Discovering Approximate Functional Dependencies using Smoothed Mutual Information

Frédéric Pennerath, Panagiotis Mandros, Jilles Vreeken

Discovering Approximate Functional Dependencies using Smoothed Mutual Information

Frédéric Pennerath
frederic.pennerath@centralesupelec.fr
Université de Lorraine,
CentraleSupélec, CNRS, LORIA
Metz, France

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

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

ABSTRACT

We consider the task of discovering the top-K reliable approximate functional dependencies $\mathbf{X} \rightarrow Y$ from high dimensional data. While naively maximizing mutual information involving high dimensional entropies over empirical data is subject to false discoveries, correcting the empirical estimator against data sparsity can lead to efficient exact algorithms for robust dependency discovery. Previous approaches focused on correcting by subtracting expected values of different null hypothesis models. In this paper, we consider a different correction strategy and counter data sparsity using uniform priors and smoothing techniques, that leads to an efficient and robust estimating process. In addition, we derive an admissible and tight bounding function for the smoothed estimator that allows us to efficiently solve via branch-and-bound the hard search problem for the top-K dependencies. Our experiments show that our approach is much faster than previous proposals, and leads to the discovery of sparse and informative functional dependencies.

CCS CONCEPTS

• **Information systems** → **Data mining**.

KEYWORDS

mutual information, functional dependencies, Laplace smoothing

1 INTRODUCTION

Discovering what subsets $\mathbf{X}$ of our descriptive variables $\mathcal{X}$ allow for an exact or approximate reconstruction of a given target variable Y is a core task in exploratory data mining [10, 12]. The example applications of **approximate functional dependency discovery** are many, as the presence of significant dependencies can provide valuable insight in the data generating process [4, 6], while the absence of such dependencies allows us to verify that the target cannot be reconstructed and is hence private within the data. Approximate functional dependency discovery is related, but different from three well known applications, namely key discovery in databases [9, 13], feature selection in machine learning [7], and Markov blanket discovery in Bayesian networks [21]. Unlike key discovery, we do not operate under a closed world assumption [5], and unlike feature selection, we are explicitly interested in discovering the top-K strongest joint dependencies, rather than pairwise associations and mostly greedy answers [2]. Lastly, Markov blankets are defined for Bayesian networks, while functional dependencies do not impose a structure (i.e., a DAG) for the data generating process and hence are more general and well-suited for exploratory data analysis.

Before we can discover the top-k strongest dependencies, we need a score for approximate functional dependencies from $\mathbf{X}$ to Y that allows for effective search. From information theory, we know that mutual information is the perfect score as it can detect dependencies regardless of their form [3]. Mutual information, however, is defined over *distributions* rather than over *data samples*, which means that in practice we have to estimate it. While asymptotically efficient [1], for limited sample sizes the empirical estimator has a positive bias [16] that is a function of the domain size of $\mathbf{X}$ and Y. This results to trivially identifying the complete set of variables $\mathcal{X}$ as a top dependency, regardless of whether $\mathcal{X}$ includes variables that are actually informative for Y [10]. To put it differently, the empirical estimator "breaks" under the data sparsity induced by the high-dimensional search space.

Recent work by Mandros et al. [10] and Nguyen et al. [12], address the *un-reliability* of the empirical estimator by subtracting expected values under the hypothesis of independence. While each of these improves over the empirical estimator, the discovery process can be inefficient in practice: Nguyen et al. bound the maximum search level meaning that all candidates below that level have to be evaluated which can be impractical for large dimensionalities, while the estimator proposed by Mandros et al. is based on sample permutations and has increased complexity.

In this paper, we solve the problem of discovering reliable top-k dependencies by exploring a different angle. Instead of subtracting mean values from null models, we correct the mutual information estimates using a uniform prior on the empirical distribution. This way, the joint distribution $P(\mathbf{X}, Y)$ for sparse data tends to uniformity, and therefore the inflated estimates of the empirical estimator are corrected. The resulting score is a Laplace smoothed mutual information estimator (SMI). We show that SMI is indeed reliable in practice, and that it can be employed to efficiently search for the top-k strongest dependencies with our recently proposed branch-and-bound framework [14] by deriving an admissible and tight bounding function for pruning.

In sum, the main contributions of this paper are as follows. We (1) propose the use of uniform priors to address the reliability problem and arrive at SMI as a score to mine statistically sound functional dependencies, (2) show that computing the exact upper bound of SMI leads to an NP-hard problem, (3) show how to instantiate a branch-and-bound algorithm that allows to efficiently discover the top-k dependencies scored by SMI thanks to an admissible, tight, yet tractable upper bound for SMI, and (4) empirically evaluate our approach in terms of runtime and statistical efficiency.

For readability and conciseness we postpone all proofs to the appendix. We make all code and data available in the supplementary.

2 SMOOTHED INFORMATION

Although we are not the first to consider estimation of mutual information using Bayesian approaches [17, 18], we are to the best of our knowledge the first to consider *smoothing* for reliable functional dependency discovery.[1] In this section we formally define smoothened mutual information (SMI) and study its properties. The computational problem of efficiently discovering the top-K dependencies based on this score is postponed to Sec. 3.

2.1 Notation and basic notions

We start with introducing the notation we will use throughout the paper. In general, we use upper cases letters X to denote random variables, and lower cases letters x for their values. Given a categorical variable Z, we denote its set of possible values, or *domain*, by $\mathbb{D}_Z$, and the number of possible values, or *cardinality* by $N_Z = |\mathbb{D}_Z|$. A set of variables $\mathbf{X}$ will be written in bold and conveniently represented by a vector containing variables X_i ordered by increasing order of rank i so that it can be unambiguously mapped to a vector $\mathbf{x}$ of corresponding values. A comma denotes vector concatenation between vectors, variables and values, e.g $(\mathbf{X}, \mathbf{Z}, Y)$ or $(\mathbf{x}, y)$. Notions of domain $\mathbb{D}_\mathbf{X}$ and number $N_\mathbf{X}$ of values are naturally extended to vectors of variables, where domains are cartesian products, i.e $\mathbb{D}_\mathbf{X} = \prod_{1\leq k\leq j} \mathbb{D}_{X_{i_k}}$. Given two sets of features $\mathbf{X}$ and $\mathbf{Z}$ such that $\mathbf{X} \subseteq \mathbf{Z}$ and given some value $\mathbf{x} \in \mathbb{D}_\mathbf{X}$, $\mathbb{D}_{\mathbf{Z}/\mathbf{x}}$ is the set of values $\mathbf{z} \in \mathbb{D}_\mathbf{Z}$ matching $\mathbf{x}$, i.e. whose restriction on variables of $\mathbf{X}$ is $\mathbf{x}$. The number $|\mathbb{D}_{\mathbf{Z}/\mathbf{x}}|$ of these values is denoted $N^\mathbf{x}_\mathbf{Z}$. Given some probability space, probability mass function of a random vector $\mathbf{Z}$ of features is denoted $P_\mathbf{Z}$. Its associated *entropy* is

$$\mathrm{H}(\mathbf{Z}) \stackrel{\text{def}}{=} \mathbb{E}\left(\log\left(\frac{1}{P_\mathbf{Z}}\right)\right) = -\sum_{\mathbf{z}\in\mathbb{D}_\mathbf{Z}} \mathrm{xlogx}\left(P_\mathbf{Z}(\mathbf{z})\right) \quad , \tag{1}$$

where xlogx denotes function $x \mapsto x\,\log(x)$, introduced for conciseness and $\log(x)$ refers to logarihm of x to an arbitrary base. Entropy computed from a set of counts $\mathbf{N} = (n_i)$ is also denoted

$$h(\mathbf{N}) = -\sum_{n\in\mathbf{N}} \mathrm{xlogx}\left(\frac{n}{\sum_{n'\in\mathbf{N}} n'}\right) \quad . \tag{2}$$

Mutual information shared between $\mathbf{X}$ and Y is then

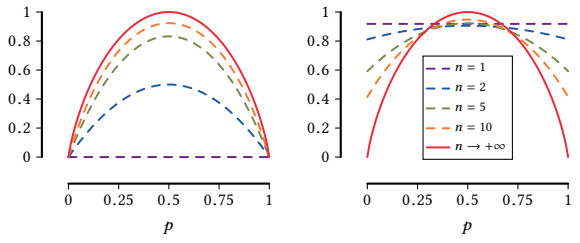
$$\begin{aligned} \mathrm{I}(\mathbf{X};Y) &\stackrel{\text{def}}{=} \mathrm{H}(Y) + \mathrm{H}(\mathbf{X}) - \mathrm{H}(\mathbf{X},Y) \quad . \\ &= \mathrm{H}(Y) - \mathrm{H}(Y|\mathbf{X}) \end{aligned} \tag{3}$$

Given a dataset and a set $\mathbf{X}$ of features, we denote by $n_\mathbf{x}$ the number of samples for which $\mathbf{X}$ is equal to $\mathbf{x}$. The total number of samples is denoted n. We can estimate entropy measures given a finite set of iid samples by replacing the oracle distributions $P_\mathbf{X}$, P_Y and $P_{\mathbf{X},Y}$ by their MLE estimators $\widehat{P}_\mathbf{X}$, $\widehat{P}_Y$ and $\widehat{P}_{\mathbf{X},Y}$ with

$$\widehat{P}_{\mathbf{X},Y}(\mathbf{x},y) = \frac{n_{\mathbf{x},y}}{n} \qquad \widehat{P}_\mathbf{X}(\mathbf{x}) = \frac{n_\mathbf{x}}{n} \qquad \widehat{P}_Y(y) = \frac{n_y}{n}$$
$$n = \sum_{\mathbf{x},y} n_{\mathbf{x},y} \qquad n_\mathbf{x} = \sum_y n_{\mathbf{x},y} \qquad n_y = \sum_\mathbf{x} n_{\mathbf{x},y} \quad .$$

We will mark such plugin estimators with a hat, e.g $\widehat{\mathrm{I}}(\mathbf{X};Y)$.

[1] The measures we present here are not be confused with the *smooth min or max entropies* as used in Quantum Information Theory.



(a) $p \mapsto \mathbb{E}\left(\widehat{\mathrm{H}}(Z|p)\right)$ **(b)** $p \mapsto \mathbb{E}\left(\tilde{\mathrm{H}}^{(1)}(Z|p)\right)$

Figure 1: Expected values (in bits) of plugin entropy $\widehat{\mathrm{H}}(Z|p)$ (left) and smoothed entropy $\tilde{\mathrm{H}}^{(1)}(Z|p)$ for $\alpha = 1$ (right) for different number of samples n from a binomial distribution $\mathcal{B}(n,p)$. The plugin underestimates under sparsity, while smoothing shows a positive bias that helps with overfitting.

2.2 Problem of Overfitting

We consider the problem of finding strong *approximate functional dependencies* $\mathbf{X} \to Y$ for a given target Y, such that once the value of $\mathbf{X}$ is known, the remaining uncertainty about value of Y is minimal. Given that we only have a finite set of samples, we can only approximate the true joint distribution $P_{\mathbf{X},Y}$. Although commonplace, simply replacing $\mathrm{I}(\mathbf{X};Y)$ by its plugin estimator $\widehat{\mathrm{I}}(\mathbf{X};Y)$ is a very bad solution in practice, as for limited sample sizes both $\widehat{\mathrm{H}}(X)$ and $\widehat{\mathrm{H}}(X \mid Y)$ tend to strongly underestimated the true entropy.

We illustrate this with an example in Fig. 1a, where Z is a Bernoulli variable of parameter p. We have the expected value $\mathbb{E}\left(\widehat{\mathrm{H}}(Z \mid p)\right)$ of the plugin entropy with respect to the binomial distribution $\mathcal{B}(n,p)$ of n iid samples as

$$\mathbb{E}\left(\widehat{\mathrm{H}}(Z|p)\right) = \sum_{i=0}^{n} \binom{n}{i} p^i\,(1-p)^{n-i} h([i, n-i]) \quad . \tag{4}$$

As Fig. 1a shows, the curves for $n < \infty$ are all below the true entropy. That is, the plugin entropy $\widehat{\mathrm{H}}(Z|p)$ exhibits a strong negative bias whenever the number n of samples is small, with the extreme case of $\widehat{\mathrm{H}}(Z)$ always being 0 whenever $n = 1$. Obviously, such a strong statement cannot be drawn from a single observation. Although no-one would voluntarily measure entropy over a single sample, in practice we do often have to estimate entropy under sparsity: as the domain of a set of features $\mathbf{X}$ grows exponentially in the size of the set, the number of samples we have to measure the conditional entropy of Y given a value instantiation of $\mathbf{X}$, $\widehat{\mathrm{H}}(Y|\mathbf{X} = \mathbf{x})$, quickly tends to 0. Because expressions of $\widehat{\mathrm{H}}(Y|\mathbf{X})$ and $\widehat{\mathrm{I}}(\mathbf{X};Y)$ result from averaging entropy terms $\widehat{\mathrm{H}}(Y|\mathbf{X} = \mathbf{x})$, these measures are systematically overconfident. In particular, we have that $\widehat{\mathrm{H}}(Y|\mathbf{X})$ (resp. $\widehat{\mathrm{I}}(\mathbf{X};Y)$) has a negative (resp. positive) bias that increases with the size of $\mathbf{X}$. Romano et al. [15] dubbed this the "just by chance" effect, also known as overfitting in Machine Learning. In contrast to existing approaches subtracting expected values of null hypothesis models, we propose to tackle this problem by "smoothing" data sparsity using a flat prior.

2.3 Laplace Smoothing

Laplace smoothing is a well-known shrinkage technique to reduce variance when estimating categorical distributions from samples: given a variable Z following a categorical distribution P_Z of unknown probabilities $(p_1, \ldots, p_{N_Z})$ and given a dataset of n iid samples, let $(n_1, \ldots, n_{N_Z})$ be the histogram of corresponding values of Z occurring in the dataset. Then the α-smoothed estimator $\tilde{P}_Z^{(\alpha)}$ of distribution P_Z for some $\alpha \geq 0$ is defined by

$$\tilde{P}_Z^{(\alpha)}(z) = \frac{n_z + \alpha}{n + N_Z \times \alpha} \quad . \tag{5}$$

Setting $\alpha = 0$ gives the standard maximum likelihood estimator (MLE) which equates probabilities of events with their frequencies of occurrence in data. The parameter α is called *pseudocount* since when α is an integer, it simulates the occurrence of α extra samples for each possible value of Z. The larger α, the smaller the variance of the estimator but the larger its bias. Put another way, these pseudocounts act as a regularizer.

Laplace smoothing is actually a sound consequence of Bayesian estimation since when a symmetric Dirichlet distribution with a concentration parameter of $\alpha + 1$ is chosen as a conjugate prior for distribution of Z, smoothed estimator $\tilde{P}_Z^{(\alpha)}$ of Eq. 5 coincides with the MAP estimator, i.e. the mode of a posteriori Dirichlet distribution conditioned on samples. Therefore, like any MAP estimator, smoothed distributions can be seen as regularized estimators moderating randomness of observations by prior knowledge. In our case, this prior knowledge is more a "prior ignorance" since the α-symmetric Dirichlet prior tends to privilege uniform distributions.

We obtain both *smoothed entropy* and *smoothed mutual information* (SMI) by first applying Laplace smoothing to empirical distributions and then applying estimators to the resulting smoothed distribution. We write $\tilde{\mathrm{H}}^{(\alpha)}(Z)$ for $\mathrm{H}\left(\tilde{P}_Z^{(\alpha)}\right) = h((n_z + \alpha)_z)$. Importantly, these smoothed estimators we so obtain are statistically *consistent*, i.e they tend to the true measure whenever the number n of samples tends to infinity. Whenever n is finite, on the other hand, they benefit from a regularization effect; compared to plugin entropy, smoothed entropy (resp. smoothed mutual information) has a positive (resp. negative) bias, that integrates uncertainty due to the partial knowledge of the true distribution (sometimes called epistemic uncertainty).

This is clearly visible on Fig. 1b, where the expected value of $\mathbb{E}\left(\tilde{\mathrm{H}}^{(1)}(Z|p)\right)$ is obtained by simply replacing the term $h([i, n-i])$ in Eq. 4 with $h([i+\alpha, n-i+\alpha])$. Like it's plug-in counterpart, smoothed entropy $\tilde{\mathrm{H}}^{(1)}(Z|p)$ is also constant when $n = 1$, however, rather than taking a minimal value, smoothed entropy takes an almost maximal value of 0.91 bits. Not only are the smoothed scores consistent with the number of observations, they are also regularized and hence have much lower variance compared to the plugin. We illustrate this in Fig. 2, where we indicate for different sample sizes n the expected values and variances of both the plugin (blue) and the smoothed estimator (green). As the figure shows, the statistical fluctuations for the smoothed score are lower than that of the plugin, which as we will see allows for robust optimization.

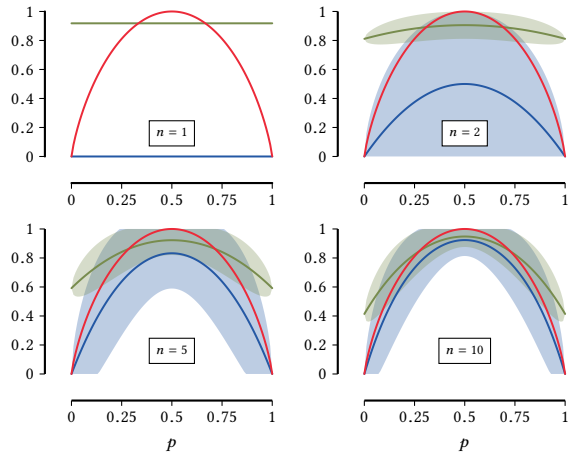


Figure 2: Expected value and variance of plugin entropy $\widehat{\mathrm{H}}(Z \mid p)$ (blue) and smoothed entropy $\tilde{\mathrm{H}}^{(1)}(Z \mid p)$ with $\alpha = 1$ (green) with respect to binomial distribution $\mathcal{B}(n, p)$ of n iid samples of a Bernoulli variable Z of parameter p as a function of p, for different values of n. Red line is the true entropy $H(Z \mid p)$. Expected values for the estimators are given by blue and green lines, and their variance σ^2 as blue resp. green shaded areas made of 2σ-wide confidence intervals centered on expected value.

	$\mathbf{x}_1$	$\mathbf{x}_2$	...	Total
y_1	$n_{\mathbf{x}_1, y_1} + \alpha$	$n_{\mathbf{x}_2, y_1} + \alpha$	...	$n_{y_1} + N_{\mathrm{X}}\alpha$
y_2	$n_{\mathbf{x}_1, y_2} + \alpha$	$n_{\mathbf{x}_2, y_2} + \alpha$	...	$n_{y_2} + N_{\mathrm{X}}\alpha$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
Total	$n_{\mathbf{x}_1} + N_Y\alpha$	$n_{\mathbf{x}_2} + N_Y\alpha$	...	$n + N_{\mathrm{X}}N_Y\alpha$

Table 1: Notations for counts in contingency table of X versus Y including pseudocounts.

2.4 Problem statement

With the above definition and preliminary analysis of SMI, we can now formally state the problem at hand.

Problem 1 (Top-K Functional Dependency Discovery). *Given a dataset D over descriptive variables $\mathbf{Z}$ and target variable Y, a coefficient α and a number K, find the K sets $\{\mathbf{X}_i\}_{1 \leq i \leq K} \subseteq \mathbf{Z}$ of descriptive variables that achieve the highest α-smoothed mutual information $\tilde{\mathrm{I}}^{(\alpha)}(\mathbf{X}; Y)$ with Y.*

To solve this problem, we first derive the exact expression of $\tilde{\mathrm{I}}^{(\alpha)}(\mathbf{X}; Y)$. Given some smoothed joint distribution and its contingency table illustrated on Tab.1 , we first observe that rows and columns (i.e. marginal distributions) sum up pseudocounts the same way they sum up real samples. We formalize this as follows.

Property 1. *Smoothed marginals*

$$\tilde{P}_{\mathbf{X}}^{(N_Y\,\alpha)}(\mathbf{x}) = \sum_{y\in\mathbb{D}_Y} \tilde{P}_{\mathbf{X},Y}^{(\alpha)}(\mathbf{x},y)$$
$$\tilde{P}_{Y}^{(N_\mathbf{X}\alpha)}(y) = \sum_{\mathbf{x}\in\mathbb{D}_\mathbf{X}} \tilde{P}_{\mathbf{X},Y}^{(\alpha)}(\mathbf{x},y) \ .$$

We can now straightforwardly derive the following forms of smoothed entropy and smoothed mutual information.

Property 2. *Smoothed entropy measures*

$$\begin{aligned}\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{X}) &= \tilde{\mathrm{H}}^{(\alpha)}(\mathbf{X},Y) - \tilde{\mathrm{H}}^{(N_Y\,\alpha)}(\mathbf{X})\\ \tilde{\mathrm{I}}^{(\alpha)}(\mathbf{X};Y) &= \tilde{\mathrm{H}}^{(N_\mathbf{X}\alpha)}(Y) - \tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{X})\\ &= \tilde{\mathrm{H}}^{(N_Y\,\alpha)}(\mathbf{X}) + \tilde{\mathrm{H}}^{(N_\mathbf{X}\alpha)}(Y) - \tilde{\mathrm{H}}^{(\alpha)}(\mathbf{X},Y) \ .\end{aligned}$$

Now that Problem 1 is fully specified, we will consider the computational problem of optimizing it next.

3 ALGORITHM

In this section we discuss how we can exactly yet efficiently discover the top-K functional dependency problem using branch-and-bound.

3.1 Introduction

Because of smoothing, SMI sadly has no structure—such as convexity, monotonicity, modularity—that we can exploit for efficient optimization. To illustrate the difference in behavior between SMI and standard mutual information, let us consider a dataset over arbitrarily many features $\mathbf{Z}$, of a finite set of n samples. It is easy to see that when we increase the number of selected features $\mathbf{X} \subseteq \mathbf{Z}$, the product $N_\mathbf{X}\,\alpha$ tends to $+\infty$ for any $\alpha > 0$. In other words, the smoothed distribution over $\mathbf{X}$ and Y, $\tilde{P}_Y^{(N_\mathbf{X}\,\alpha)}$ tends to the uniform distribution, and by an argument of continuity, the smoothed entropy $\tilde{\mathrm{H}}^{(N_\mathbf{X}\,\alpha)}(Y)$ tends to $\log(N_Y)$. As we can rewrite conditional entropy, $\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{X})$, as an average of smoothed entropy terms

$$\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{X}) = \sum_{\mathbf{x}} \frac{n_\mathbf{x} + N_Y\alpha}{n + N_\mathbf{X}N_Y\alpha}\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{X}=\mathbf{x}) \ ,$$

and the relative number of "real counts" of $\tilde{P}^{(\alpha)}_{Y|\mathbf{X}=\mathbf{x}}$ gets smaller and smaller relative to the (uniformly spread) pseudocounts α, $\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{X})$ also tends to $\log(N_Y)$. As SMI is defined as the difference of these terms, it tends to zero. As a direct result of these observations, we have that if we start from a singleton feature X, the score will increase whenever we add a relevant feature, up to the point where the smoothing effect of α is not negligible anymore, after which the score will decrease and tend to zero.

An approach to exactly solve such non trivial optimization problems is the branch and bound (BnB) search scheme, in which we prune the current exploration branch as soon as we can infer that there does not exist any pattern we can reach via this branch that is better than the worst of the top-K patterns we have discovered so far. To effectively use BnB we need an upper bound $\bar{\mathrm{I}}$ on SMI that is 1) *tight* as possible (as then we have maximum pruning power) while 2) being *efficiently computable* (as we will have to compute it very often), but most importantly, 3) be *admissible* (as else we cannot guarantee that we return the true top-K). That is, we require

$$\forall \mathbf{Z},\ \mathbf{X}\subseteq\mathbf{Z} \implies \tilde{\mathrm{I}}^{(\alpha)}(\mathbf{Z};Y) \le \bar{\mathrm{I}}(\mathbf{X}) \ ,$$

as then and only then an exploration branch can safely be pruned whenever $\bar{\mathrm{I}}(\mathbf{X}) \le \tilde{\mathrm{I}}^{(\alpha)}(\mathbf{X}_K;Y)$ where $\mathbf{X}_K$ is the K^{th} last ranked top-K pattern found so far.

From here on, we will follow the convention where $\mathbf{X}$ denotes the set of features that have so far been considered in the current branch, while $\mathbf{Z} \supseteq \mathbf{X}$ denotes any set of features containing $\mathbf{X}$.

3.2 Problem decomposition

It is easy to see that the tightest admissible bound $\bar{\mathrm{I}}_{opt}(\mathbf{X})$ is given by the maximal smoothed mutual information $\tilde{\mathrm{I}}^{(\alpha)}(\mathbf{Z};Y)$ that is reachable in the current branch, and that can be computed using only the information we used to compute score of just $\mathbf{X}$, i.e. the counts $(n_{\mathbf{x},y})$. We can formalize the computation of this bound as the following constrained optimization problem.

Problem 2. *Given α and counts $(n_{\mathbf{x},y})$ for all values $\mathbf{x}$ of $\mathbf{X}$ and y of Y, find the number $N_\mathbf{Z}$ of values of $\mathbf{Z}$ and the $N_\mathbf{Z}\times N_Y$ contingency table $(n_{\mathbf{z},y})$ for all values $\mathbf{z}$ and y, maximizing SMI with Y, i.e.*

$$\bar{\mathrm{I}}_{opt}(\mathbf{X}) \overset{def}{=} \max_{(n_{\mathbf{z},y})} \left(\tilde{\mathrm{I}}^{(\alpha)}(\mathbf{Z};Y)\right) \ ,$$

given the $N_\mathbf{X}\times N_Y$ constraints

$$\forall \mathbf{x},\ \forall y, \quad n_{\mathbf{x},y} = \sum_{\mathbf{z}\in\mathbb{D}_{\mathbf{Z}/\mathbf{x}}} n_{\mathbf{z},y} \ ,$$

and where

$$\tilde{\mathrm{I}}^{(\alpha)}(\mathbf{Z};Y) = \tilde{\mathrm{H}}^{(N_\mathbf{Z}\,\alpha)}(Y) - \tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z}) \tag{6}$$

with $$\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z}) = \sum_{\mathbf{z}} \frac{n_\mathbf{z} + N_Y\,\alpha}{n + N_\mathbf{Z}\,N_Y\,\alpha}\,\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z}=\mathbf{z})$$

and $$\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z}=\mathbf{z}) = -\sum_{y} \mathrm{xlogx}\left(\frac{n_{\mathbf{z},y}+\alpha}{n_\mathbf{z}+N_Y\,\alpha}\right) \ .$$

Each constraint here has a variable number $N_\mathbf{Z}^\mathbf{x} \ge 1$ of unknowns such that

$$N_\mathbf{Z} = \sum_{\mathbf{x}} N_\mathbf{Z}^\mathbf{x} \ .$$

Because each value $\mathbf{x}$ is mapped to N_Y constraints that only relate to part of the dataset matching $\mathbf{x}$, the term $\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z})$ is better decomposed in a way that makes the partition of the dataset induced by the different values of $\mathbf{X}$ explicit.

$$\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z}) = \sum_{\mathbf{x}} \frac{n_\mathbf{x} + N_\mathbf{Z}^\mathbf{x}\,N_Y\,\alpha}{n + N_\mathbf{Z}\,N_Y\,\alpha}\,\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z},\mathbf{X}=\mathbf{x}) \tag{7}$$

with $$\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z},\mathbf{X}=\mathbf{x}) \overset{\text{def}}{=} \sum_{\mathbf{z}\in\mathbb{D}_{\mathbf{Z}/\mathbf{x}}} \frac{n_\mathbf{z} + N_Y\,\alpha}{n_\mathbf{x} + N_\mathbf{Z}^\mathbf{x}\,N_Y\,\alpha}\,\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z}=\mathbf{z}) \ .$$

These latter terms $\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z},\mathbf{X}=\mathbf{x})$ are "local" in the sense that $\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z},\mathbf{X}=\mathbf{x})$ only depends on counts $n_{\mathbf{z},y}$ for $\mathbf{z}\in\mathbb{D}_{\mathbf{Z}/\mathbf{x}}$ and on the subset of the N_Y previous constraints mapped to value $\mathbf{x}$. In other words, it only depends on the subset of samples matching

value $\mathbf{x}$. Therefore the initial optimization problem can be decomposed in two nested layers where the inner layer consists of "local" optimization problems mapped to each value $\mathbf{x}$ of $\mathbf{X}$ and where the outer layer is a single "global" optimization problem merging solutions of the inner layer. More formally, using Eq. 6 and 7,

$$\begin{aligned}\tilde{\mathrm{I}}_{opt}(\mathbf{X}) &= \max_{(N_{\mathbf{Z}}^{\mathbf{x}}),(n_{\mathbf{z},y})} \left(\tilde{\mathrm{I}}^{(\alpha)}(\mathbf{Z};Y)\right) \\ &= \max_{(N_{\mathbf{Z}}^{\mathbf{x}})} \left(\tilde{\mathrm{H}}^{(N_{\mathbf{Z}}\,\alpha)}(Y) - \min_{(n_{\mathbf{z},y})} \left(\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z})\right)\right) \\ &= \max_{(N_{\mathbf{Z}}^{\mathbf{x}})} \left(\tilde{\mathrm{H}}^{(N_{\mathbf{Z}}\,\alpha)}(Y) - \sum_{\mathbf{x}} \frac{n_{\mathbf{x}} + N_{\mathbf{Z}}^{\mathbf{x}}\, N_Y\, \alpha}{n + N_{\mathbf{Z}}\, N_Y\, \alpha} \underline{\tilde{H}}_{opt}^{(\alpha,\mathbf{x})}(N_{\mathbf{Z}}^{\mathbf{x}})\right) \quad , \end{aligned} \tag{8}$$

where the lower bound $\underline{\tilde{H}}_{opt}^{(\alpha,\mathbf{x})}(N)$ denotes the solution of the "local" optimization problem mapped to value $\mathbf{x}$ when the number $N_{\mathbf{Z}}^{\mathbf{x}}$ is set to a given constant N and where these "local" optimization problems are defined as follows:

Problem 3. *Given α, a value $\mathbf{x}$, N_Y counts $(n_{\mathbf{x},y})$ and a fixed number $N_{\mathbf{Z}}^{\mathbf{x}} = N \geq 1$ of values $\mathbf{z}$ matching $\mathbf{x}$, find the local optimal bound*

$$\underline{\tilde{H}}_{opt}^{(\alpha,\mathbf{x})}(N) \stackrel{def}{=} \min_{(n_{\mathbf{z},y})} \left(\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z},\mathbf{X}=\mathbf{x})\right) \quad ,$$

reached by the $N_{\mathbf{Z}}^{\mathbf{x}} \times N_Y$ contingency table $(n_{\mathbf{z},y})$ minimizing the "local" smoothed conditional entropy $\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z},\mathbf{X}=\mathbf{x})$ while satisfying the N_Y constraints

$$\forall y,\ n_{\mathbf{x},y} = \sum_{\mathbf{z}} n_{\mathbf{z},y} \quad .$$

While this problem admits a trivial solution for standard entropy (i.e for $\alpha = 0$), finding its counterparts for $\alpha > 0$ is anything but straightforward. To illustrate this, we consider the simple example of $N_Y = 4$ with counts $(n_{\mathbf{x},y})_y = (4,2,1,1)$ for some value $\mathbf{x}$. In the plugin case (i.e. $\alpha = 0$), the optimal contingency tables are those that perfectly separate all values of Y, i.e. $N_{\mathbf{Z}}^{\mathbf{x}} = N_Y$, such as the example depicted as Tab. 1a of which the entropy $\mathrm{H}(Y|\mathbf{Z},\mathbf{X}=\mathbf{x})$ is zero. However, when we set $\alpha = 1$, the entropy $\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z},\mathbf{X}=\mathbf{x})$ of this table increases to 1.77 bits, and now it is the table as Tab. 1b that achieves the minimum (1.73 bits). The optimal partition changes again when we set $\alpha \geq 2$, and is depicted as Tab. 1c.

This example illustrates that when α increases, the optimal number $N_{\mathbf{Z}}^{\mathbf{x}}$ of values for $\mathbf{Z}$ decreases. Intuitively, the weight of pseudocounts relatively to the weight of "real" counts $(n_{\mathbf{z},y})$ increases with α and the number of columns. Since pseudocounts are uniformly distributed they also tend to increase the global entropy. Therefore when α increases, optimality tends to merge real counts in fewer columns to counterbalance the increasing weight of pseudocounts.

This illustrates that Problem 3 does not have a trivial solution, which is a pity. The worse news is that the exact computation of the upper bound is NP-hard. Before we formally show this in Sec. 3.6, it will be helpful to first introduce an admissible *local* bound.

3.3 Local Bound

Instead of using the exact upper bound, we can also use an estimator that solves approximately Problem 3 by deriving an admissible local

(a) $\alpha = 0$

	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	$\mathbf{z}_4$
y_1	4	0	0	0
y_2	0	2	0	0
y_3	0	0	1	0
y_4	0	0	0	1

(b) $\alpha = 1$

	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$
y_1	4	0	0
y_2	0	2	0
y_3	0	0	1
y_4	0	0	1

(c) $\alpha = 2$

	$\mathbf{z}_1$	$\mathbf{z}_2$
y_1	4	0
y_2	0	2
y_3	0	1
y_4	0	1

Table 2: Example contingency tables minimizing $\tilde{\mathrm{H}}^{(\alpha)}(Y \mid \mathbf{Z},\mathbf{X}=\mathbf{x})$ when $N_Y = 4$ and $(n_{\mathbf{x},y})_y = (4,2,1,1)$, for different values of α. Pseudocounts are omitted for clarity.

lower bound $\underline{\tilde{H}}_{app}^{(\alpha,\mathbf{x})}(N)$ for conditional entropy, i.e. such that

$$\underline{\tilde{H}}_{app}^{(\alpha,\mathbf{x})}(N) \leq \underline{\tilde{H}}_{opt}^{(\alpha,\mathbf{x})}(N) \quad .$$

The main idea is to rewrite $\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z},\mathbf{X}=\mathbf{x})$ as the uncontrolled entropy term we aim to minimize, in terms of $\tilde{\mathrm{H}}^{(\alpha)}(\mathbf{Z}|Y,\mathbf{X}=\mathbf{x})$ of which we *can* express its optimal value as a function of known counts $n_{\mathbf{x},y}$. To this end, we consider the symmetric definition of mutual information $\tilde{\mathrm{I}}^{(\alpha)}(\mathbf{Z};Y|\mathbf{X}=\mathbf{x})$:

$$\begin{aligned}\underline{\tilde{H}}_{opt}^{(\alpha,\mathbf{x})}(N) &= \min_{\substack{(n_{\mathbf{z},y}) \\ \forall y,\ \sum n_{\mathbf{z},y} = n_{\mathbf{x},y}}} \left(\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z},\mathbf{X}=\mathbf{x})\right) \\ &= \min_{\substack{(n_{\mathbf{z},y}) \\ \forall y,\ \sum n_{\mathbf{z},y} = n_{\mathbf{x},y}}} \left(\tilde{\mathrm{H}}^{(N\,\alpha)}(Y|\mathbf{X}=\mathbf{x})) - \tilde{\mathrm{I}}^{(\alpha)}(\mathbf{Z};Y|\mathbf{X}=\mathbf{x}))\right) \\ &= \tilde{\mathrm{H}}^{(N\,\alpha)}(Y|\mathbf{X}=\mathbf{x}) \quad + \\ &\qquad \min_{\substack{(n_{\mathbf{z},y}) \\ \forall y,\ \sum n_{\mathbf{z},y} = n_{\mathbf{x},y}}} \left(\tilde{\mathrm{H}}^{(\alpha)}(\mathbf{Z}|Y,\mathbf{X}=\mathbf{x}) - \tilde{\mathrm{H}}^{(N_Y\,\alpha)}(\mathbf{Z}|\mathbf{X}=\mathbf{x})\right) \end{aligned} \tag{9}$$

Although computing this minimum is NP-hard, we can obtain a lower bound of it using the general property that for every pair of real functions f and g with the same domain,

$$\min_w(f(w)) + \min_w(g(w)) \leq \min_w(f(w) + g(w)) \quad . \tag{10}$$

Straightforwardly applying this to Eq. 9 gives an approximate admissible local bound $\underline{\tilde{H}}_{app}^{(\alpha,\mathbf{x})}(N)$, i.e.

$$\underline{\tilde{H}}_{app}^{(\alpha,\mathbf{x})}(N) \leq \underline{\tilde{H}}_{opt}^{(\alpha,\mathbf{x})}(N) \quad ,$$

where

$$\underline{\tilde{H}}_{app}^{(\alpha,\mathbf{x})}(N) \stackrel{\text{def}}{=} \tilde{\mathrm{H}}^{(N\,\alpha)}(Y|\mathbf{X}=\mathbf{x}) + \underline{\tilde{H}}_1^{(\alpha,\mathbf{x})}(N) - \overline{\tilde{H}}_2^{(\alpha,\mathbf{x})}(N) \quad , \tag{11}$$

with

$$\underline{\tilde{H}}_1^{(\alpha,\mathbf{x})}(N) \stackrel{\text{def}}{=} \min_{\substack{(n_{z,y}) \\ \forall y,\, \sum n_{z,y}=n_{\mathbf{x},y}}} \left(\tilde{\mathrm{H}}^{(\alpha)}(\mathbf{Z}|Y,\mathbf{X}=\mathbf{x})\right)$$

$$\overline{\tilde{H}}_2^{(\alpha,\mathbf{x})}(N) \stackrel{\text{def}}{=} \max_{\substack{(n_{z,y}) \\ \forall y,\, \sum n_{z,y}=n_{\mathbf{x},y}}} \left(\tilde{\mathrm{H}}^{(N_Y\,\alpha)}(\mathbf{Z}|\mathbf{X}=\mathbf{x})\right) \ .$$

On the one hand, $\overline{\tilde{H}}_2^{(\alpha,\mathbf{x})}(N)$ is the maximal reachable value for smoothed entropy $\tilde{\mathrm{H}}^{(N_Y\,\alpha)}(\mathbf{Z}|\mathbf{X}=\mathbf{x})$. This maximum is straightforwardly reached when $\tilde{P}^{(\alpha)}_{\mathbf{Z}|\mathbf{X}=\mathbf{x}}$ is uniform, so

$$\overline{\tilde{H}}_2^{(\alpha,\mathbf{x})}(N) = \log(N) \ . \tag{12}$$

Note that while optimistic in general, $\overline{\tilde{H}}_2^{(\alpha,\mathbf{x})}(N)$ is in fact sometimes reachable, such as on Fig. 1c for $N_{\mathbf{Z}} = 2$ (since all columns sum up to the same total of 4), but often unreachable, such as in Fig. 1b for $N_{\mathbf{Z}} = 3$ (since counts $4, 2, 1, 1$ cannot be equally distributed over the three columns).

On the other hand, $\underline{\tilde{H}}_1^{(\alpha,\mathbf{x})}(N)$ is the minimal reachable value of $\tilde{\mathrm{H}}^{(\alpha)}(\mathbf{Z}|Y,\mathbf{X}=\mathbf{x})$ whose expression relies on the next property.

Property 3. *Given some random variable W known to take a given number N_W of possible values (observed or not), and considering all values of the α-smoothed entropy estimator $\tilde{\mathrm{H}}^{(\alpha)}(W)$ computed from all possible sequences of n samples for W, then the minimal value is reached when all samples have the same value, so that*

$$\min_{\substack{(n_w)\in\mathbb{N}^{N_W} \\ \sum_w n_w = n}} \left(\tilde{\mathrm{H}}^{(\alpha)}(W)\right) = \tilde{h}^{(\alpha)}_{N_W}(n) \ ,$$

with $\tilde{h}^{(\alpha)}_{N_W}(n) = h([n+\alpha, \underbrace{\alpha,\ldots,\alpha}_{\times N_W - 1}])$

$$= -\operatorname{xlogx}\left(\frac{n+\alpha}{n+N_W\,\alpha}\right) - (N_W-1)\,\operatorname{xlogx}\left(\frac{\alpha}{n+N_W\,\alpha}\right) \ .$$

We postpone the proof to Appendix 6.2.1. Following from this property, we have that the minimal smoothed entropy is reached by the same deterministic distributions as standard entropy. That is, when all counts n_w are equal to zero but one. However, whereas the minimal value of standard entropy is zero, the minimal value $\tilde{h}^{(\alpha)}_{N_W}(n)$ of our smoothed entropy is a strictly positive function decreasing with n and increasing with N_W. For instance, given some value y of Y, each of the three lines mapped to y in each table of Fig. 2 is such a deterministic distribution $\tilde{P}^{(\alpha)}_{\mathbf{Z}|Y=y,\mathbf{X}=\mathbf{x}}$, thus reaching its minimal bound $\tilde{h}^{(\alpha)}_{N_{\mathbf{Z}}}(n_{\mathbf{x},y})$.

Using Prop. 3, we can rewrite $\underline{\tilde{H}}_1^{(\alpha,\mathbf{x})}(N)$ as

$$\underline{\tilde{H}}_1^{(\alpha,\mathbf{x})}(N) = \sum_y \frac{n_{\mathbf{x},y}+N\,\alpha}{n_{\mathbf{x}}+N\,N_Y\,\alpha} \min_{\substack{(n_{z,y}) \\ \sum n_{z,y}=n_{\mathbf{x},y}}} \left(\tilde{\mathrm{H}}^{(\alpha)}(\mathbf{Z}|Y=y,\mathbf{X}=\mathbf{x})\right)$$

$$= \sum_y \frac{n_{\mathbf{x},y}+N\,\alpha}{n_{\mathbf{x}}+N\,N_Y\,\alpha}\,\tilde{h}^{(\alpha)}_N(n_{\mathbf{x},y}) \ . \tag{13}$$

Finally, combining Eq. 11, 12 and 13 gives us an efficiently computable and admissible local bound

$$\underline{\tilde{H}}^{(\alpha,\mathbf{x})}_{app}(N) = \log\left(\frac{n_{\mathbf{x}}+N\,N_Y\,\alpha}{N}\right) - \frac{\left(\sum_y \operatorname{xlogx}\left(n_{\mathbf{x},y}+\alpha\right)\right) + (N-1)\,N_Y\,\operatorname{xlogx}(\alpha)}{n_{\mathbf{x}}+N\,N_Y\,\alpha} \ , \tag{14}$$

that we can use in a branch-and-bound algorithm.

3.4 Global Bound

If we inject Eq. 14 into Eq. 8, skipping the intermediate results, we obtain a global bound $\bar{\mathrm{I}}_{app}$ defined as

$$\bar{\mathrm{I}}_{app}(\mathbf{X}) = \max_{N_{\mathbf{Z}} \geq N_{\mathbf{X}}} (h(N_{\mathbf{Z}})) \tag{15}$$

$$\text{with } h(N_{\mathbf{Z}}) = \log(N_{\mathbf{Z}}) + \frac{c_0 - \left(\sum_y \operatorname{xlogx}(n_y + N_{\mathbf{Z}}\,\alpha)\right) + N_{\mathbf{Z}}\,N_Y\,\operatorname{xlogx}(\alpha)}{n + N_{\mathbf{Z}}\,N_Y\,\alpha} \ ,$$

and where c_0 is a constant equal to

$$c_0 = \left(\sum_{\mathbf{x},y} \operatorname{xlogx}(n_{\mathbf{x},y}+\alpha)\right) - N_{\mathbf{X}}\,N_Y\,\operatorname{xlogx}(\alpha) \ .$$

Details of the derivation are given in Appendix 6.2.2. Expression 15 only depends on a single scalar, i.e. sum $N_{\mathbf{Z}} = \sum_{\mathbf{x}} N^{\mathbf{x}}_{\mathbf{Z}}$, and not on the whole vector of variables $N^{\mathbf{x}}_{\mathbf{Z}}$. Bound $\bar{\mathrm{I}}_{app}(\mathbf{X})$ is thus the maximum value of function $N_{\mathbf{Z}} \mapsto h(N_{\mathbf{Z}})$. Since $\frac{d^2h}{dN_{\mathbf{Z}}^2} < 0$, this function h is concave so it has a unique maximum, which is necessarily $\bar{\mathrm{I}}_{app}(\mathbf{X})$. While this maximum has no simple analytic expression, $\frac{dh}{dN_{\mathbf{Z}}}$ and $\frac{d^2h}{dN_{\mathbf{Z}}^2}$ have simple expressions so that the Newton-Raphson algorithm can give a very accurate value for $\bar{\mathrm{I}}_{app}(\mathbf{X})$ very quickly in only few iterations.

3.5 Implementation

To find a solution to Problem 1, i.e. to compute the top-K SMI dependencies, we implemented in C++ a branch-and-bound algorithm based on $\bar{\mathrm{I}}_{app}(\mathbf{X})$. Although the branch-and-bound scheme is relatively standard, we provide some additional details on how we significantly optimized our implementation. First the crucial computationally intensive step is the extraction of counts $n_{\mathbf{x},y}$ from data samples, as we need these for every current pattern $\mathbf{X}$ score $\tilde{\mathrm{I}}^{(\alpha)}(\mathbf{X};Y)$ and bound $\bar{\mathrm{I}}_{app}(\mathbf{X})$. To do so efficiently we use the HFP-Growth algorithm [14] which relies on FP-trees [8] to compute the counts of contingency tables. For fair comparison we implement both our own scores, as well as the competitors we will compare to in the experiments based on these data structures. Second, expressions of score $\tilde{\mathrm{I}}^{(\alpha)}(\mathbf{X};Y)$ and bound $\bar{\mathrm{I}}_{app}(\mathbf{X})$ can be written as functions of sums $\sum_{\mathbf{x},y} \operatorname{xlogx}(n_{\mathbf{x},y}+\alpha)$ and $\sum_{\mathbf{x}} \operatorname{xlogx}(n_{\mathbf{x}}+N_Y\,\alpha)$, which are the only terms depending on counts $n_{\mathbf{x}}$ and $n_{\mathbf{x},y}$. These sums can directly be updated within the loops computing the intersection of partitions $\mathcal{P}(\mathbf{X}\cup\{Y\}) = \mathcal{P}(\mathbf{X}) \cap \mathcal{P}(Y)$ so that it almost costs no additional memory and time. Then iterations of Newton-Raphson to compute $\bar{\mathrm{I}}_{app}(\mathbf{X})$ only depend on previously computed

constants, without requiring to access the FP-tree. The result is thus immediate.

3.6 Problem complexity

In Sec. 3.3 we already gave an indication why Problem 3 is NP-hard: the only place in the derivation of bound $\bar{\mathrm{I}}_{app}(\mathbf{X})$ where optimality is broken is when applying the min inequality of Eq. 10 at Eq. 11. One can therefore wonder when this inequality becomes an equality, that is, when the bound $\underline{\tilde{H}}^{(\alpha,\mathbf{x})}_{app}(N)$ becomes equal to optimal bound $\underline{\tilde{H}}^{(\alpha,\mathbf{x})}_{opt}(N)$. We answer this question with the following theorem.

Theorem 3.1. *Bound $\underline{\tilde{H}}^{(\alpha,\mathbf{x})}_{app}(N)$ is optimal, equal to $\underline{\tilde{H}}^{(\alpha,\mathbf{x})}_{opt}(N)$, if and only if there exists a N-partition $(\mathbf{P}_z)_{1\leq z\leq N}$ of counts $(n_{\mathbf{x},y})$ such that the N sums $\sum_{n_{\mathbf{x},y}\in\mathbf{P}_z} n_{\mathbf{x},y}$ are all equal.*

We postpone the proof to Appendix 6.2.3. We can now use this theorem to give a polynomial time reduction of the NP-complete *partition problem* to Problem 3.

Theorem 3.2. *Problem 3 is NP-hard.*

We postpone the proof to Appendix 6.2.4.

4 EXPERIMENTS

In this section we evaluate SMI and the search algorithm that we proposed above. The first question we aim to answer is whether smoothed mutual information identifies statistically reliable dependencies. The second question we consider is whether it is feasible to use SMI to discover top-K dependencies from real world data.

4.1 Dependency Discovery

To allow evaluation with known ground truth where we have control over the difficulty over the task, we consider synthetic data in this experiment. In particular, we consider a simple Bayesian network, where features of $\mathcal{X}$ are all independent and where target variable Y depends only on a subset of parent features. We will use this data to evaluate whether the different methods are able to recover all parents of Y without including any unrelated variables X, for different numbers of parent variables and samples.

To this end, we generate synthetic data in the following way. We consider a generative model whose descriptive features are made of 10 independent random variables X_k and a target variable Y. All variables have a domain size of 4. Only $X_1,\ldots,X_P$ are actual parents of the target variable Y, all others features being independent with Y. In the following, we consider values 1, 3 and 5 for P to cover the cases of low, medium and high dimensional dependencies relatively to the total number of features. We can straightforwardly sample a model with P dependent variables X by uniformly sampling from the joint probability distribution $P(Y,\mathcal{X}) = P(Y|X_1,\ldots,X_P)\times\prod_{k=1}^{10} P(X_k)$. Out of all models, we only consider those whose fraction of information $F(\mathcal{X},Y) = \frac{\mathrm{I}(X;Y)}{\mathrm{H}(Y)}$ is within the interval $[0.2, 0.3]$, as to ensure there does exist a true dependency between Y and its parents, without it being so strong that discovery becomes trivial. For every value of P we consider different sample sizes n, distributed on a log scale from 10 to 10 000. For each value, we sample 20 models of type P, which results in 20 joint distributions denoted $p^{(i)}_{i=1,\ldots,20}$, and for each such model, we then sample 100 datasets $\mathbf{D}^{(i)}_{n,j}$ of size n. That is, every point (P,n) we report is an average over 2 000 datasets.

As here we are interested in evaluating the quality of each estimator τ, we determine the top-1 maximizer $\mathbf{X}^*_{i,j,n,\tau}$ for each dataset $\mathbf{D}^{(i)}_{n,j}$. As our evaluation metric in this experiment we use the **normalized intersection** between the ground truth and the discovered set. Formally, it is defined as

$$\mathcal{I}^{(P,n)}(\tau) = \mathbb{E}\left(\frac{|\mathbf{X}^*_i \cap \mathbf{X}^*_{i,j,n,\tau}|}{\max\{|\mathbf{X}^*_i|, |\mathbf{X}^*_{i,j,n,\tau}|\}}\right) ,$$

where $\mathbf{X}^*_i$ represents the true maximizer of model $p^{(i)}$ and the expected value is with respect to $i\in[1,20]$ and $j\in[1,100]$. We also consider **precision** $\mathcal{P}$ and **recall** $\mathcal{R}$, which are defined as

$$\mathcal{P}^{(P,n)}(\tau) = \mathbb{E}\left(\frac{|\mathbf{X}^*_i \cap \mathbf{X}^*_{i,j,n,\tau}|}{|\mathbf{X}^*_{i,j,n,\tau}|}\right)$$

$$\text{and}\quad \mathcal{R}^{(P,n)}(\tau) = \mathbb{E}\left(\frac{|\mathbf{X}^*_i \cap \mathbf{X}^*_{i,j,n,\tau}|}{|\mathbf{X}^*_i|}\right) .$$

Low values of precision (resp. recall) expresses the tendency of an estimator to "overfit", i.e. to include variables independent with Y (resp. to "underfit", i.e. to miss parent variables of Y).

In addition to our SMI estimator $\tilde{\mathrm{I}}^{(\alpha)}$, we consider three competitors. The one proposed by Vinh et al. [12], which we denote as $\hat{I}_{\chi,\alpha}$, estimates mutual information by substracting from the plugin estimator a corrective term based on a χ^2 statistical test

$$\hat{I}_{\chi,\alpha}(\mathbf{X};Y) = \hat{I}(\mathbf{X};Y) - \frac{1}{2n}\chi_{\alpha,l(\mathbf{X},Y)} , \tag{16}$$

where $\chi_{\alpha,l(\mathbf{X},Y)}$ is the critical value corresponding to a significance level $1-\alpha$ and degrees of freedom $l(\mathbf{X},Y) = (\prod_{X\in\mathbf{X}} N_X - 1)(N_Y - 1)$ in χ^2 test. Here, α can be thought as a parameter regulating the amount of penalty. The second, proposed by Mandros et al. [10] is the *reliable fraction of information* (RFI). It estimates mutual information as

$$\hat{I}_0(\mathbf{X};Y) = \hat{I}(\mathbf{X};Y) - \hat{m}_0(\mathbf{X},Y,n) ,$$

where $\hat{m}_0(\mathbf{X},Y,n)$ is the expected value of $\hat{I}(\mathbf{X},Y)$ with respect to the uniform distribution over all n-permutations of the n Y-value samples (see Mandros et al. [10] for the exact definition). Finally we also evaluate to which extent the score of Suzuki [19] could address our problem, while it has been introduced in the slightly different context of Bayesian network structure extraction. This estimator includes a correction based on the Minimum Description Length (MDL) principle. It is defined as

$$\hat{I}_S(\mathbf{X};Y) = \hat{I}(\mathbf{X};Y) - \frac{(N_{\mathbf{X}}-1)(N_Y-1)}{2n}\log(n) . \tag{17}$$

We consider three different pseudocounts α, i.e. 0.5, 1, and 2 to parameterize $\tilde{\mathrm{I}}^{(\alpha)}$, and use probability level $p = 0.95$ for $\hat{I}_{\chi,p}$.[2] Overall, we hence consider six estimators: three variants of SMI, namely $\tilde{\mathrm{I}}^{(0.5)}$, $\tilde{\mathrm{I}}^{(1)}$, and $\tilde{\mathrm{I}}^{(2)}$, and three competitors, $\hat{I}_{\chi,95}$, $\hat{I}_0$, and $\hat{I}_S$.

[2] In addition we considered $p = 0.99$, but as we found this provides results highly similar to those for $p = 0.95$ we omit these for succinctness.

We present the results for small, medium, and high dimensional dependencies as Fig. 3, Fig. 4 resp. Fig. 5. From the figures we see that $\hat{I}_S$ is clearly not adapted to our problem as it serously underfits for medium and large dependencies. The remaining scores can then be divided in two groups: on one side, $\hat{I}_0$ and $\hat{I}_{\chi,95}$ perform well for high dimensional dependencies (i.e $P = 5$ on Fig. 5), of which $\hat{I}_0$ outperforms $\hat{I}_{\chi,95}$. However, as the precision curve of Fig. 3 shows, both strongly "overfit" for smaller dependencies. Paradoxically, both scores perform worse for higher sample sizes—which is rather counter-intuitive as a larger number of samples should lead to a better estimation. The reason we find for this is that the regularization effect of these estimators disappears too early as n grows, so that their respective maximizer picks up more and more independent variables just by chance. This effect is only observable on small or medium sized models as for large models, the maximizer is close to the whole set of variables. We note that in practice, this is unlikely to be the case—if all our data would be dependent, chances are high we would already have discovered this by hand.

In the more interesting and practical domain of discovering lower order dependencies from large sets of variables, SMI performs very well. While the regularization effect of Laplace smoothing makes SMI a conservative estimator that requires a larger number of samples to learn large models, it is the only estimator whose performance continuously grows with n for all settings, but for overly small values of α, like $\alpha = 0.5$.

Overall, SMI is the only estimator that performs robustly in every setting of this experiment; on lower order dependencies it wins by a clear margin and is the only that behaves like it should, on medium dependencies it performs on par with the state of the art, while in the experiment considering higher order dependencies—which inherently favours scores that tend to overfit—it achieves excellent precision and recall better than those of $\hat{I}_{\chi,95}$.

4.2 Computational Efficiency

Next, we study the computational efficiency of SMI and its competitors on real data. To make sure that we fairly compare scores and bound efficiencies rather than the quality of their implementation, we implemented all scores in the same optimized branch-and-bound framework described in Sec. 3.5. To obtain upper bounds of scores $\hat{I}_{\chi,\alpha}$ and $\hat{I}_S$, we replace term $\hat{I}(\mathbf{X};Y)$ with $\mathrm{H}(Y)$ in Eq. 16 and Eq. 17. For $\hat{I}_0$, we use the improved bound proposed by Mandros et al. [11]. To allow for easy comparison, we evaluate on the same datasets used and further described by Mandros et al. [10], reporting on all those that take more than 10 seconds for at least one of the scores. We give the wall-clock run times in Tab. 3. We make the code and all data available for reproducibility.[3]

We see that SMI is up to two orders of magnitude faster than the slowest competitor, RFI. The higher we set alpha, the faster SMI is, which is explained by the fact that it will need increasingly strong evidence to consider higher cardinality sets in the search lattice. As we saw in the previous experiment, $\hat{I}_S$ penalizes so strongly that it effectively always disregards most of the search space.

[3] https://github.com/P-Fred/Smoothie

Dataset	$\hat{I}_0$	$\hat{I}_{\chi,95}$	$\hat{I}_S$	$\tilde{\mathrm{I}}^{(0.5)}$	$\tilde{\mathrm{I}}^{(1)}$	$\tilde{\mathrm{I}}^{(2)}$	$\tilde{\mathrm{I}}^{(5)}$
lymphography	16	6	1	2	1	1	1
german	19	13	1	4	3	2	2
segment	28	34	1	2	1	1	1
sonar	123	53	2	28	15	5	2
penbased	129	66	1	3	3	3	1
wdbc	143	176	2	121	47	17	3
vehicle	154	24	1	3	2	1	1
twonorm	273	372	4	90	25	24	5
satimage	928	68	3	50	11	6	1
ionosphere	939	154	1	66	28	13	3
ring	1028	99	4	68	37	20	9
spectfheart	1178	249	3	180	116	66	15
texture	3423	173	1	13	7	6	1
splice	> 1h	> 1h	142	> 1h	> 1h	2238	499
optdigits	> 1h	> 1h	229	> 1h	> 1h	> 1h	109
average	**372.5**	**50.0**	**0.9**	**16.6**	**8.4**	**5.6**	**1.0**

Table 3: Processing times (in sec.) to mine top-100 dependencies for different scores and datasets. Last row gives for each estimator, the average processing time ratio relatively to $\tilde{\mathrm{I}}^{(5)}$.

5 CONCLUSION

We considered uniform priors to counter the inflated estimates during the high dimensional optimization of mutual information for reliable functional dependency discovery. For the resulting smoothed mutual information estimator (SMI), we provided both theoretical justifications and showed important properties. In addition, we derived a tight and admissible bounding function for SMI to be used for pruning, leading to an effective branch-and-bound algorithm for the hard problem of finding the top-k dependencies. Our evaluation showed that the resulting method can very efficiently discover sparse and informative dependencies, allowing for data analysis on bigger data than previous proposals.

Our approach to reliable dependency discovery via smoothing can be further explored with different smoothing techniques, e.g., Good-Turing. In addition, a tighter bounding function can be derived for more effective search, while a greedy algorithm with performance guarantees would allow access to even bigger data.

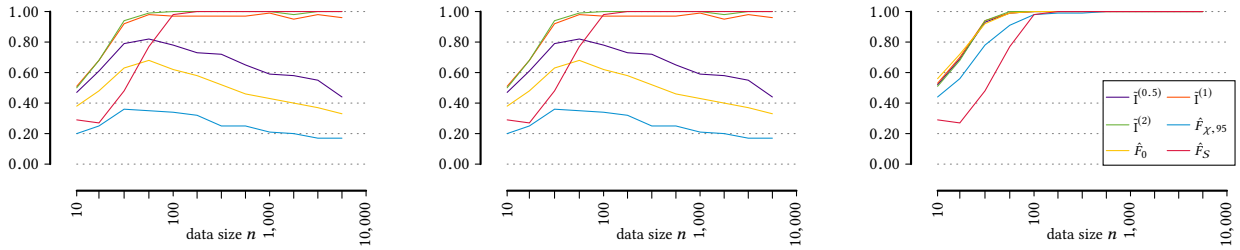


Figure 3: Normalized intersection (left), precision (middle) and recall (right) for low dimensional dependencies ($P = 1$).

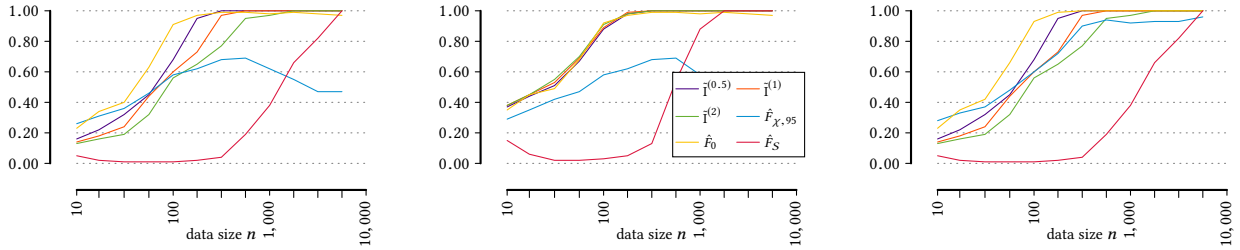


Figure 4: Normalized intersection (left), precision (middle) and recall (right) for medium dimensional dependencies ($P = 3$).

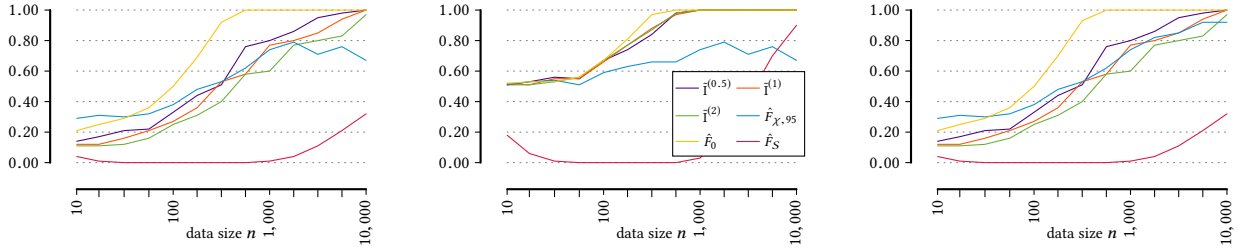


Figure 5: Normalized intersection (left), precision (middle) and recall (right) for high dimensional dependencies ($P = 5$).

REFERENCES

[1] András Antos and Ioannis Kontoyiannis. 2001. Convergence properties of functional estimates for discrete distributions. *Random Struct. Algor.* 19, 3-4 (2001), 163–193.

[2] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. 2012. Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection. *J. Mach. Learn. Res.* 13 (2012), 27–66.

[3] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*. Wiley.

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

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

[6] Bryan R Goldsmith, Mario Boley, Jilles Vreeken, Matthias Scheffler, and Luca M Ghiringhelli. 2017. Uncovering structure-property relationships of materials by subgroup discovery. *New J. Phys.* 19, 1 (2017), 013031.

[7] Isabelle Guyon and André Elisseeff. 2003. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.* 3 (2003), 1157–1182.

[8] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. 2004. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Min. Knowl. Discov.* 8, 1 (2004), 53–87.

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

[10] Panagiotis Mandros, Mario Boley, and Jilles Vreeken. 2017. Discovering Reliable Approximate Functional Dependencies. In *KDD*. ACM, 355–363.

[11] Panagiotis Mandros, Mario Boley, and Jilles Vreeken. 2018. Discovering Reliable Dependencies from Data: Hardness and Improved Algorithms. In *ICDM*. IEEE Comuter Society, 317–326.

[12] Xuan Vinh Nguyen, Jeffrey Chan, and James Bailey. 2014. Reconsidering Mutual Information Based Feature Selection: A Statistical Significance View. In *AAAI*. AAAI Press, 2092–2098.

[13] Thorsten Papenbrock, Jens Ehrlich, Jannik Marten, Tommy Neubert, Jan-Peer Rudolph, Martin Schönberg, Jakob Zwiener, and Felix Naumann. 2015. Functional dependency discovery: An experimental evaluation of seven algorithms. *VLDB J.* 8, 10 (2015), 1082–1093.

[14] Frédéric Pennerath. 2018. An Efficient Algorithm for Computing Entropic Measures of Feature Subsets. In *ECML-PKDD (LNCS)*, Vol. 11052. Springer, 483–499.

[15] Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. 2016. A Framework to Adjust Dependency Measure Estimates for Chance. In *SDM*. SIAM, 423–431.

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

[17] Steffen Schober. 2013. Some worst-case bounds for Bayesian estimators of discrete distributions. In *ISIT*. IEEE, 2194–2198.

[18] Thomas SchÃijrmann and Peter Grassberger. 1996. Entropy estimation of symbol sequences. *Chaos* 6, 3 (1996), 414–427.

[19] Joe Suzuki. 1993. A construction of Bayesian networks from databases based on an MDL principle. In *UAI*. Morgan Kaufmann, 266–273.

[20] Joe Suzuki. 2019. Mutual Information Estimation: Independence Detection and Consistency. In *ISIT*. IEEE, 2514–2518.

[21] Ioannis Tsamardinos, Constantin Aliferis, Alexander Statnikov, and Er Statnikov. 2003. Algorithms for Large Scale Markov Blanket Discovery. In *FLAIRS*. AAAI Press, 376–380.

6 APPENDIX

6.1 Reproducibility

The C++ source code that implements the branch-and-bound scheme for SMI and all aforementioned scoring functions along with all datasets are downloadable from https://github.com/P-Fred/Smoothie. See the README file for compilation and usage instructions

6.1.1 Experiment on quality of top-1 dependency (Sec. 4.1). The experimental procedure is detailed below:

Given a set of estimators, a number $P \leq 10$ of parents and a set of numbers of samples, we do the following for each number n of samples:

(1) We generate 20 models as follows:
   (a) We consider 10 features X_1 to X_{10} and we choose features X_1 to X_P as the parent variables of Y.
   (b) Then for each tuple $\mathbf{x}_P = (x_1, \ldots, x_P)$ of values of $\mathbf{X}_P = \{X_1, \ldots, X_P\}$, we draw uniformly a sample distribution $P(Y|\mathbf{X}_P = \mathbf{x}_P)$ on the 3-simplex thanks to a symmetric Dirichlet distribution $\mathtt{Dir}(1)$ of order $N_Y = 4$.
   (c) We reject the model while mutual information $\mathrm{I}(\mathbf{X}; Y)$ is outside the interval $[0.2, 0.4]$.
   (d) Then for each feature X_k for $1 \leq k \leq 10$, we draw uniformly a sample distribution $P(X_k)$ on the 3-simplex thanks to a symmetric Dirichlet distribution $\mathtt{Dir}(1)$ of order $N_{X_K} = 4$.
   (e) Given each model, we sample 100 datasets of size n applying ancestral sampling on the joint distribution:

$$P(Y, \mathcal{X}) = P(Y|X_1, \ldots, X_P) \times \prod_{k=1}^{10} P(X_k)$$

(2) For each of the 20×100 generated datasets and for each estimator τ, we find the top-1 dependency $\mathbf{X}^*_{i,j,n,\tau} \rightarrow Y$ using our implementation and we compute its normalized intersection, precision and recall, taking $\mathbf{X}^*_i = \{X_1, \ldots, X_P\}$ as the true maximizer.
(3) We compute the average scores $\mathcal{I}^{(P,n)}(\tau)$, $\mathcal{P}^{(P,n)}(\tau)$ and $\mathcal{R}^{(P,n)}(\tau)$ for each estimator τ over all 20×100 datasets.

6.1.2 Experiment on computational efficiency (Sec. 4.2). All tests have been run on Intel i7-8700 CPU (3.20GHz) with 16GB RAM. Bounds used for the different estimators are given below:

- For the *smoothed mutual information* (SMI), the bound given in Sec. 3.4 of the present article.
- For the *reliable fraction of information* (RFI), we use the improved bound given in Mandros et al [11].
- For the *adjusted fraction of information* proposed by Vinh et al. [12], the bound is

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

- For the MDL score proposed by Suzuki [20], the bound is

$$\overline{\hat{I}_S}(\mathbf{X}; Y) = H(Y) - \frac{(N_{\mathbf{X}} - 1)(N_Y - 1)}{2n}\log(n) \ .$$

We limited the processing time of each run to one hour. We ignored datasets for which processing times for all estimators were all less than 10 seconds or all more than one hour.

6.2 Proofs

6.2.1 Proof of Property 3.

Proof. Suppose W has the minimal possible entropy $\tilde{\mathrm{H}}^{(\alpha)}(W)$ with at least two different observed values, say a and b, with a respective number of observations $n_a \geq 1$ and $n_b \geq 1$. Let W^* be the admissible variable that has the same observations as W but a is observed $n_a + n_b$ times whereas b is not observed. One next proves the contradiction $\tilde{\mathrm{H}}^{(\alpha)}(W^*) < \tilde{\mathrm{H}}^{(\alpha)}(W)$ that will conclude the proof. To this end, Δ is introduced as the difference $(n + N_W\alpha)\left(\tilde{\mathrm{H}}^{(\alpha)}(W) - \tilde{\mathrm{H}}^{(\alpha)}(W^*)\right)$ in order to show $\Delta > 0$:

$$\begin{aligned}\Delta = &\ \mathrm{xlogx}(\alpha) + \mathrm{xlogx}(n_a + n_b + \alpha) \\ &- \mathrm{xlogx}(n_a + \alpha) - \mathrm{xlogx}(n_b + \alpha) \\ = &\ f(n_a + \alpha) - f(\alpha) \ ,\end{aligned}$$

with

$$f : x > 0 \mapsto \mathrm{xlogx}(x + n_b) - \mathrm{xlogx}(x) \ .$$

But

$$\frac{df}{dx}(x) = \log\left(1 + \frac{n_b}{x}\right) > 0 \ .$$

Therefore f is a increasing function, and since $n_a > 0$, $f(\alpha) < f(n_a + \alpha)$, $\Delta > 0$. □

6.2.2 Derivation of global bound expression 15 in Sec. 3.4 .

Proof. According to Eq. 8 and given a local admissible bound $\underline{\tilde{H}}^{(\alpha,\mathbf{x})}_{app}$ we can obtain a global admissible bound as

$$\tilde{\mathrm{I}}_{app}(\mathbf{X}) = \max_{(N^{\mathbf{x}}_{\mathbf{Z}})} \left(h\left((N^{\mathbf{x}}_{\mathbf{Z}})\right)\right) \ \text{ with}$$

$$h\left((N^{\mathbf{x}}_{\mathbf{Z}})\right) = \tilde{\mathrm{H}}^{(N_{\mathbf{Z}}\,\alpha)}(Y) - \sum_{\mathbf{x}} \frac{n_{\mathbf{x}} + N^{\mathbf{x}}_{\mathbf{Z}} N_Y\,\alpha}{n + N_{\mathbf{Z}} N_Y\,\alpha} \underline{\tilde{H}}^{(\alpha,\mathbf{x})}_{app}(N^{\mathbf{x}}_{\mathbf{Z}}) \ .$$

where $N_{\mathbf{Z}} = \sum_{\mathbf{x}} N^{\mathbf{x}}_{\mathbf{Z}}$. Inserting expression of $\underline{\tilde{H}}^{(\alpha,\mathbf{x})}_{app}$ given by Eq. 14, we get

$$\begin{aligned}h\left((N^{\mathbf{x}}_{\mathbf{Z}})\right) = &\ \tilde{\mathrm{H}}^{(N_{\mathbf{Z}}\,\alpha)}(Y) - \sum_{\mathbf{x}} \frac{N^{\mathbf{x}}_{\mathbf{Z}}\ \mathrm{xlogx}\left(\frac{n_{\mathbf{x}}}{N^{\mathbf{x}}_{\mathbf{Z}}} + N_Y\,\alpha\right)}{n + N_{\mathbf{Z}} N_Y\,\alpha} \\ &+ \sum_{\mathbf{x}} \frac{\left(\sum_y \mathrm{xlogx}(n_{\mathbf{x},y} + \alpha)\right) + (N^{\mathbf{x}}_{\mathbf{Z}} - 1)\, N_Y\ \mathrm{xlogx}(\alpha)}{n + N_{\mathbf{Z}} N_Y\,\alpha} \\ = &\ \log\left(n + N_{\mathbf{Z}} N_Y\,\alpha\right) - \frac{\sum_y \mathrm{xlogx}(n_y + N_{\mathbf{Z}}\,\alpha)}{n + N_{\mathbf{Z}} N_Y\,\alpha} \\ &- \sum_{\mathbf{x}} \frac{N^{\mathbf{x}}_{\mathbf{Z}}\ \mathrm{xlogx}\left(\frac{n_{\mathbf{x}}}{N^{\mathbf{x}}_{\mathbf{Z}}} + N_Y\,\alpha\right)}{n + N_{\mathbf{Z}} N_Y\,\alpha} \\ &+ \frac{\left(\sum_{\mathbf{x},y} \mathrm{xlogx}(n_{\mathbf{x},y} + \alpha)\right) + \sum_{\mathbf{x}}(N^{\mathbf{x}}_{\mathbf{Z}} - 1)\, N_Y\ \mathrm{xlogx}(\alpha)}{n + N_{\mathbf{Z}} N_Y\,\alpha} \ .\end{aligned}$$

Permuting terms of the latter expression, we regroup unknowns $(N^{\mathbf{x}}_{\mathbf{Z}})$ into what can be interpreted as the entropy of a virtual variable $\mathbf{Z}'$. This variable is such that $\forall \mathbf{x}, N^{\mathbf{x}}_{\mathbf{Z}'} = N^{\mathbf{x}}_{\mathbf{Z}}$ and such that for each

of its values $\mathbf{z}'$, its count $n_{\mathbf{z}'}$ is $\frac{n_{\mathbf{x}}}{N_{\mathbf{Z}}^{\mathbf{x}}}$, where $\mathbf{x}$ is the restriction of $\mathbf{z}'$ on $\mathbf{X}$ (i.e. $\mathbf{z}' \in \mathbb{D}_{\mathbf{Z}/\mathbf{x}}$). This gives

$$
\begin{aligned}
h\left((N_{\mathbf{Z}}^{\mathbf{x}})\right) &= \log\left(n + N_{\mathbf{Z}}\, N_Y\, \alpha\right) - \sum_{\mathbf{x}} \sum_{\mathbf{z} \in \mathbb{D}_{\mathbf{Z}/\mathbf{x}}} \frac{\operatorname{xlogx}\left(\frac{n_{\mathbf{x}}}{N_{\mathbf{Z}}^{\mathbf{x}}} + N_Y\, \alpha\right)}{n + N_{\mathbf{Z}}\, N_Y\, \alpha} \\
&\quad - \frac{\sum_y \operatorname{xlogx}(n_y + N_{\mathbf{Z}}\, \alpha)}{n + N_{\mathbf{Z}}\, N_Y\, \alpha} \\
&\quad + \frac{\left(\sum_{\mathbf{x},y} \operatorname{xlogx}(n_{\mathbf{x},y} + \alpha)\right) + (N_{\mathbf{Z}} - N_{\mathbf{X}})\, N_Y\ \operatorname{xlogx}(\alpha)}{n + N_{\mathbf{Z}}\, N_Y\, \alpha} \\
&= \tilde{\mathrm{H}}^{(N_Y\, \alpha)}(\mathbf{Z}') \\
&\quad + \frac{\left(-\sum_y \operatorname{xlogx}(n_y + N_{\mathbf{Z}}\, \alpha)\right) + N_{\mathbf{Z}}\, N_Y\ \operatorname{xlogx}(\alpha) + c_0}{n + N_{\mathbf{Z}}\, N_Y\, \alpha} \quad ,
\end{aligned}
$$

grouping terms independent of $\mathbf{Z}$ into a constant

$$
c_0 = \left(\sum_{\mathbf{x},y} \operatorname{xlogx}(n_{\mathbf{x},y} + \alpha)\right) - N_{\mathbf{X}}\, N_Y\ \operatorname{xlogx}(\alpha) \quad .
$$

Using the fact $\tilde{\mathrm{H}}^{(N_Y\, \alpha)}(\mathbf{Z}') \leq \log(N_{\mathbf{Z}})$, we get Expr. 15 provided in Sec. 3.4. We note that a tighter bound can be obtained by finding the maximal value of $\tilde{\mathrm{H}}^{(N_Y\, \alpha)}(\mathbf{Z}')$ reachable under the constraints induced by counts $n_{\mathbf{x}}$. However the gain is so small in practice that we choose the slightly coarser bound $\log(N_{\mathbf{Z}})$ instead. □

6.2.3 Proof of Theorem 3.1. Before proving Theorem 3.1, one recalls the following property:

Property 4. *Given two real functions f and g defined on the same compact set S, let be $\mathbf{w}_{opt} \in \arg\min_{\mathbf{w} \in S}(f(\mathbf{w}) - g(\mathbf{w}))$, $\underline{f} = \min_{\mathbf{w} \in S}(f(\mathbf{w}))$ and $\overline{g} = \max_{\mathbf{w} \in S}(g(\mathbf{w}))$. Then*

$$
\min_{\mathbf{w} \in S}(f(\mathbf{w}) - g(\mathbf{w})) = \underline{f} - \overline{g} \Leftrightarrow f(\mathbf{w}_{opt}) = \underline{f} \text{ and } g(\mathbf{w}_{opt}) = \overline{g} \quad .
$$

Proof. (1) $\Leftarrow$ (2) is trivial. Then if $\min_{\mathbf{w} \in S}(f(\mathbf{w}) - g(\mathbf{w})) = \underline{f} - \overline{g}$, necessarily $f(\mathbf{w}_{opt}) - \underline{f} = g(\mathbf{w}_{opt}) - \overline{g}$. The left side of this equation is nonnegative while the right side is nonpositive. Therefore both sides are equal to zero, proving (1) $\Rightarrow$ (2). □

This property is used in the next proof of Theorem 3.1:

Proof. We first consider (1) $\Rightarrow$ (2). We reuse notations of Prop. 4 by taking $\mathbf{w} = (n_{\mathbf{z},y})$, S as the finite (thus compact) intersection of $\mathbb{N}^N$ with the N_Y plane constraints $\forall y,\ \sum n_{\mathbf{z},y} = n_{\mathbf{x},y}$, functions $f(\mathbf{w}) = \tilde{\mathrm{H}}^{(\alpha)}(\mathbf{Z}|Y, \mathbf{X} = \mathbf{x})$ and $g(\mathbf{w}) = \tilde{\mathrm{H}}^{(N_Y\, \alpha)}(\mathbf{Z}|\mathbf{X} = \mathbf{x})$ and $\mathbf{w}_{opt} \in \arg\min_{\mathbf{w} \in S}(f(\mathbf{w}) - g(\mathbf{w}))$ some optimal set of counts. Then according to equations 9 and 11,

$$
\begin{aligned}
\underline{\tilde{H}}_{opt}^{(\alpha,\mathbf{x})}(N) &= \tilde{\mathrm{H}}^{(N\, \alpha)}(Y|\mathbf{X} = \mathbf{x}) + \min_{\mathbf{w} \in S}(f(\mathbf{w}) - g(\mathbf{w})) \\
\underline{\tilde{H}}_{app}^{(\alpha,\mathbf{x})}(N) &= \tilde{\mathrm{H}}^{(N\, \alpha)}(Y|\mathbf{X} = \mathbf{x}) + \underline{f} - \overline{g} \quad .
\end{aligned}
$$

Hypothesis $\underline{\tilde{H}}_{app}^{(\alpha,\mathbf{x})}(N) = \underline{\tilde{H}}_{opt}^{(\alpha,\mathbf{x})}(N)$ thus implies $\min_{\mathbf{w} \in S}(f(\mathbf{w}) - g(\mathbf{w})) = \underline{f} - \overline{g}$. Property 4 thus applies so that $f(\mathbf{w}_{opt}) = \underline{f}$ and $g(\mathbf{w}_{opt}) = \overline{g}$.

Statement $f(\mathbf{w}_{opt}) = \underline{f}$ means for all y, $\tilde{\mathrm{H}}^{(\alpha)}(\mathbf{Z}|Y = y, \mathbf{X} = \mathbf{x}) = \tilde{h}_N^{(\alpha)}(n_{\mathbf{x},y})$. This implies according to Prop. 3 that the optimal contingency table of $\mathbf{w}_{opt}$ has only one non-empty cell with count $n_{\mathbf{x},y}$ on each row mapped to value y, as illustrated by tables on Fig. 2. Thus, counts $n_{\mathbf{x},y}$ are never spread on multiple cells.

Statement $g(\mathbf{w}_{opt}) = \overline{g}$ means $\tilde{\mathrm{H}}^{(N_Y\, \alpha)}(\mathbf{Z}|\mathbf{X} = \mathbf{x}) = \log(N)$. This is only possible if $\widehat{P}(\mathbf{Z}|\mathbf{X} = \mathbf{x})$ is uniform so that

$$
\forall \mathbf{z}, n_{\mathbf{z}} = \sum_y n_{\mathbf{z},y} = \frac{n_{\mathbf{x}}}{N} \quad .
$$

But counts $n_{\mathbf{z},y}$ are either equal to some count $n_{\mathbf{x},y}$ or 0. Thus the N columns of the contingency table induce a partition of $(n_{\mathbf{x},y})$ whose sums are all equal, as illustrated by table of Fig. 1c.

Conversely, for (2) $\Rightarrow$ (1), if counts $(n_{\mathbf{x},y})$ can be partitioned in N sets $(\mathbf{P}_z)_{1 \leq z \leq N}$ of equal sums, these counts $(n_{\mathbf{x},y})$ can be arranged in a $N_Y \times N$ contingency table so that if count $n_{\mathbf{x},y}$ is part of P_z then it is put in cell of coordinate (y, z). This way, every row has one and only one non-empty cell and sums of counts on every column are all equal. These are sufficient conditions for the entropy $\tilde{\mathrm{H}}^{(\alpha)}(Y|\mathbf{Z}, \mathbf{X} = \mathbf{x})$ to be equal to $\underline{\tilde{H}}_{app}^{(\alpha,\mathbf{x})}(N)$, proving $\underline{\tilde{H}}_{opt}^{(\alpha,\mathbf{x})}(N) \geq \underline{\tilde{H}}_{app}^{(\alpha,\mathbf{x})}(N)$ and thus the equality. □

6.2.4 Proof of Theorem 3.2.

Proof. The NP-complete partition problem is defined as follows: Given a set $(x_i)_{1 \leq i \leq m}$ of positive integers and a number k, decide whether there exists a k-partition $(\mathbf{P}_j)_{1 \leq j \leq k}$ of indexes $\{1..m\}$ such that all sums $S_j = \sum_{i \in \mathbf{P}_j} x_i$ for $j \in \{1..k\}$ are equal. Such problem instance ca be mapped to an instance $\mathcal{I}$ of Problem 3: take $\alpha = 1$, $N_Y = m$, and $\forall y \in \{1..m\}$, $n_{\mathbf{x},y} = x_y$. In addition, Eq. 14 allows to compute $\underline{\tilde{H}}_{app}^{(\alpha,\mathbf{x})}(N)$ in polynomial time. Then after solving $\mathcal{I}$ and according to Theorem 3.1, there exists a partition of equal sums if and only if the returned solution $\underline{\tilde{H}}_{opt}^{(\alpha,\mathbf{x})}(N)$ is equal to $\underline{\tilde{H}}_{app}^{(\alpha,\mathbf{x})}(N)$. The certificate is provided by the contingency table $(n_{\mathbf{z},y})$. The NP-complete *partition problem* is thus polynomial-time reducible to Problem 3. □