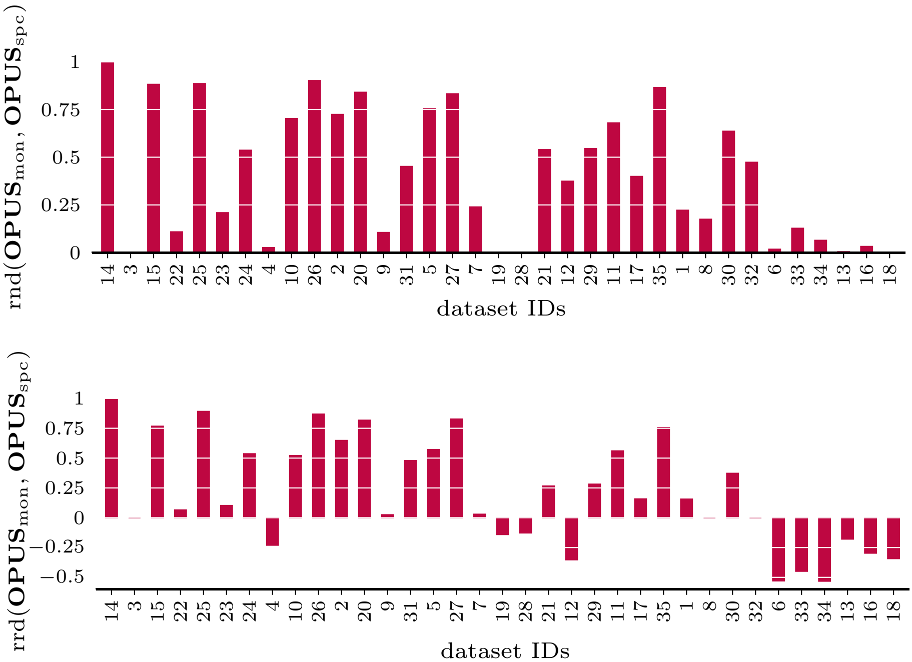


Fig. 6 Evaluating the branch-and-bound optimization. Relative nodes explored difference (top) and relative runtime difference (bottom) between methods $\mathbf{OPUS}_{\text{spc}}$ and $\mathbf{OPUS}_{\text{mon}}$. Positive (negative) numbers indicate that $\mathbf{OPUS}_{\text{spc}}$ ($\mathbf{OPUS}_{\text{mon}}$)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

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., Alg. 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 are in Table 4.

In Fig. 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 in 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 (Sect. 4). In terms of runtime efficiency (bottom plot), $\mathbf{OPUS}_{\text{spc}}$ 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 (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

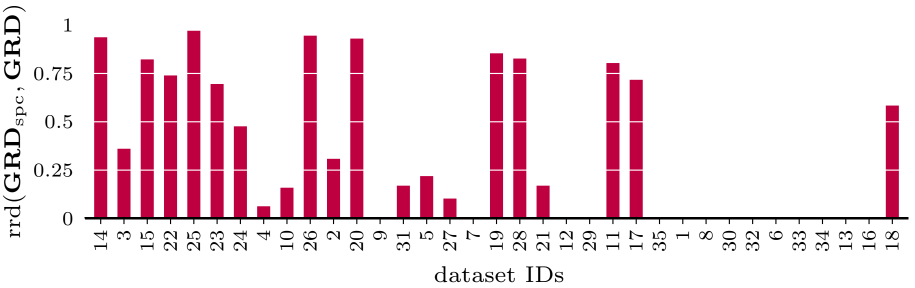


Fig. 7 Evaluating $\bar{f}_{\mathrm{spc}}$ for heuristic optimization. Relative time difference between methods $\mathbf{GRD}_{\mathrm{spc}}$ and **GRD**. Positive (negative) numbers indicate that $\mathbf{GRD}_{\mathrm{spc}}$ (**GRD**) is proportionally "better". The datasets are sorted in decreasing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

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 Fig. 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 Fig. 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 same α values as with the experiments in Sec 6.3.1, and that we only plot the nonzero 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 Fig. 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 Fig. 9 we consider the relative runtime difference between greedy and branch-and-bound, i.e., $\mathbf{GRD}_{\mathrm{spc}}$ and $\mathbf{OPUS}_{\mathrm{spc}}$. As expected, the greedy algorithm is significantly faster in the majority of cases. There are, however, 4 cases where branch-and-bound terminates

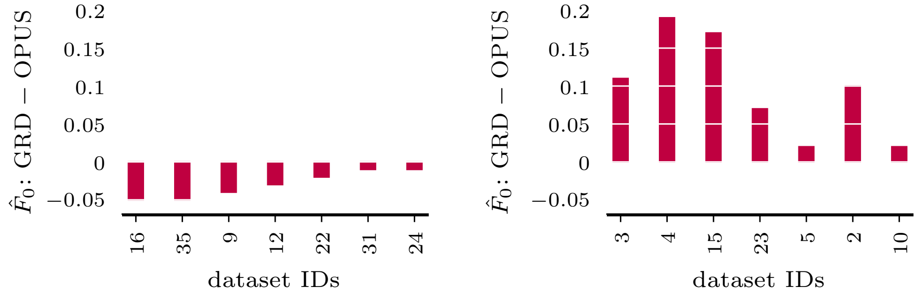


Fig. 8 Evaluating the heuristic algorithm for result quality. Left: difference in $\hat{F}_0$ between methods $\mathbf{GRD}_{\text{spc}}$ and $\mathbf{OPUS}_{\text{spc}}$ (i.e., $\hat{F}_0(\mathcal{X}^*_{grd}; Y) - \hat{F}_0(\mathcal{X}^*_{bnb}; Y)$ where $\mathcal{X}^*_{grd}$ and $\mathcal{X}^*_{bnb}$ 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

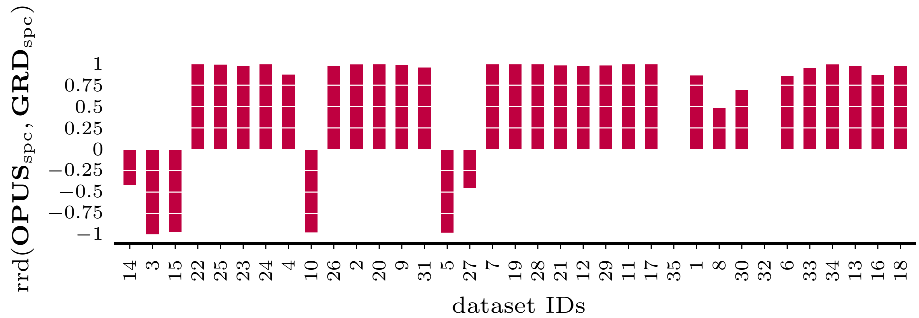


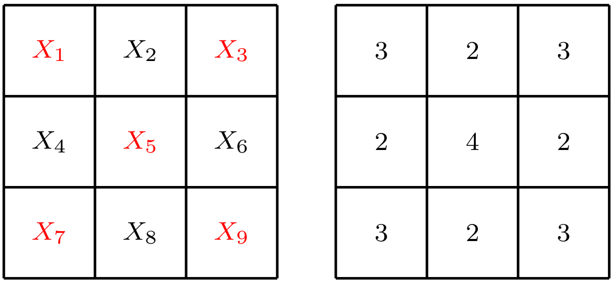
Fig. 9 Evaluating the heuristic algorithm in terms of running time. Relative time difference between methods $\mathbf{GRD}_{\text{spc}}$ and $\mathbf{OPUS}_{\text{spc}}$. Positive (negative) numbers indicate that $\mathbf{GRD}_{\text{spc}}$ ($\mathbf{OPUS}_{\text{spc}}$) is proportionally "better". Datasets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

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 nonlinear models have to be used or—if interpretable models are sought—complex auxiliary features have to be constructed from the given elementary features.

The game of tic-tac-toe [30] 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



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: 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

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 Fig. 10). The output variable Y with $V(Y) = \{\text{win, loss}\}$ 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 Fig. 10). Moreover, removing a variable results in the 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 [31], which has meanwhile become a canonical example for the challenge of the automatic discovery of interpretable and “physically meaningful” prediction models of material properties [1,2]. 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 $Y = \{\text{rocksalt, zincblende}\}$ 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 nonlinear subgroup discovery approach [1] (see Fig. 11). Also both features are involved in the best linear LASSO model based on systematically constructed nonlinear 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

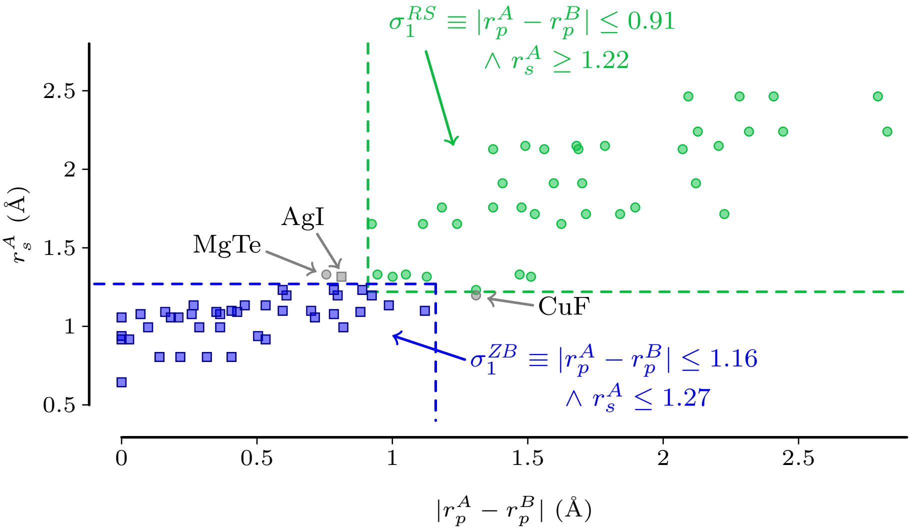


Fig. 11 Materials Science example. Binary semiconductors that crystalize as zinkblende (boxes) and rocksalt (circles). Blue and green materials are correctly classified by subgroup-based prediction model—the involved rules (annotated) use elements of the top dependency discovered. *Source*: Goldsmith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

only 82 entries, which renders the discovery of reliable patterns with more than two variables very challenging.

7 Discussion and Conclusions

We considered the dual problem of measuring and efficiently discovering functional dependencies from data. For a model-agnostic and interpretable knowledge discovery procedure, we adopted an information-theoretic approach and proposed a consistent and robust estimator for mutual information suitable for optimization in high-dimensional data. We proved the NP-hardness of the problem, and derived two bounding functions for the estimator that can be used to prune the search space. With these, we can effectively discover the optimal, or α-approximate top-k dependencies with branch-and-bound. The experimental evaluation showed that the estimator has desired statistical properties, the bounding functions are very effective for both exhaustive and heuristic algorithms, and the greedy algorithm provides solutions that are nearly optimal. Qualitative experiments on two case studies indicate that our proposed framework indeed discovers informative dependencies.

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 $\hat{F}_0$ could have a certain structure favored by the greedy algorithm, e.g., some weaker form of submodularity (we remind that $\hat{F}_0$ is neither submodular nor monotone). For instance, one could explore ideas from Horel and Singer [32] where a monotone function is ϵ-approximately submodular if it can be bounded by a submodular function within $1 \pm \epsilon$. Another idea is that of restricted submodularity for monotone functions [33], where a function is submodular over a subset of the search space. One can also explore the submodularity index for general set functions [34], where a proxy for the degree of non-submodularity is incorporated in the approximation guarantee.

For future work, the proposed bounding functions are likely to be applicable to a larger selection of corrected-for-chance dependency measures. For example, the monotonicity-based bounding function only requires a correction term that is monotonically increasing with the superset relation. Additionally, it is also of interest to discover functional dependencies given continuous data. As entropy has been defined for such data, e.g., differential and cumulative entropy [35], it is possible to instantiate fraction of information scores. The question would be in what way these measures can be corrected for chance, and whether optimistic estimators exist that allow for efficient and exact optimization.

Acknowledgements Open Access funding provided by Projekt DEAL.

References

1. Goldsmith BR, Boley M, Vreeken J, Scheffler M, Ghiringhelli LM (2017) Uncovering structure-property relationships of materials by subgroup discovery. New J Phys 19:013031
2. Ghiringhelli LM, Vybiral J, Levchenko SV, Draxl C, Scheffler M (2015) Big data of materials science: Critical role of the descriptor. Phys Rev Lett 114(10):105503
3. Reshef DN, Reshef YA, Finucane HK, Grossman SR, McVean G, Turnbaugh PJ, Lander ES, Mitzenmacher M, Sabeti PC (2011) Detecting novel associations in large data sets. Science 334:1518–1524
4. Papenbrock T, Ehrlich J, Marten J, Neubert T, Rudolph J-P, Schönberg M, Zwiener J, Naumann F (2015) Functional dependency discovery: An experimental evaluation of seven algorithms. Proc VLDB Endowm 8(10):1082–1093
5. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
6. Peña JM, Nilsson R, Björkegren J, Tegnér J (2007) Towards scalable and data efficient learning of Markov boundaries. Int J Approx Reason 45:211–232
7. Brown G, Pocock A, Zhao M-J, Luján M (2012) Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. J Mach Learn Res 13:27–66
8. Tsamardinos I, Aliferis C, Statnikov A, Statnikov E (2003) Algorithms for large scale markov blanket discovery. In: Proceedings of the 16th international FLAIRS conference, St, AAAI Press, pp 376–380
9. Cavallo R, Pittarelli M (1987) The theory of probabilistic databases. In: Proceedings of the 13th international conference on very large data bases (VLDB), Brighton, UK, pp 71–81
10. Giannella C, Robertson EL (2004) On approximation measures for functional dependencies. Inf Syst 29(6):483–507
11. Reimherr M, Nicolae DL (2013) On quantifying dependence: a framework for developing interpretable measures. Stat Sci 28(1):116–130
12. Romano S, Vinh NX, Bailey J, Verspoor K (2016) A framework to adjust dependency measure estimates for chance. In: Proceedings of the SIAM international conference on data mining (SDM), Miami, FL, SIAM
13. Antos A, Kontoyiannis I (2001) Convergence properties of functional estimates for discrete distributions. Random Struct Algorithm 19:163–193
14. Roulston MS (1999) Estimating the errors on measured entropy and mutual information. Physica D 125(3):285–294
15. Lancaster H (1969) *The chi-squared distribution*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics, Wiley

16. Mandros P, Boley M, Vreeken J (2017) Discovering reliable approximate functional dependencies. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, KDD'17, ACM
17. Mandros P, Boley M, Vreeken J (2018) Discovering reliable dependencies from data: hardness and improved algorithms. In: 2018 IEEE international conference on data mining (ICDM), pp 317–326, IEEE
18. Vinh NX, Epps J, Bailey J (2009) Information theoretic measures for clusterings comparison: is a correction for chance necessary?. In: Proceedings of the 26th international conference on machine learning, ACM, pp 1073–1080
19. Romano S, Bailey J, Vinh NX, Verspoor K (2014) 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
20. Vinh NX, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J Mach Learn Res 11:2837–2854
21. Cover TM, Thomas JA (1991) Elements of information theory. Wiley-Interscience, New York, NY
22. Korte B, Vygen J (2012) Combinatorial optimization: theory and algorithms, 5th edn. Springer Publishing Company, Berlin
23. Mehlhorn K, Sanders P (2008) Algorithms and data structures: the basic toolbox. Springer Science & Business Media, Berlin
24. Webb GI (1995) Opus: an efficient admissible algorithm for unordered search. J Artif Intell Res 3:431–465
25. Feige U, Mirrokni VS, Vondrak J (2011) Maximizing non-monotone submodular functions. SIAM J Comput 40(4):1133–1153
26. Das A, Kempe D (2011) 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
27. Bian AA, Buhmann JM, Krause A, Tschiatschek S (2017) Guarantees for greedy maximization of non-submodular functions with applications. In: International conference on machine learning (ICML)
28. Vinh NX, Chan J, Bailey J (2014) Reconsidering mutual information based feature selection: a statistical significance view. In: Proceedings of the 28th AAAI conference on artificial intelligence
29. Alcalà-Fdez J, Fernàndez A, Luengo J, Derrac J, Garcìa S (2011) Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Multip Val Log Soft Comput 17(2–3):255–287
30. Matheus CJ, Rendell LA (1989) Constructive induction on decision trees. In: Proceedings of the 11th international joint conference on artificial intelligence (IJCAI), Detroit, MI
31. Van Vechten JA (1969) Quantum dielectric theory of electronegativity in covalent systems. i. electronic dielectric constant. Phys Rev 182(3):891
32. Horel T, Singer Y (2016) Maximization of approximately submodular functions. In: Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R (eds) Advances in neural information processing systems 29, Curran Associates, Inc, pp 3045–3053
33. Du D-Z, Graham RL, Pardalos PM, Wan P-J, Wu W, Zhao W (2008) 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
34. Zhou Y, Spanos CJ (2016) Causal meets submodular: subset selection with directed information. In: Proceedings of the neural information processing systems conference, pp 2649–2657
35. Rao M, Chen Y, Vemuri BC, Wang F (2004) Cumulative residual entropy: a new measure of information. IEEE Trans Inf Technol 50(6):1220–1228

Panagiotis Mandros is a PhD candidate at the Max Planck Institute for Informatics and the University of Saarland. He works on the intersection of Information Theory, Statistics, and Optimization, developing theory and algorithms for discovering interesting and robust associations in high-dimensional data. He is the leading author in several publications at top tier Data Mining and Knowledge Discovery venues, receiving for his contributions the 2018 IEEE ICDM Best Paper Award. In his free time you may find him deadlifting like a beast, climbing like a mountain goat, or partying like it's 1999.

Mario Boley is a lecturer at the Department of Data Science and AI at Monash University. His main research interest are the algorithms, applications, and statistical foundation of human-centered data analysis methods. These are methods that seek to empower human users by producing simple and interpretable answers to critical analysis questions. As a key application, Mario collaborates with materials science research groups around the world to contribute to the discovery of the next generation of functional materials. Other than his research, Mario enjoys political history, ginger tea, and functional programming.

Jilles Vreeken is tenured faculty at the CISPA Helmholtz Center for Information Security, where he leads the Exploratory Data Analysis group. In addition, he is Honorary Professor at Saarland University, and Senior Researcher at the Max Planck Institute for Informatics. His research interests include virtually all topics in data mining and machine learning. He authored over 90 conference and journal papers, 3 book chapters, won three best paper awards, the 2010 ACM SIGKDD Doctoral Dissertation Runner-Up Award, and the 2018 IEEE ICDM Tao Li Award. He likes to travel, to think, and to think while traveling.