Discovering Reliable Approximate Functional Dependencies

Panagiotis Mandros
Max Planck Institute for Informatics
and Saarland University, Germany
pmandros@mpi-inf.mpg.de

Mario Boley
Max Planck Institute for Informatics
and Saarland University, Germany
mboley@mpi-inf.mpg.de

Jilles Vreeken
Max Planck Institute for Informatics
and Saarland University, Germany
jilles@mpi-inf.mpg.de

ABSTRACT

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

As we want to be agnostic on the form of the dependence, we adopt an information-theoretic approach, and construct a reliable, bias correcting score that can be efficiently computed. Moreover, we give an effective optimistic estimator of this score, by which for the first time we can mine the approximate functional dependencies from data with guarantees of optimality. Empirical evaluation shows that the derived score achieves a good bias for variance trade-off, can be used within an efficient discovery algorithm, and indeed discovers meaningful dependencies. Most important, it remains reliable in the face of data sparsity.

CCS CONCEPTS

•Information systems →Data mining; •Mathematics of computing →*Probability and statistics;*

KEYWORDS

Pattern discovery, Information theory

ACM Reference format:
Panagiotis Mandros, Mario Boley, and Jilles Vreeken. 2017. Discovering Reliable Approximate Functional Dependencies. In *Proceedings of KDD '17, August 13-17, 2017, Halifax, NS, Canada*, , 9 pages.
DOI: 10.1145/3097983.3098062

1 INTRODUCTION

Discovering dependencies is an important and well-studied topic in data mining. Most proposals, however, focus specifically on *symmetric* dependencies. That is, they aim to find variable sets that strongly correlate or associate with a target variable. In many applications, however, *asymmetric*, or targeted dependencies are of particular interest. When anonymizing a dataset, for example, we need to be certain a private attribute cannot be reconstructed given the public attribute, while we do not care for the opposite direction. Similarly, in scientific applications we want to hypothesize whether a certain target variable, say an effect, can be explained by the observed variables, the potential causes, and not the other way around. Generally, an effective procedure to detect functional dependencies from data allows us to rule out alternate theories about our domain and to determine whether finding concrete models, e.g., by statistical learning, is worthwhile, or if we rather should acquire more data or enrich our feature space first [6].

More formally, given a target variable Y and a set of attributes $\mathcal{X}$, we want to *measure* the degree at which Y has a functional, or an *approximate* functional dependence on $\mathcal{X}$, i.e., if $Y \approx f(\mathcal{X})$. Additionally, we want to efficiently discover whether any such $\mathcal{X}$ exists in our data. The database community studied how to infer *exact* functional dependencies, as these allow for normalization, i.e., reducing redundancy. These methods are not suited to our end, however, as they do not measure the approximation in terms of an intuitive score, and in addition, make implicit closed-world assumptions based on the schema of the data [7, 10, 14].

On the contrary, information theory provides an intuitive and interpretable measure to address these issues. The fraction of information quantifies functional dependence in terms of proportional reduction of uncertainty about Y when observing $\mathcal{X}$ [1, 3, 19]. Information-theoretic measures, however, are sensitive to data sparsity and as a result, the fraction of information overestimates the amount of dependence [21]. For large dimensionalities of $\mathcal{X}$, it is even possible that a functional dependence is indicated when $\mathcal{X}$ and Y are actually independent. This makes it a non-*reliable* score. In addition, maximizing it is **NP**-hard [12].

In this paper we propose a *reliable* measure for approximate functional dependencies based on the fraction of information. Even in extreme cases of data sparsity, it does not show dependence. In addition, we derive an effective optimistic estimator for this score, that allows for an admissible branch-and-bound algorithm to discover the top-k optimal, or α-approximate optimal strongest dependencies. Empirical evaluation shows that the derived score achieves a good bias for variance trade-off, and in addition, it does not favor spurious dependencies. The corresponding optimistic estimator is a data-dependent quantity, and by exploiting the structure of the data, leads to an effective search algorithm. Lastly, concrete findings in two exemplary application domains, AI and Materials Science, reproduce sensible domain information.

The main contributions of this paper are the following. We

(i) propose a consistent estimator for the fraction for information score that is not prone to spurious dependencies,
(ii) provide an efficient branch-and-bound algorithm for the discovery of optimal, and α-approximate optimal top-k dependencies, and

KDD '17, August 13-17, 2017, Halifax, NS, Canada
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 978-1-4503-4887-4/17/08…$15.00
DOI: 10.1145/3097983.3098062

(iii) provide empirical evaluation on a wide range of real and synthetic datasets.

The paper is structured as follows. We formally introduce the two problems we consider in Section 2. Next, in Section 3 we propose our fraction of information score, and in Section 4 we detail how to optimize it by deriving a bounding function for a branch-and-bound search scheme. Following, in Section 5 we evaluate the performance on a variety of tasks. Finally, we round up with conclusions in Section 6.

2 PROBLEM DEFINITION

We consider a discrete sample space governed by some probability mass function p for which we have defined $d + 1$ discrete random variables $\mathcal{R} = \{X_1, \ldots, X_d, Y\}$ with domains $V(X_1), \ldots, V(X_d)$, and $V(Y)$, respectively. Subsets $\mathcal{S} \subseteq \mathcal{R}$ are identified with vector-valued random variables in the usual way with domain $\mathbf{V}(\mathcal{S}) = \times_{R \in \mathcal{S}} V(R)$. We consider the variable Y as the **output variable** and the remaining variables $\mathcal{I} = \{X_1, \ldots, X_d\}$ as the **input variables**, and our goal is to discover subsets of the input variables $\mathcal{X} \subseteq \mathcal{I}$ that approximately *determine* Y. In particular, we are interested in approximations to the concept of **functional dependencies**, i.e., the case when there is a function $f : \mathbf{V}(\mathcal{X}) \to V(Y)$ such that for all $\mathbf{x} \in \mathbf{V}(\mathcal{X})$ it holds that

$$p(Y = y \mid \mathcal{X} = \mathbf{x}) = \begin{cases} 1 & \text{, if } y = f(\mathbf{x}) \\ 0 & \text{, otherwise} \end{cases} . \tag{1}$$

Relaxing this rather strict concept is necessary because it is rare that such a completely deterministic relationship exists—if the random variables correspond to measurements of real-world quantities there are usually unobserved subtle effects or noise that cause Eq. (1) to not hold exactly.

One traditional approach to relax Eq. (1) is to use instead the condition $p(Y = y \mid \mathcal{X} = \mathbf{x}) \geq 1 - \epsilon$ if $y = f(\mathbf{x})$, for some fixed value $\epsilon \in (0, 1]$, i.e., to allow a certain fraction of events to not obey the functional relation. However, as with any parameterization based on a hard threshold, this parameter is difficult to set in practice and additionally only provides a qualitative and not a quantitative relaxation [7]. That is, it does not allow us to express "how far" is Y from being determined by $\mathcal{X}$. In order to address these issues one can quantify the degree of functional dependence through information theoretic measures. A particularly useful way of doing this is to use the concept of **fraction of information** (F) [1, 3, 19], which is defined as

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

where $H(Y) = -\sum_{y \in V(Y)} p(y) \log(p(y))$ denotes the **Shannon entropy** and $H(Y \mid \mathcal{X}) = \sum_{\mathbf{x} \in \mathbf{V}(\mathcal{X})} p(\mathbf{x}) H(Y \mid \mathcal{X} = \mathbf{x})$ the conditional Shannon entropy [23]. The numerator is referred to as **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 of uncertainty about Y by knowing $\mathcal{X}$. Moreover, the extreme values, $F(\mathcal{X}; Y) = 1$ and $F(\mathcal{X}; Y) = 0$, correspond to a functional dependence and statistical independence, respectively. With this notion, we can go about discovering approximate functional dependencies from data.

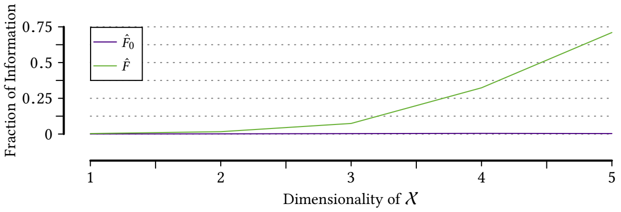


Figure 1: The fraction of information score against increasing dimensionality for $\mathcal{X}$; using $n = 1000$ samples from $\mathcal{R} = \{X_1, \ldots, X_5, Y\}$ where all variables are mutually independent and $|V(A)| = 4$ for all $A \in \mathcal{R}$. Since $\mathcal{X}$ and Y are independent, the reliable fraction of information should be constantly 0. However, the baseline estimator $\hat{F}$ shows increasing functional dependence. On the contrary, our proposed corrected score, $\hat{F}_0$, is always 0.

For that we assume that a dataset $\mathbf{D}_n \in \mathbf{V}(\mathcal{R})^n$ is given consisting of n i.i.d. samples $\mathbf{d}_1, \ldots, \mathbf{d}_n$ generated according to the joint distribution p. Such a dataset induces empirical probability estimates for all our random variables $\mathcal{S} \subseteq \mathcal{R}$ given by $\hat{p}(\mathcal{S} = \mathbf{v}) = c(\mathcal{S} = \mathbf{v})/n$ with the **empirical counts**

$$c(\mathcal{S} = \mathbf{v}) = |\{\mathbf{d} \in \mathbf{D}_n : \mathbf{d}(S) = \mathbf{v}(S) \text{ for all } S \in \mathcal{S}\}|$$

(where $\mathbf{d}(S)$ is the entry in $\mathbf{d}$ corresponding to variable S). In turn, these empirical probabilities give rise to empirical estimators $\hat{H}, \hat{I}, \hat{F}$ for our quantities of interest, H, I, and F. However, trying to directly discover approximate functional dependence using the empirical fraction of information $\hat{F}$ is bound to fail, because this estimator is not unbiased, i.e., we have $\mathbb{E}[\hat{F}(\mathcal{X}; Y)] \neq F(\mathcal{X}; Y)$ for finite n, as it is the case with many dependence measures [4, 9, 11, 17, 20, 26–28]. This holds in particular also for the case when $F(\mathcal{X}; Y) = 0$, i.e., when $\mathcal{X}$ contains no information about Y. This situation, which is referred to as the (lack of) zero-baseline property [21], can be misleading in practice. Even worse, independent of the true value $F(\mathcal{X}; Y)$, the bias depends on the number of attainable distinct values for $\mathcal{X}$, and favors larger attribute sets over smaller ones (which follows from the bias of the empirical mutual information, see, e.g., [22]). See Fig. 1 for a quantitative demonstration of both of these facts.

Even if a more suitable estimator was available, the challenge of which variable sets $\mathcal{X} \subseteq \mathcal{I}$ to test for high functional dependence scores remains—naively considering all 2^d options is practically infeasible. Thus, to derive a useful method for the reliable discovery of functional dependencies from data, we have to solve the following two problems:

(i) Find a more reliable empirical estimator $\hat{F}'$ for F; in particular one that satisfies the zero-baseline property and obtains better dimensionality bias.
(ii) Identify structural properties of $\hat{F}'$ that allow to derive an effective search algorithm for discovering the variable sets with the highest functional dependence scores.

We will present solutions to each of these problems in turn, in Sections 3 and 4, respectively.

3 RELIABLE FRACTION OF INFORMATION

Intuitively, the reason why $\hat{F}$ is unreliable as an estimator for F is that it does not take into account the confidence in the empirical estimates $\hat{H}(Y|\mathcal{X} = \mathbf{x})$. This is especially obvious in the extreme case when the empirical count $c(\mathcal{X} = \mathbf{x})$ is equal to 1. In this situation $c(Y = y, \mathcal{X} = x) = 1$ 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. Moreover, if the data size n is small compared to the observed domain of $\mathcal{X}$, this case is likely to occur for many of the sampled values for $\mathcal{X}$—even when $F(\mathcal{X}; Y) = 0$, which coincides with the highest error, because then $H(Y|\mathcal{X} = \mathbf{x}) = H(Y)$.

This last observation suggests a path to a more reasonable estimator: while it is hard to determine the bias in the general case, it is likely much easier under the assumption of independence $F(\mathcal{X}; Y) = 0$, which, as pointed out above, corresponds exactly to the case of highest estimation error when the empirical observations are sparse. More concretely, let us denote by $b_0(\mathcal{X}, Y, n)$ the **bias under independence** defined as

$$b_0(\mathcal{X}, Y, n) = \mathbb{E}[\hat{F}(\mathcal{X}; Y) \mid F(\mathcal{X}; Y) = 0]$$

where the expectation is taken w.r.t. the random dataset $\mathbf{D}_n \sim p$ of size n. Let us assume that we have a good estimator $\hat{b}_0$ for this quantity. With this we can define a corrected estimator, let us refer to it as **reliable fraction of information** $\hat{F}_0$, as follows:

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

This approach essentially trades the bias of $\hat{F}$ with that of $\hat{b}_0$ when $F(\mathcal{X}; Y) = 0$. We have:

$$\begin{aligned}&\mathbb{E}[\hat{F}_0(\mathcal{X}; Y) - F(\mathcal{X}; Y) \mid F(\mathcal{X}; Y) = 0]\\ =&\mathbb{E}[\hat{F}(\mathcal{X}; Y) - \hat{b}_0(\mathcal{X}, Y, n) - 0 \mid F(\mathcal{X}; Y) = 0]\\ =&b_0(\mathcal{X}, Y, n) - \mathbb{E}[\hat{b}_0(\mathcal{X}, Y, n) \mid F(\mathcal{X}; Y) = 0] \ .\end{aligned}$$

A non-parametric choice for $\hat{b}_0$, which we use in this paper, corresponds to the *permutation model* (Lancaster [13, Chap. 11.2]), i.e., considering all possible datasets $\mathbf{D}'_n$ resulting from independently permuting the Y-values associated to the $\mathcal{X}$-values in the given empirical data $\mathbf{D}_n$. Formally, let S_n denote the symmetric group of degree n, i.e., S_n consists of all $n!$ bijections $\sigma\colon \{1, \ldots, n\} \to \{1, \ldots, n\}$. For a bijection $\sigma \in S_n$, let Y_σ denote the permuted version of Y, i.e., the variable with data entries $\mathbf{d}_i(Y_\sigma) = \mathbf{d}_{\sigma(i)}(Y)$. With this we can define the **permutation model** as the probabilities $\hat{\mathbb{P}}_0$ (and corresponding expectations $\hat{\mathbb{E}}_0$) resulting from permuting the empirical data of Y by a uniform random permutation from S_n. Using this model, the expectation of the empirical mutual information under independence, $\hat{m}_o$, is estimated as

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

Clearly, a naive evaluation of this expression is computationally infeasible (order of $n!$). However, one can dramatically reduce the complexity by reformulating the above expression as a function of contingency table cell values and exploiting its symmetries [25]. More precisely, let the observed domains of $\mathcal{X}$ and Y be $\hat{\mathbf{V}}(\mathcal{X}) = \{\mathbf{x}_1, \ldots, \mathbf{x}_R\}$ and $\hat{V}(Y) = \{y_1, \ldots, y_C\}$, respectively. Moreover, 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 complete joint count configuration $c = (c_{i,j}\colon 1 \leq i \leq C, 1 \leq j \leq R)$ we refer to as **contingency table**. Noting that $\hat{I}(\mathcal{X}, Y_\sigma)$ is a function of the contingency table c resulting from the random permutation, the estimator $\hat{m}_0$ can be rewritten as

$$\hat{m}_0(\mathcal{X}, Y, n) = \sum_{c \in \mathcal{T}} \hat{\mathbb{P}}_0[c] \hat{I}(c) = \sum_{c \in \mathcal{T}} \hat{\mathbb{P}}_0[c] \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{c_{ij}}{n} \log \frac{c_{ij} n}{a_i b_j} \quad (2)$$

where $\mathcal{T} = \mathcal{T}(\mathcal{X}, Y)$ is the set of all possible contingency tables indexed by the values $\hat{\mathbf{V}}(\mathcal{X})$ and $\hat{V}(Y)$ (note that $\hat{\mathbb{P}}_0[c] > 0$ only for c resulting in the observed marginal counts a, b).

As this expression is still infeasible, Vinh et al. [25] propose to re-order the terms of the sum according to the possible count values that can be found in individual table cells. The permutation model implies that the empirical counts c_{ij} for the joint events $\mathcal{X} = \mathbf{x}_i, Y = y_j$ are generated according to the probabilities

$$\hat{\mathbb{P}}_0[c_{ij} = k] = h(k; a_i, b_j, n)$$

where h is the probability mass function of the hypergeometric distribution with c_{ij} the number of successes, a_i the number of draws, b_j the number of total successes, and n the population size. This allows us to group terms according to their count values for a specific table cell, which can be systematically enumerated from the support of the hypergeometric distribution, i.e., $k \in [\max(0, a_i + b_j - n), \min(a_i, b_j)]$. We can then compute $\hat{m}_0$ as

$$\hat{m}_0(\mathcal{X}, Y, n) = \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{k=\max(0, a_i + b_j - n)}^{\min(a_i, b_j)} h(k; a_i, b_j, n) \frac{k}{n} \log \frac{kn}{a_i b_j} \ .$$

Using the recurrence relation of the hypergeometric distribution, the computational complexity can be further reduced to the order of $\max(nR, nC)$ [20]. Moreover, it is easily parallelizable. Hence, we end up with an efficiently computable estimator for the bias under independence $\hat{b}_0(\mathcal{X}, Y, n) = \hat{m}_0(\mathcal{X}, Y, n)/\hat{H}(Y)$.

In addition to being computationally efficient, the resulting reliable functional dependence score $\hat{F}_0(\mathcal{X}; Y) = \hat{F}(\mathcal{X}; Y) - \hat{b}_0(\mathcal{X}, Y, n)$ satisfies several other properties. First of all, it is indeed a consistent estimator of F. One can show [26] that $\lim_{n \to \infty} \hat{m}_0(\mathcal{X}, Y, n) = 0$, which implies together with the consistency of $\hat{F}$ that

$$\lim_{n \to \infty} \hat{F}_0(\mathcal{X}; Y) = F(\mathcal{X}; Y) \ .$$

Moreover, $\hat{F}_0$ remains upper-bounded by 1, although this value is only attainable in the limit case $n \to \infty$ (for true functional dependencies). Most importantly, in contrast to the naive estimator, $\hat{F}_0$ approaches zero[1] as the data size relative to the empirical domain $\hat{\mathbf{V}}(\mathcal{X})$ approaches one. In other words, $\hat{b}_0(\mathcal{X}, Y, n)$ penalizes spurious dependencies that can easily appear for high dimensional $\mathcal{X}$—justifying the name *reliable* fraction of information.

4 SEARCH SCHEME

After deriving a suitably corrected empirical estimator for the fraction of information, we can now turn to the problem of using it for the discovery of approximate functional dependencies from a given dataset. Essentially, this is a combinatorial optimization problem

[1] It fact, it is principally not lower bounded by 0 since the empirical fraction of information can be less than the correction term. However, these are rare cases, which strongly indicate independence.

Algorithm 1: Best-first branch-and-bound; Given input and output variables $\mathcal{I}$ and Y, finds α-approximation to top-k result set $\mathcal{F}_k \subseteq \mathcal{P}(\mathcal{I})$ w.r.t. $f(\mathcal{X}) = \hat{F}_0(\mathcal{X}; Y)$ using optimistic estimator $\overline{f}(\mathcal{X}) = 1 - \hat{b}_0(\mathcal{X}; Y, n)$.

```
Bst-BB(Q,$\mathcal{F}_k$,$k$): begin
    if $\mathbf{Q} = \emptyset$ or $\overline{f}(\mathtt{top}(\mathbf{Q}))/f(\mathcal{F}_k[k]) \leq \alpha$ then
        return $\mathcal{F}_k$
    else
        $\mathbf{R} = \mathbf{r}(\mathtt{top}(\mathbf{Q}))$
        $\mathcal{F}_k' = \mathtt{top\text{-}k}(\mathcal{F}_k \cup \mathbf{R})$
        $\mathbf{Q}' = (\mathbf{Q} \setminus \{\mathtt{top}(\mathbf{Q})\}) \cup \{\mathcal{X} \in \mathbf{R}\colon \overline{f}(\mathcal{X})/f(\mathcal{F}_k'[k]) \geq \alpha\}$
        return Bst-BB($\mathbf{Q}'$,$\mathcal{F}_k$,$k$)
$\mathcal{F}_k$ = Bst-BB($\{\bot\}, \bot, k$)
```

where, given a dataset $\mathbf{D}_n$, the problem is to find a subset $\mathcal{X} \subseteq \mathcal{I}$ with maximal value of $\hat{F}_0$. However, as usual in pattern discovery, we are not just interested in one but several score maximizers—to produce more diverse insights into the data domain and to provide alternatives in subsequent applications. Hence, we end up with a top-k pattern search formulation:

Given *a dataset* $\mathbf{D}_n$ *consisting of* n *i.i.d. samples of random variables* $\mathcal{I}$ *and* Y *and a number* k, **find** *a family* $\mathcal{F}_k$ *of variable sets* $\mathcal{X}_1, \ldots, \mathcal{X}_k \subseteq \mathcal{I}$ *such that no variable set outside* $\mathcal{F}_k$ *has a higher* $\hat{F}_0$*-score than any of the sets in* $\mathcal{F}_k$*, i.e., for all* $\mathcal{X} \in \mathcal{F}_k$ *and* $\mathcal{Z} \in \mathcal{P}(\mathcal{I}) \backslash \mathcal{F}_k$ *it holds that* $\hat{F}_0(\mathcal{Z}; Y) \leq \hat{F}_0(\mathcal{X}; Y)$.

In the search for an algorithm solving the above problem, it is first important to note that maximizing mutual information is **NP**-hard [12]—even approximately and even in the restricted case when all $\mathcal{X}, \mathcal{Z} \subseteq \mathcal{I}$ are conditionally independent given Y. While it is an open question whether this result implies hardness of $\hat{F}_0$-maximization (the correction term changes maximization order), this is a substantial indication that no polynomial time algorithm for our problem exists (even with constant k). On the other hand, the branch-and-bound framework (see, e.g., Mehlhorn and Sanders [16, Chap. 12.4]), while not efficient in terms of the worst-case complexity, can often yield algorithms for hard optimization problems that are very effective in practice—particularly in the best-first search variant.

In a nutshell, best-first branch-and-bound maximizes an objective function $f: \Omega \to \mathbb{R}$ defined on some abstract search space Ω with the help of a **branch operator** $\mathbf{r}: \mathcal{P}(\Omega) \to \mathcal{P}(\Omega)$ and a matching auxiliary selection and **bounding function** $\overline{f}: \Omega \to \mathbb{R}$. The role of the branch operator is to non-redundantly generate the search space from some designated root element $\bot \in \Omega$, i.e., for all $\omega \in \Omega$ there must be a unique sequence $\bot = \omega_1, \ldots, \omega_l = \omega$ such that $\omega_{i+1} \in \mathbf{r}(\omega_i)$ for $i = 1, \ldots, l-1$. The bounding function must guarantee the property

$$\overline{f}(\omega) \geq \max\{f(\omega')\colon \omega' \in \mathbf{r}^*(\omega)\}$$

where $\mathbf{r}^*(\omega)$ denotes the set of all $\omega' \in \Omega$ that can be generated from ω by multiple applications of $\mathbf{r}$. Based on these ingredients, a branch-and-bound algorithm simply enumerates Ω starting from $\bot$, but uses $\overline{f}$ to avoid expanding elements that cannot yield an improvement over the best solution found so far.

For our problem the search space is $\mathcal{P}(\mathcal{I})$, for which a suitable branch operator is simply given by

$$\mathbf{r}(\mathcal{X}) = \{\mathcal{X} \cup \{X_i\}\colon i > \max\{j\colon X_j \in \mathcal{X}\}\}$$

i.e., we ensure non-redundant generation by creating a lexicographical order on the power set of the input variables and only enumerate lexicographically larger elements from a given set $\mathcal{X}$. In order to derive a bounding function $\overline{f}$ for our objective function $f(\mathcal{X}) = \hat{F}_0(\mathcal{X}; Y)$, we first need to establish another central property of the correction term $\hat{b}_0(\mathcal{X}, Y, n)$.

Theorem 4.1. *Given two sets of variables* $\mathcal{X} \subset \mathcal{X}' \subseteq \mathcal{I}$ *then* $\hat{b}_0(\mathcal{X}, Y, n) \leq \hat{b}_0(\mathcal{X}', Y, n)$*, i.e., the correction term is monotonically increasing with respect to the subset relation.*

Proof. It is sufficient to consider the case $\mathcal{X}' = \mathcal{X} \cup \{X\}$ for some $X \notin \mathcal{X}$, i.e. the cardinality differs by one variable. The general case follows inductively. Following the notation of Eq. (2), let the marginals of $Y, \mathcal{X}$, and $\mathcal{X}'$ be a_i for $i = 1, \ldots, R$, b_j for $j = 1, \ldots, C$, and b_j' for $j = 1, \ldots, C'$, respectively. Note that $C' \geq C$. We need to show that $\hat{m}_0(\mathcal{X}, Y, n) \leq \hat{m}_0(\mathcal{X}', Y, n)$, i.e., per definition $\sum_{c \in \mathcal{T}} \hat{\mathbb{P}}_0[c]\hat{I}(c) \leq \sum_{c' \in \mathcal{T}'} \hat{\mathbb{P}}_0[c']\hat{I}(c')$.

In order to do this, we first define a relation between the contingency tables of $\mathcal{T} = \mathcal{T}(\mathcal{X}, Y)$ and $\mathcal{T}' = \mathcal{T}(\mathcal{X}', Y)$. Let $\pi\colon \hat{\mathbf{V}}(\mathcal{X}') \to \hat{\mathbf{V}}(\mathcal{X})$ be the projection of values from $\mathcal{X}'$ to values of $\mathcal{X}$ defined by $\pi(\mathbf{x}') = \mathbf{x}$. We can extend this projection to the sets of contingency tables $\pi\colon \mathcal{T}' \to \mathcal{T}$ by finding the counts in the column corresponding to $\mathbf{x} \in \mathbf{V}(X)$ of $\pi(c')$ as the sum of the columns in c' corresponding to $\pi^{-1}(\mathbf{x})$. We will prove the claim by showing that for all $c \in \mathcal{T}$ we have $\hat{\mathbb{P}}_0[c]\hat{I}(c) \leq \sum_{c' \in \pi^{-1}(c)} \hat{\mathbb{P}}_0[c']\hat{I}(c')$.

First, it follows from the chain rule of information and from mutual information being non-negative [2] that $\hat{I}(c) \leq \hat{I}(c')$ for $c = \pi(c')$. Next we show that $\hat{\mathbb{P}}_0[c] = \sum_{c' \in \pi^{-1}(c)} \hat{\mathbb{P}}_0[c']$, which concludes the proof. For any contingency table $z \in \mathcal{T}(\mathcal{Z}, Y)$ let $S_n[z] = \{\sigma \in S_n\colon c(\mathcal{Z}, Y_\sigma) = c\}$ denote the set of permutations that result in z. Let $\sigma \in S_n \setminus S_n[c]$. This means that $c_{i,j}(\mathcal{X}, Y) \neq c_{i,j}(\mathcal{X}, Y_\sigma)$ for at least one cell i, j. Denoting by

$$\pi^{-1}(j) = \{j'\colon 1 \leq j' \leq C', \pi(\mathbf{x}_{j'}') = \mathbf{x}_j\}$$

the set of all indices of values of $\mathcal{X}'$ that are projected down to $\mathbf{x}$, it follows by the definition of π that

$$\sum_{j' \in \pi^{-1}(j)} c_{i,j'}'(\mathcal{X}', Y) \neq \sum_{j' \in \pi^{-1}(j)} c_{i,j'}'(\mathcal{X}', Y_\sigma) \ .$$

So for at least one $j' \in \pi^{-1}(j)$ it is $c_{i,j'}'(\mathcal{X}', Y) \neq c_{i,j'}'(\mathcal{X}', Y_\sigma)$, and, thus we also have that $\sigma \notin S_n[c']$ and can conclude

$$S_n[c] \supseteq \bigcup_{c' \in \pi^{-1}(c)} S_n[c'] \ . \tag{3}$$

Now let $z' \in \mathcal{T}(\mathcal{X}', Y)$ with $\pi(z') \neq c$ and assume for a contradiction that $S_n[c] \supset S_n[c']$, i.e., there is an $\sigma \in S_n[c] \cap S_n[z']$. Let us denote $z = \pi(z')$. Since $S_n[c] \cap S_n[z] = \emptyset$, we know that $\sigma \notin S_n[z]$. However, it follows from Eq. (3) that $\sigma \notin S_n[z']$—a contradiction, and, hence $S_n[c] = \bigcup_{c' \in \pi^{-1}(c)} S_n[c']$. Thus, as desired

$$\hat{\mathbb{P}}_0[c] = \frac{|S_n(c)|}{|S_n|} = \sum_{c' \in \pi^{-1}(c)} \frac{|S_n(c')|}{|S_n|} = \sum_{c' \in \pi^{-1}(c)} \hat{\mathbb{P}}_0[c'] \ .$$

□

With this theorem (and the fact that the conditional entropy is bounded from below by 0), it follows for all $\mathcal{X} \subseteq \mathcal{X}' \subseteq \mathcal{I}$ that

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

Hence, since $\mathcal{X} \subseteq \mathcal{X}'$ for $\mathcal{X}' \in \mathbf{r}(\mathcal{X})$, we can use $\overline{f}(\mathcal{X}) = 1 - \hat{b}_0(\mathcal{X}, Y, n)$ as valid bounding function for the branch-and-bound search. Besides the features mentioned above, the search scheme also provides the option of relaxing the required result guarantee to that of an α-approximation for accuracy parameter $\alpha \in (0, 1]$. This means that the resulting family of variable sets $\mathcal{F}_k$ will satisfy the relaxed condition that for all $\mathcal{X} \in \mathcal{F}_k$ and $\mathcal{Z} \in \mathcal{P}(\mathcal{I}) \setminus \mathcal{F}_k$ it holds that $\alpha \hat{F}_0(\mathcal{Z}; Y) \leq \hat{F}_0(\mathcal{X}; Y)$. Hence, using α-values of less than 1 allows to trade accuracy for computation time.

The pseudocode in Algorithm 1 summarizes the resulting method for the discovery of approximate functional dependencies. The algorithm maintains a priority queue $\mathbf{Q}$ that holds the search frontier and a current result set $\mathcal{F}_k$ throughout the search. In the beginning of each iteration, it checks whether the search has terminated, which is the case either when the frontier is empty or the potential of top-potential element from the frontier (w.r.t. the bounding function) is less than the k-th best $\hat{F}_0$-value of the current result. As long as this condition is not satisfied, the search continuous by expanding the top-potential variable set (line 5), and using its successors to update the current result set (line 6), as well as the priority queue (line 7).

5 EMPIRICAL EVALUATION

In this section, we study the empirical performance of discovering approximate functional dependencies based on $\hat{F}_0$. This includes, the bias of $\hat{F}_0$ as an estimator of the true F functional, the performance of the bounding function $\overline{f}$ in branch-and-bound search, and two concrete examples of functional dependencies in real datasets.

5.1 Estimation Bias

In this section we evaluate the bias and variance of our corrected estimator $\hat{F}_0$. It is instructive to see the behavior of the bias for various amounts of dependence, and not in the particular case of independence, i.e., $F = 0$, where $\hat{F}_0$ aims to be unbiased (see Fig. 1 for an empirical confirmation of this fact). For that let us denote by P the set of all joint probability mass functions over two random variables X and Y with $|V(X)| = |V(Y)| = 3$, and by $P[a, b]$ all such probability mass functions for which we have a functional dependence score of $F_p(X; Y) \in (a, b]$. We are interested in the behavior of the estimation bias over P under a distribution that puts equal weight on the four different regimes "weak" ($P[0, 0.25]$), "low" ($P[0.25, 0.5]$), "high" ($P[0.5, 0.75]$), and "strong" ($P[0.75, 1]$).

More specifically, let $\tau(\mathbf{D}_n)$ be the result of the F-estimator τ computed on data $\mathbf{D}_n$ and $b(n)(p, \tau)$ the bias of τ when fixing the underlying pmf to p, i.e.,

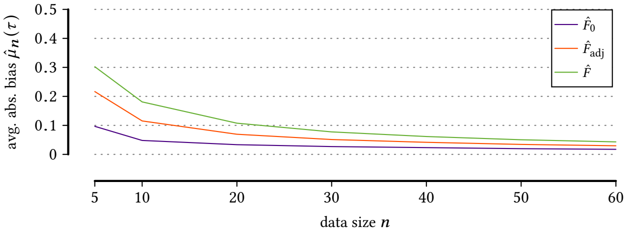


Figure 2: Estimated absolute bias $\hat{\mu}_n(\tau)$ of estimators $\tau = \hat{F}$, $\hat{F}_{\text{adj}}$, and the reliable fraction of information $\hat{F}_0$, averaged over all 100 pmfs $p \in P$ across different data sizes n.

$$b(n)(p, \tau) = \mathbb{E}_{\mathbf{D}_n \sim p}[\tau(\mathbf{D}_n)] - F_p(X; Y) \ . \tag{4}$$

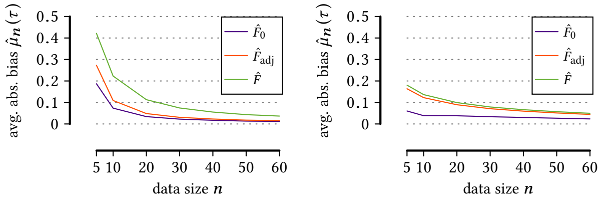
To estimate the expected value $\mu_n(\tau)$ and standard variation $\sigma_n(\tau)$ of the absolute bias $|b(n)(p, \tau)|$ for the pmfs from P, we uniformly sample 100 pmfs $p^{(1)}, \ldots, p^{(100)}$ in equal proportions from the four different regimes, i.e., 25 each from $P[0, 0.25]$, $P[0.25, 0.5]$, $P[0.5, 0.75]$, and $P[0.75, 1]$. For each pmf $p^{(i)}$ we can calculate the true F value directly from its definition. To compute $b(n)(p^{(i)}, \tau)$ it then only remains to empirically estimate the expectation term in Eq. (4), for which we sample per pmf $p^{(i)}$ a total of 1000 datasets $D_n \sim p^{(i)}$ of size n. By averaging over all $p^{(i)}$, we end up with the desired estimates $\hat{\mu}_n(\tau)$ and $\hat{\sigma}_n(\tau)$ for the absolute bias of estimator τ with sample size n.

Equipped with this procedure we can go ahead and compare the performance of $\hat{F}_0$ to other estimators. In addition to the naive uncorrected estimator $\hat{F}$ we also introduce an alternative correction resulting from the application of the quantification adjustment framework proposed by Romano et al. [21]. This correction, which we denote by $\hat{F}_{\text{adj}}$, is defined as

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

Thus, we consider $\tau \in \{\hat{F}, \hat{F}_{\text{adj}}, \hat{F}_0\}$. For each estimator, we compute $\hat{\mu}_n(\tau)$ and $\hat{\sigma}_n(\tau)$ for data sizes $n \in \{5, 10, 20, 30, 40, 50, 60\}$. We focus on small data sizes, because any consistent estimator converges to F for $n \to \infty$. Furthermore, we can expect the small data sizes for the small domain size $|V(X)| = 3$ of this experiment to behave similar to larger data sizes combined with the potentially huge domains occurring during the algorithmic search (resulting from the complex random variables $\mathcal{X}$).

In Figure 2 we present the results for the estimated absolute bias $\hat{\mu}_n(\tau)$ of estimators $\tau = \hat{F}$, $\hat{F}_{\text{adj}}$, and the reliable fraction of information $\hat{F}_0$, averaged over all 100 pmfs $p \in P$ across different data sizes n. We observe that our estimator $\hat{F}_0$ achieves a lower bias for all n compared to $\hat{F}$ and $\hat{F}_{\text{adj}}$, and converges fast to a bias close to zero after 10 samples. The differences in bias are apparent in the cases of $n = 5$ and $n = 10$ samples. These are cases where the insufficient data samples cause $\hat{H}(Y \mid \mathcal{X} = \mathbf{x})$ to approach 0, independent of the true distribution $\hat{p}$. In such scenarios, the estimators $\hat{F}$ and $\hat{F}_{\text{adj}}$ start to show functional dependence, while



(a) Weak dependence (b) Strong dependence

Figure 3: Estimated absolute bias $\hat{\mu}_n(\tau)$ of estimators $\tau = \hat{F}$, $\hat{F}_{\text{adj}}$, and the reliable fraction of information $\hat{F}_0$, averaged over $p \in P[0, 0.25]$ (left) and over $p \in P[0.75, 1]$ (right), across different data sizes n.

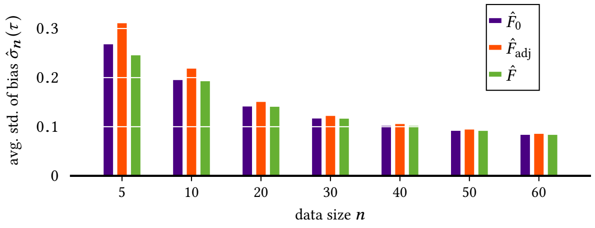


Figure 4: Estimated standard deviation $\hat{\sigma}_n(\tau)$ of estimators $\tau = \hat{F}$, $\hat{F}_{\text{adj}}$, and the reliable fraction of information $\hat{F}_0$, averaged over all 100 pmfs $p \in P$ across different data sizes n.

$\hat{F}_0$ is designed to show independence for reliability reasons. So it is useful to see that this design, also offers a better bias.

We can further draw conclusions about the behavior of $\hat{F}_0$ by considering only the "weak" and "strong" dependencies, i.e., where F is closer to independence and functional dependence respectively. As such, we present in Figure 3 the estimated absolute bias $\hat{\mu}_n(\tau)$, averaged over $p \in P[0, 0.25]$ (left), and $p \in P[0.75, 1]$ (right). We see that in both cases, i.e., when there is low and high functional dependence, $\hat{F}_0$ achieves a lower bias, as it was the case with $p \in P[0, 1]$. Since our score aims to be unbiased under the null hypothesis, we observe a high correction over the $\hat{F}$ for weak dependencies. Even for high dependencies, where one could expect to have less correction, we see that $\hat{F}_0$ is practically unbiased across all n.

For the standard deviation $\hat{\mu}_n(\tau)$, we present in Figure 4 the results after averaging over all 100 pmfs $p \in P$ across different data sizes n. We observe that $\hat{F}_0$ has an almost equal $\hat{\mu}_n$ for all n in comparison with $\hat{F}$, and a lower one with $\hat{F}_{\text{adj}}$. Estimators achieve better bias by trading variance, and from Figures 2, 3, and 4, we see that in comparison with $\hat{F}$ and $\hat{F}_{\text{adj}}$, we trade very little variance for a large bias correction. With the previous observations, we can conclude that $\hat{F}_0$ is a suitable estimator for the fraction of information F, as desired.

5.2 Optimization performance

In this section, we investigate the performance of the branch-and-bound algorithm combined with our optimistic estimator $\overline{f}(\mathcal{X}) = 1 - \hat{b}_0(\mathcal{X}, Y, n)$. Specifically, we are interested in the effects of having a data dependent quantity in our bounding function, i.e., $\hat{b}_0(\mathcal{X}, Y, n)$, which in addition, acts as a penalty for non-reliable dependencies.

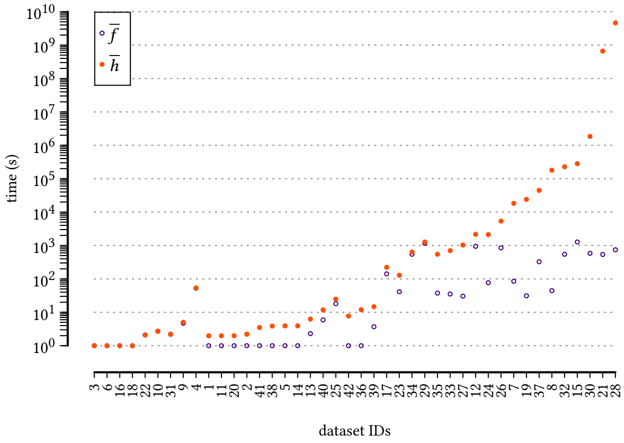


Figure 5: Computation time of branch-and-bound with $\overline{f}$ and $\overline{h}$ as bounding function respectively, across all datasets sorted according to speed-up.

For this experiment we utilize the KEEL data repository.[2] We use all classification datasets with $n \in [100, 13000]$, $d \in [6, 90]$, without missing values, resulting in 42 datasets with an average number of 2800 data samples and 24 attributes.[3] All metric attributes are discretized using the method of Fayyad and Irani [5]. The datasets are summarized in Table 1. All experiments were executed on a dedicated Intel Xeon E5-2643 v3 machine with 256 GB memory. We make our code available online for research purposes.[4]

We employ the algorithm to retrieve the top dependence, and individually set the α for each dataset, such that the algorithm terminates in less than 30 minutes. We report time, α used, pruning percentage of the search space, depth of the solution, and the maximum depth the algorithm had to explore in Table 1.

The percentage of the datasets where $\alpha = 1$, i.e., 30 out of 42, show that an optimal solution can be discovered in under 30 minutes for the majority of the cases considered. In 28 datasets it takes a maximum of 6 minutes. For the rest, reasonable approximations to the optimal solution can be achieved, e.g., $\alpha = 0.9, 0.8$.

We observe that our bounding function $\overline{f}$, is effective in pruning a considerable amount of the search space, i.e., 67.9% on average. In addition, an average of 7.5 maximum depth combined with an average solution depth of 4.0, show that $\overline{f}$ is not simply pruning on set cardinality, but it is a data dependent quantity that selectively explores the search space based on the structure of the data. That is, it can potentially go to higher levels for promising candidates.

To further corroborate on the previous observation, we consider a hypothetical optimistic estimator $\overline{h}$ that prunes based solely on the cardinality of $\mathcal{X}$. For a meaningful comparison, we provide an oracle to this method and restrict the search space to the maximum depth that $\overline{f}$ had to explore, and not the complete space of size 2^d

[2] http://www.keel.es/

[3] Some attributes with obviously wrong type declarations were corrected, e.g., 'sex' was often corrected to be categoric and not metric.

[4] http://eda.mmci.uni-saarland.de/dora/

X_1	X_2	X_3
X_4	X_5	X_6
X_7	X_8	X_9

3	1	3
1	4	1
3	1	3

Figure 6: Tic-tac-toe board with input variables in corresponding board positions and variables contained in top approximate functional dependency marked in red (left); and number of winning combinations each position is involved in (right).

nodes. For example, if the maximum depth is l, then the branch-and-bound with $\overline{h}$ as bounding function will visit $q = \sum_{i=1}^{l} \binom{d}{i}$ nodes. The estimated time for every dataset is then $q \times t$, where t is a node processing time estimated by dividing the completion time of branch-and-bound with $\overline{f}$, with the number of nodes visited. We plot the computation time for both estimators in Figure 5. The datasets are sorted in ascending order of speed-up.

We see that in the majority of the datasets, taking into account the structure of the data is of crucial importance. This is most evident in the last two datasets, where it would take 20 and 146 years respectively to find the solution based on cardinality alone. This plot shows the potential of a data dependent optimistic estimator, as opposed to a simple function evaluating statistics as cardinality, in a potentially hard optimization problem.

Regarding reliability, useful conclusions can be drawn from the average solution dimensionality of 4.0, which is a reasonable number for the size of the data considered. Trying to maximize other estimators for example, such as $\hat{F}_{\text{adj}}$ or $\hat{F}$, would result in very large dimensionalities.

5.3 Exemplary discoveries

After investigating the statistical properties of $\hat{F}_0$ and its algorithmic performance, we close this section with examples of concrete approximate functional dependencies discovered in two different applications: determining the winner of a tic-tac-toe configuration and predicting the preferred crystal structure of octet binary semi-conductors. Both settings are examples of problems where elementary input features are available, but to correctly represent the input/output relation either non-linear models have to be used or—if interpretable models are sought—complex auxiliary feature have to be constructed from the given elementary features.

The tic-tac-toe application [15] is one of the earliest examples of this complex feature construction problem. Tic-tac-toe is a game of two players where each player picks a symbol from $\{x, o\}$ and, taking turns, marks his symbol in an unoccupied cell of a 3×3 game board. A player wins the game if he marks 3 consecutive cells in a row, column, or diagonal. A game can end in draw, if the board configuration does not allow for any winning move. The dataset consists of 958 end game, winning configurations (i.e., there are no draws). The 9 input variables $\mathcal{I} = \{X_1, \ldots, X_9\}$ represent the cells of the board, and can have 3 values $\{x, o, b\}$, where b denotes an empty cell (see Fig. 6). The output variable Y with $V(Y) = \{\text{win}, \text{loss}\}$ is the outcome of the game for player x.

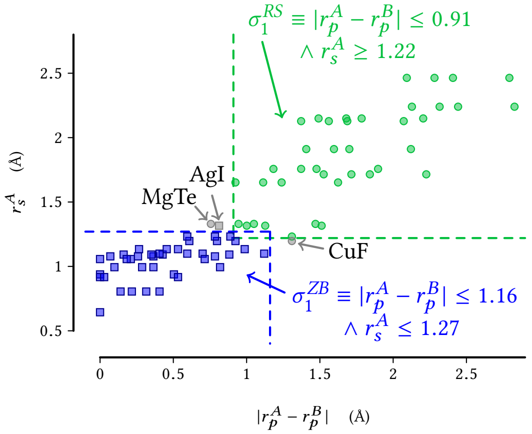


Figure 7: Binary semiconductors that crystalize as zinkblende (boxes) and rocksalt (circles); blue and green materials are correctly classified by subgroup-based prediction model—the involved rules (annotated) use elements of top functional dependency as two out of three variables. (source: Goldsmith et al. [8]).

Searching for approximate functional 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. 6). Knowing the state of these cell provides, therefore, a high amount of information about the outcome of the game. Moreover, removing a variable results in a loss of a considerable amount of information, while adding a variable would provide more information, but also redundancy. That is, the increase of fraction of information would not be higher than the increase of $\hat{b}_0$.

Our second example is a classical problem from Materials Science [24], which has meanwhile become a canonical example for the challenge of the automatic discovery of interpretable and "physically meaningful" prediction models of material properties [6, 8]. The task is to predict the symmetry or crystal structure, in which a given binary compound semi-conductor material will crystalize. That is, each of the 82 material involved consist of two atom types (A and B) and the output variable $Y = \{\text{rocksalt}, \text{zincblende}\}$ describes the crystal structure it prefers energetically. 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 approximate functional dependency with $\hat{F}_0 = 0.707$ and uncorrected empirical fraction of information $\hat{F} =$

0.735 is

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

i.e., the atomical *s* and *p* radii of component A. Again, this is a sensible finding, since these two variables constitute two out of three variables contained in the best structure prediction model that can be identified using the non-linear subgroup discovery approach [8]. Also both features are involved in the best linear LASSO model based on systematically constructed non-linear combinations of the elementary input variables [6]. The fact that not all variables of those models are identified by the functional dependency discovery algorithm can likely be explained by the facts that (a) the continuous input variables had to be discretized and (b) the dataset is extremely small with only 82 entries, which renders the discovery of reliable patterns with more than two variables very challenging.

6 CONCLUSION

We considered the dual problem of measuring and efficiently discovering approximate functional dependencies from data. We adopted an information theoretic approach, and proposed a fraction of information score that is reliable and achieves a good bias. In addition, we proposed an efficient optimistic estimator that allows for the effective discovery of the optimal, or α-approximate top-k dependencies of the target variable.

Although we carefully constructed the proposed correction term $\hat{b}_0$ such that bias under those regimes that are most problematic for searching for high-dimensional functional dependencies is removed, other scores could potentially be found that estimate the fraction of information with even less bias.

Other correction terms could also lead to other algorithms. The computational complexity for finding reliable functional dependencies is still open. Hence, polynomial time algorithm for this or adapted problem variants are a possibility. Similarly so, efficient tight(er) optimistic estimators would improve the runtime of branch-and-bound, as fewer nodes would have to be expanded to discover the optimal solution.

Both our score and optimistic estimator are specifically defined for discrete data. While in this paper we only considered univariate discrete targets, our scores can be trivially extended to multivariate discrete variables. Clearly, it is also of interest to discover approximate functional dependencies from continuous real-valued data. As entropy has been defined for such data, e.g. differential entropy [23] and cumulative entropy [18], it is possible to instantiate fraction of information scores. It will be interesting to see whether we can also efficiently correct these scores for chance, and whether optimistic estimators exist that allow for effective search.

ACKNOWLEDGMENTS

The authors wish to thank Luca Ghiringhelli and Brian Goldsmith for insightful discussions. Panagiotis Mandros is supported by the International Max Planck Research School for Computer Science (IMPRS-CS). The authors are supported by the Cluster of Excellence "Multimodal Computing and Interaction" within the Excellence Initiative of the German Federal Government.

REFERENCES

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

[2] Thomas M. Cover and Joy A. Thomas. 1991. *Elements of Information Theory.* Wiley-Interscience, New York, NY, USA.

[3] Mehmet M. Dalkilic and Edward L. Roberston. 2000. Information Dependencies. In *Proceedings of the 19th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems.* ACM, 245–253.

[4] Alin Dobra and Johannes Gehrke. 2001. Bias Correction in Classification Tree Construction. In *Proceedings of the 18th International Conference on Machine Learning (ICML), Williams College, MA.* Morgan Kaufmann, 90–97.

[5] Usama M. Fayyad and Keki B. Irani. 1993. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI), Chambéry, France.* 1022–1029.

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

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

[8] Bryan R Goldsmith, Mario Boley, Jilles Vreeken, Matthias Scheffler, and Luca M Ghiringhelli. 2017. Uncovering structure-property relationships of materials by subgroup discovery. *New Journal of Physics* 19, Article 013031 (2017), 14 pages.

[9] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification* 2, 1 (1985), 193–218.

[10] Ykä Huhtala, Juha Kärkkäinen, Pasi Porkka, and Hannu Toivonen. 1999. TANE: An efficient algorithm for discovering functional and approximate dependencies. *Comput. J.* 42, 2 (1999), 100–111.

[11] Igor Kononenko. 1995. On Biases in Estimating Multi-valued Attributes. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), San Francisco,CA,USA.* 1034–1040.

[12] Andreas Krause and Carlos E Guestrin. 2012. *Near-optimal nonmyopic value of information in graphical models.* Technical Report 1207.1394. arXiv.

[13] H.O. Lancaster. 1969. *The chi-squared distribution.* Wiley.

[14] Jixue Liu, Jiuyong Li, Chengfei Liu, and Yongfeng Chen. 2012. Discover dependencies from data–a review. *IEEE Transactions on Knowledge and Data Engineering* 24, 2 (2012), 251–264.

[15] Christopher J Matheus and Larry A Rendell. 1989. Constructive Induction On Decision Trees. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI), Detroit, MI.* 645.

[16] Kurt Mehlhorn and Peter Sanders. 2008. *Algorithms and data structures: The basic toolbox.* Springer Science & Business Media.

[17] Hoang Vu Nguyen, Panagiotis Mandros, and Jilles Vreeken. 2016. Universal Dependency Analysis. In *Proceedings of the SIAM International Conference on Data Mining (SDM), Miami, FL.* SIAM, 792–800.

[18] Murali Rao, Yunmei Chen, Baba C. Vemuri, and Fei Wang. 2004. Cumulative Residual Entropy: A New Measure of Information. *IEEE Transactions on Information Technology* 50, 6 (2004), 1220–1228.

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

[20] Simone Romano, James Bailey, Xuan Vinh Nguyen, and Karin Verspoor. 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.* 1143–1151.

[21] Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. 2016. A Framework to Adjust Dependency Measure Estimates for Chance. In *Proceedings of the SIAM International Conference on Data Mining (SDM), Miami, FL.* SIAM.

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

[23] Claude E. Shannon. 1948. A Mathematical Theory of Communication. *The Bell System Technical Journal* 27, 3 (1948), 379–423.

[24] James A Van Vechten. 1969. Quantum dielectric theory of electronegativity in covalent systems. I. Electronic dielectric constant. *Physical Review* 182, 3 (1969), 891.

[25] Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary?. In *Proceedings of the 26th International Conference on Machine Learning (ICML), Montreal, Canada.* ACM, 1073–1080.

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

[27] Yisen Wang, Simone Romano, Vinh Nguyen, James Bailey, Xingjun Ma, and Shu-Tao Xia. 2017. Unbiased Multivariate Correlation Analysis. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI).* AAAI.

[28] Allan P. White and Wei Zhong Liu. 1994. Technical Note: Bias in Information-Based Measures in Decision Tree Induction. *Machine Learning* 15, 3 (1994), 321–329.

ID	Name	#rows	#attrs.	#clases	α	time(s)	max dep.	sol. dep.	prune %
1	abalone	4174	8	28	1	2.1	8	3	0.00
2	appendicitis	106	7	2	1	1.0	6	3	50.00
3	tic	958	9	2	1	1.0	7	5	1.95
4	australian	690	14	2	1	3.7	11	5	75.01
5	bupa	345	6	2	1	1.0	1	1	96.88
6	car	1728	6	4	1	1.0	5	4	1.56
7	chess	3196	36	2	0.7	84.8	7	4	99.99
8	coil2000	9822	85	2	0.1	44.5	5	4	99.99
9	contraceptive	1473	9	3	1	1.0	7	4	75.00
10	ecoli	336	7	8	1	1.0	6	4	50.00
11	flare	1066	11	6	1	2.7	11	3	0.00
12	german	1000	20	2	1	76.9	11	7	97.29
13	glass	214	9	7	1	1.0	7	4	75.00
14	heart	270	13	2	1	1.0	9	5	75.34
15	ionosphere	351	33	2	1	1272.7	11	4	99.98
16	led7digit	500	7	10	1	1.0	7	5	0.00
17	lymphography	148	18	4	1	41.2	11	5	71.79
18	monk	432	6	2	1	1.0	4	3	10.94
19	movement-libras	360	90	15	0.4	31.2	5	3	99.99
20	nursery	12690	8	5	1	2.2	7	5	0.78
21	optdigits	5620	64	10	0.5	538.1	10	3	99.99
22	page	5472	10	5	1	4.7	8	4	7.23
23	penbased	10992	16	10	1	141.2	5	3	93.33
24	ring	7400	20	2	0.6	944.4	7	3	94.24
25	saheart	462	9	2	1	1.0	5	4	93.75
26	satimage	6435	36	7	0.9	850.9	5	3	99.99
27	segment	2310	19	7	1	35.3	8	2	98.37
28	sonar	208	60	2	1	744.7	13	6	99.99
29	spambase	4597	57	2	0.5	30.4	4	3	99.99
30	spectfheart	267	44	2	0.5	583.1	10	5	99.99
31	splice	3190	60	3	0.6	52.5	3	3	99.99
32	texture	5500	40	11	0.8	546.4	7	3	99.99
33	thyroid	7200	21	3	0.9	37.5	7	3	99.35
34	twonorm	7400	20	2	0.9	1160.2	6	4	94.74
35	vehicle	846	18	4	1	547.2	13	4	15.44
36	vowel	990	13	11	1	18.1	10	3	27.69
37	wdbc	569	30	2	1	326.1	10	4	99.96
38	wine	178	13	3	1	1.0	6	3	77.37
39	winequality red	1599	11	11	1	1.0	8	6	87.50
40	winequality white	4898	11	11	1	5.9	10	9	50.00
41	yeast	1484	8	10	1	1.0	7	7	50.00
42	zoo	101	15	7	1	2.3	9	5	84.41
Average		2800.0	24.0	5.6	0.89	194.0	7.5	4.0	67.9

Table 1: Datasets used in Section 5.2. The table contains information about the name and ID of the datasets used, number of rows, input variables, and classes, the α used for completion in less than 30 minutes, the time in seconds, the maximum depth of the algorithm, the depth of the best solution, and the percentage of the pruned search space.