Universal Dependency Analysis

Hoang-Vu Nguyen° Panagiotis Mandros° Jilles Vreeken°

Abstract

Most data is multi-dimensional. Discovering whether any subset of dimensions, or *subspaces*, shows dependence is a core task in data mining. To do so, we require a measure that quantifies how dependent a subspace is. For practical use, such a measure should be *universal* in the sense that it captures correlation in subspaces of any dimensionality and allows to meaningfully compare scores across different subspaces, regardless how many dimensions they have and what specific statistical properties their dimensions possess. Further, it would be nice if the measure can non-parametrically and efficiently capture both linear and non-linear correlations.

In this paper, we propose UDS, a multivariate dependence measure that fulfils all of these desiderata. In short, we define UDS based on cumulative entropy and propose a principled normalisation scheme to bring its scores across different subspaces to the same domain, enabling *universal* dependence assessment. UDS is purely non-parametric as we make no assumption on data distributions nor types of correlation. To compute it on empirical data, we introduce an efficient and non-parametric method. Extensive experiments show that UDS outperforms state of the art.

1 Introduction

Dependency and correlation analysis are key elements of data mining. It has applications in many domains, including biology and neuroscience [13, 24]. Traditionally, it focuses on two dimensions. More often than not, however, data is multi-dimensional and can contain multivariate dependencies e.g. hidden in subsets of dimensions, or *subspaces* [3]. Identifying such subspaces is an important step towards understanding the data. To do so, we need a *universal* measure.

First of all, the measure should be universal in the sense that it is able to detect correlations in subspaces of any dimensionality. Second, for both usability by the domain experts as well as efficient search, it should allow for universal comparison of scores – regardless the number of dimensions they were computed over, or the statistical properties of these dimensions. Third, for exploratory analysis the measure should be able to non-parametrically capture both linear and non-linear correlations, making no assumption on data distributions nor types of correlation the data may contain. Fourth, the measure should permit non-parametric and efficient computation on empirical data.

Of the above desiderata, although perhaps most important for practical use, universality is most often overlooked in the literature. For instance, commonly used measures in subspace search such as total correlation [4], CMI [16], and quadratic measure of dependency QR [15] in general produce scores that are not comparable across subspaces of different dimensionality. Pearson's correlation, HICS [11], and MAC [17], while addressing universality, have other issues. In particular, Pearson's correlation is for pairwise linear correlations. HICS relies on high dimensional conditional distributions and hence is prone to the curse of dimensionality. For each subspace, MAC needs to compute correlation scores of all dimension pairs before outputting the final score – a potential source of inefficiency.

In this paper, we aim at addressing all desiderata. We do so by proposing UDS, for universal dependency score. In short, we define UDS based on cumulative entropy [6, 20] – a new type of entropy permitting non-parametric computation on empirical data. To address universality, we propose a *principled* normalisation scheme to bring scores of UDS across different subspaces to the same domain, enabling universal dependency assessment. Further, UDS is highly suited to non-parametric exploratory analysis as we make no assumption on data distributions nor types of correlation. Lastly, we propose a non-parametric method to reliably and efficiently compute UDS on empirical data. Our method does not require to compute pairwise correlations of the involved dimensions and scales near linearly to the data size. Extensive experiments on both synthetic and real-world data sets show that UDS has high statistical power and performs very well in subspace search.

The road map is as follows. First, we present the principles of our measure. In Section 3 we review cumulative entropy, and introduce UDS in Sec. 4, followed by its computation in Sec. 5. We review related work in in Sec. 6. We empirically evaluate in Sec. 7, and round up with conclusions in Sec. 8. For readability and succinctness, we postpone the proofs for the theorems to the Appendix.

2 Correlation Measures – A Brief Primer

We consider a multivariate data set $\mathbf{D}$ with m records and n real-valued dimensions $X_1, \ldots, X_n$. For each dimension X_i, we assume that $dom(X_i) = [min_i, max_i]$. Further, we write $p(X_i)$ as its probability distribution function (pdf) and $P(X_i)$ as its cumulative distribution function (cdf).

Each non-empty subset $S \subset \{X_1, \ldots, X_n\}$ constitutes a subspace. To discover correlated subspaces, we need to

°Max Planck Institute for Informatics and Saarland University, Saarbrücken, Germany.
{hnguyen,pmandros,jilles}@mpi-inf.mpg.de

quantify correlation score $corr(S)$ of S. We will mainly use subspace $\{X_1, \ldots, X_d\}$ where $d \in [1, n]$ in our analysis. We write $X_{1,\ldots,i}$ for shorthand of $X_1, \ldots, X_i$ ($i \geq 1$).

In principle, $corr(X_{1,\ldots,d})$ quantifies to how much the relation of $X_{1,\ldots,d}$ deviates from the statistical independence condition, i.e. how much their joint probability distribution differs from the product of their marginal probability distributions [10, 26]. The larger the difference, the higher $corr(X_{1,\ldots,d})$ is. Formally, we have

$$corr(X_{1,\ldots,d}) = \mathit{diff}\left(p(X_{1,\ldots,d}), \prod_{i=1}^{d} p(X_i)\right) \tag{2.1}$$

with *diff* being an instantiation of a divergence measure. An important property for data analysis is that [23] $corr(X_{1,\ldots,d})$ is non-negative and zero iff $X_{1,\ldots,d}$ are statistically independent, i.e. $p(X_{1,\ldots,d}) = \prod_{i=1}^{d} p(X_i)$.

As Eq. (2.1) works with multivariate distribution $p(X_{1,\ldots,d})$, when d is large $corr(X_{1,\ldots,d})$ is prone to the curse of dimensionality. Recognising this issue, [7, 16, 29] factorise $p(X_{1,\ldots,d})$ and define

$$corr(X_{1,\ldots,d}) = \sum_{i=2}^{d} \mathit{diff}\left(p(X_i), p(X_i \mid X_{1,\ldots,i-1})\right) \quad . \tag{2.2}$$

To uphold the non-negativity and zero score requirements, it suffices that $\mathit{diff}(p(X_i), p(X_i \mid \cdot))$ must be non-negative and is zero iff $p(X_i) = p(X_i \mid \cdot)$ [16].

As we can see, Eq. (2.2) is a factorised form of Equation (2.1). In particular, it computes $corr(X_{1,\ldots,d})$ by summing up the difference between the marginal distribution $p(X_i)$ and the conditional distribution $p(X_i \mid X_{1,\ldots,i-1})$ for $i \in [2, d]$. In this way, loosely speaking $corr(S)$ is the sum of the correlation scores of subspaces

$$(X_1, X_2), \ldots, (X_1, \ldots, X_i), \ldots, (X_1, \ldots, X_d)$$

if we consider $\mathit{diff}(p(X_i), p(X_i \mid X_{1,\ldots,i-1}))$ to be the correlation score of the subspace $(X_{1,\ldots,i})$. The advantage of using lower dimensional subspaces is that $corr(X_{1,\ldots,d})$ in Eq. (2.2) is more robust to high dimensionality. It, however, in general is variant to the way we form the factorisation, i.e. the permutation of dimensions used.

We eliminate such dependence by taking the maximum score over all permutations. By considering the maximum value, we aim at uncovering the best correlation score of the dimensions involved, which is in line with maximal correlation analysis [2, 21]. Formally, letting $\mathcal{F}_d$ be the set of bijective functions $\sigma : \{1, \ldots, d\} \to \{1, \ldots, d\}$, we have

$$corr(X_{1,\ldots,d}) = \max_{\sigma \in \mathcal{F}_d} \sum_{i=2}^{d} \mathit{diff}\left(p(X_i^\sigma), p(X_i^\sigma \mid X_{1,\ldots,i-1}^\sigma)\right) \tag{2.3}$$

where $X_i^\sigma = X_{\sigma(i)}$ for $i \in [1, d]$. Like Eq. (2.2), $corr(X_{1,\ldots,d})$ in Eq. (2.3) also is more robust to high dimensionality. Further, it is permutation invariant. We design UDS based on this factorised form.

To this end, one important issue however remains open, which is: How to quantify $\mathit{diff}(p(X_i), p(X_i \mid \cdot))$ to fulfill universality? We address this by means of cumulative entropy [6, 20] which we introduce next.

3 Cumulative Entropy

In this section, we first provide background of cumulative entropy (CE). Then, we review Cumulative Mutual Information (CMI) [16] – a correlation measure that is defined based on CE but does not address universality.

3.1 Background of Cumulative Entropy In principle, CE captures the information content (i.e. complexity) of a probability distribution. However, different from Shannon entropy, it works with cdfs and can be regarded as a substitute for Shannon entropy on real-valued data. Formally, the CE of a real-valued univariate random variable X is given as

$$h(X) = -\int P(x) \log P(x) dx \quad .$$

The conditional CE of a real-valued univariate random variable X given $Z \in \mathbb{R}^d$ is defined as

$$h(X \mid Z) = \int h(X \mid z) p(z) dz \quad .$$

The conditional CE has two important properties given by the following theorem [6, 20].

THEOREM 3.1. *$h(X \mid Z) \geq 0$ with equality iff X is a function of Z. $h(X \mid Z) \leq h(X)$ with equality iff X is statistically independent of Z.*

Besides, unconditional CE can be computed in closed-form on empirical data. Let $x_1 \leq \ldots \leq x_m$ be the ordered records of X. We have $h(X) = -\sum_{i=1}^{m-1} (x_{i+1} - x_i) \frac{i}{m} \log \frac{i}{m}$. Having introduced CE, next we review CMI.

3.2 Cumulative Mutual Information CMI follows the factorised model of correlation measures in Eq. (2.3). It instantiates $\mathit{diff}(p(X_i), p(X_i \mid \cdot))$ by $h(X_i) - h(X_i \mid \cdot)$. Following Theorem 3.1, this instantiation is non-negative and zero iff $p(X_i) = p(X_i \mid \cdot)$, which is desirable (cf. Section 2). Formally, we have:

DEFINITION 3.1. ***Cumulative Mutual Information* (CMI)** *The* CMI *of $X_{1,\ldots,d}$ is*

$$\text{CMI}(X_{1,\ldots,d}) = \max_{\sigma \in \mathcal{F}_d} \sum_{i=2}^{d} h(X_{\sigma(i)}) - h(X_{\sigma(i)} \mid X_{\sigma(1),\ldots,\sigma(i-1)})$$

where $h(X_{\sigma(i)} \mid X_{\sigma(1)\ldots,\sigma(i-1)})$ is $h(X_{\sigma(i)} \mid Z)$ with $Z = X_{\sigma(1),\ldots,\sigma(i-1)}$ being a random vector whose domain is $dom(X_{\sigma(1)}) \times \cdots \times dom(X_{\sigma(i-1)})$.

Following Theorem 3.1, CMI satisfies the non-negativity and zero score requirements of correlation measure. Further, it is non-parametric and can capture different types of correlation. It however has some pitfalls that we explain next.

3.3 Drawbacks of CMI First, CMI does not address universality. In particular, it tends to give higher dimensional subspaces higher scores, as follows.

LEMMA 3.1. $\text{CMI}(X_{1,\ldots,d}) \leq \text{CMI}(X_{1,\ldots,d+1})$.

Proof. We postpone the proof to the online Appendix.

In addition, CMI scores of subspaces with the same dimensionality may also have different scales. To show this, we prove that the scale of $\text{CMI}(X_{1,\ldots,d})$ is dependent on $h(X_1), \ldots, h(X_d)$.

LEMMA 3.2. $\text{CMI}(X_{1,\ldots,d}) \leq \max_{\sigma \in \mathcal{F}_d} \sum_{i=2}^{d} h(X_{\sigma(i)})$.

Proof. We postpone the proof to the online Appendix.

That is, if two subspaces of the same dimensionality have no common dimension, the upper bound of their CMI scores may be different, rendering incomparable scales. Combining the results of Lemma 3.1 and 3.2, we can see that CMI does not address universality.

As the second issue, to compute CMI score of two or more dimensions CMI needs to search for the optimal permutation of the dimensions. The quality of this search is dependent on how well conditional CE terms are estimated. As CMI computes such terms using clustering, the search quality and hence the quality of CMI are dependent on how good the selected clustering algorithm is. Choosing a suitable clustering method, however, is non-trivial.

UDS is also based on CE. In contrast to CMI, it does address all requirements of a universal measure.

4 Universal Dependency Analysis

In short, UDS builds upon and non-trivially extends CMI to fulfil universality. More specifically, to bring correlation scores to the same scale – regardless of the number as well as statistical properties of dimensions involved, we first perform normalisation of the scores. Second, we judiciously fix permutation of dimensions, i.e. no search is required. Third, to avoid data clustering in correlation computation we propose *optimal* discretisation to compute conditional CE terms. Our optimisation problem is formulated such that resulting conditional CE values do not overfit. In the following, we focus on the first two aspects of UDS and postpone the third one to Section 5.

4.1 Universal Dependency Score Function Our goal is to normalise the scores such that they fall into the range $[0, 1]$ where 0 means no correlation at all. Note that simply normalising $\text{CMI}(X_{1,\ldots,d})$ by d is not the solution. This is because if we did so, following Lemma 3.2 the normalised score would be upper-bounded by $\max_{\sigma \in \mathcal{F}_d} \sum_{i=2}^{d} h(X_{\sigma(i)})/d$. As $\max_{\sigma \in \mathcal{F}_d} \sum_{i=2}^{d} h(X_{\sigma(i)})$ is dependent on $X_1, \ldots, X_d$, the requirement of unbiased scores is not met. Hence, we instead perform normalisation based on the following observation.

LEMMA 4.1. *For each permutation $\sigma \in \mathcal{F}_d$,*
$\sum_{i=2}^{d} h(X_{\sigma(i)}) - h(X_{\sigma(i)} \mid X_{\sigma(1),\ldots,\sigma(i-1)}) \leq \sum_{i=2}^{d} h(X_{\sigma(i)})$
with equality iff $X_{\sigma(2),\ldots,\sigma(d)}$ are functions of $X_{\sigma(1)}$.

Proof. We postpone the proof to the online Appendix.

With Lemma 4.1, we are now ready to define UDS. In particular, we have:

DEFINITION 4.1. ***Universal Dependency Score (UDS)*** *The* UDS *of $X_{1,\ldots,d}$ is* $\text{UDS}(X_{1,\ldots,d}) = \max_{\sigma \in \mathcal{F}_d} \Phi_\sigma(X_{1,\ldots,d})$
where

$$\Phi_\sigma(X_{1,\ldots,d}) = \frac{\sum_{i=2}^{d} h(X_{\sigma(i)}) - h(X_{\sigma(i)} | X_{\sigma(1),\ldots,\sigma(i-1)})}{\sum_{i=2}^{d} h(X_{\sigma(i)})} \quad (4.4)$$

and with the convention that $\frac{0}{0} = 0$.

That is, $\text{UDS}(X_{1,\ldots,d})$ is the maximum *normalised* correlation over all permutation of $X_{1,\ldots,d}$. It is non-parametric and can capture both linear or non-linear correlations. To show that UDS meets universality, we prove some of its relevant properties below.

LEMMA 4.2. *We have*
- $0 \leq \text{UDS}(X_{1,\ldots,d}) \leq 1$.
- $\text{UDS}(X_{1,\ldots,d}) = 0$ *iff $X_{1,\ldots,d}$ are independent.*
- $\text{UDS}(X_{1,\ldots,d}) = 1$ *iff there exists X_i such that each $X_j \in \{X_{1,\ldots,d}\} \setminus \{X_i\}$ is a function of X_i.*

Proof. We postpone the proof to the online Appendix.

Hence, UDS score of *any* subspace falls in the range $[0, 1]$, which means that we can compare correlation scores across different subspaces. Thus, UDS addresses universality. It also meets the non-negativity and zero score requirements. In addition, as the values of UDS are bounded on both sides we can interpret its scores more easily, which is clearly a desirable property for practical correlation analysis [23].

4.2 Practical UDS To compute UDS, we still need to look for the permutation that maximises the score. There are $d!$ candidate permutations in total. When d is large, the search space is prohibitively large while in general has no

clear structure to optimise over. Thus, to boost efficiency we propose a practical (heuristic) version of UDS. It intuitively fixes a permutation for correlation computation and hence saves time. Our experiments confirm that it works very well in practice. Below we provide its formal definition.

DEFINITION 4.2. ***Practical* UDS**
The practical UDS *of* $X_{1,\dots,d}$ *is*

$$\text{UDS}_{pr}(X_{1,\dots,d}) = \Phi_\sigma(X_{1,\dots,d})$$

where $\sigma \in \mathcal{F}_d$ *is such that* $h(X_{\sigma(1)}) \geq \dots \geq h(X_{\sigma(d)})$.

In other words, UDS_{pr} chooses the permutation corresponding to the sorting of dimensions in descending order of CE values. We now give the intuition behind this design choice.

We see that to compute UDS, we must find the permutation $\pi \in \mathcal{F}_d$ such that $\Phi_\pi(X_{1,\dots,d})$ is maximal. To maximise this term, we should minimise its denominator and maximise its numerator; see Eq. (4.4). For the former, it would most likely help to exclude $h(X_{\sigma(1)})$, the largest unconditional CE term. Thus, we expect a permutation where $X_{\sigma(1)}$ appears first to be good.

For the numerator, we make the following observation. Assume that $h(X_i) \geq h(X_j)$, i.e., X_i is more random than X_j. Then $h(X_k \mid X_i)$ tends to be smaller than $h(X_k \mid X_j)$ [20]. For instance, if X_j is deterministic, $h(X_k \mid X_i) \leq h(X_k \mid X_j) = h(X_k)$. So $h(X_k \mid \cdot)$ tends to get further away from $h(X_k)$ as the conditional part becomes more random, and vice versa.

Now assume that $h(X_k) \geq h(X_i)$. If X_k is after X_i in the permutation, $h(X_k)$ will appear in the numerator. However, $h(X_k \mid \cdot)$ tends to get close to $h(X_k)$ as the conditional part containing X_i is less random, i.e. $h(X_k) - h(X_k \mid \cdot)$ tends to be small.

If in the permutation X_k instead is before X_i, we will have $h(X_i)$ in the numerator. However, $h(X_i \mid \cdot)$ gets further away from $h(X_i)$, i.e., $h(X_i) - h(X_i \mid \cdot)$ tends to be relatively large.

All in all, these suggest that: *dimensions with large* CE *values should be placed before those with small* CE *values to maximise the numerator*. Our experiments reveal that UDS_{pr} works very well in practice. In addition, similarly to UDS its scores are in the range $[0, 1]$. Thus, UDS_{pr} also fulfils universality. Further, it satisfies the non-negativity and zero score requirements.

We will use UDS_{pr} in the rest of this paper and simply call it UDS. In summary, UDS allows for universal correlation assessment of subspaces with potentially different dimensionality as well as different statistical properties of the involved dimensions. In addition, its computation does not require searching for an optimal permutation nor computing all pairwise correlations. Next, we explain how to efficiently and non-parametrically compute UDS on empirical data.

5 Computing UDS

In this section, for exposition we focus on computing $\text{UDS}(X_{1,\dots,d})$. W.l.o.g., we assume that

$$\text{UDS}(X_{1,\dots,d}) = \frac{\sum_{i=2}^{d} h(X_i) - h(X_i \mid X_{1,\dots,i-1})}{\sum_{i=2}^{d} h(X_i)} \quad .$$

That is, the permutation of dimensions in $\text{UDS}(X_{1,\dots,d})$, which is fixed, is assumed to be $X_1, \dots, X_d$.

To compute $\text{UDS}(X_{1,\dots,d})$, we need to compute unconditional CE terms $h(X_i)$ and conditional CE terms $h(X_i \mid X_{1,\dots,i-1})$. Following Section 3.1, $h(X_i)$ where $i \in [2, d]$ can be computed in closed form.

For the conditional CE terms, we propose to compute them by *optimal discretisation*, which has been shown to preserve correlation structures in data [17, 27]. In particular, we formulate the computation of conditional CE terms as optimisation problems where we search for discretisations that *robustly* maximise $\text{UDS}(X_{1,\dots,d})$. Being robust here means that we aim to obtain bins that preserve true correlation in the data, while avoiding seeing structure when there is none, i.e. *overfitting*. Note that optimal discretisation have been proposed by us in [17, 27]. Our current work differ from previous work in that we here introduce a different way to formulate the solution of optimal discretisation. Therewith we are able to provide non-trivial details of the algorithmic approach, which consequently gives us more insight on the runtime complexity. More on this will come shortly. For now, we compute $\text{UDS}(X_{1,\dots,d})$ as follows.

Computing $h(X_2 \mid X_1)$*:* The value of $h(X_2 \mid X_1)$ depends on how we estimate the distribution of X_1, or in other words, how we discretise the realisations of X_1. Here, we propose to search for the discretisation of X_1 that robustly maximises $h(X_2) - h(X_2 \mid X_1)$. As $h(X_2)$ is fixed given the realisations of X_2, maximising $h(X_2) - h(X_2 \mid X_1)$ is equivalent to minimising $h(X_2 \mid X_1)$.

Computing $h(X_i \mid X_{1,\dots,i-1})$ *for* $i \geq 3$*:* Ideally, one would simultaneously search for the optimal discretisations of $X_{1,\dots,i-1}$ that robustly minimise $h(X_i \mid X_{1,\dots,i-1})$. This however is very computationally expensive. We overcome this by observing that, to this end, we have already discretised $X_{1,\dots,i-2}$, and the resulting discretisations are for robustly maximising $\text{UDS}(X_{1,\dots,d})$. Hence, we choose to search for the discretisation of X_{i-1} only. By not re-discretising any dimension already processed, we strongly reduce runtime.

We now prove that the discretisation at each step can be searched efficiently by *dynamic programming*. Naïvely, the optimisation of each step can be cast as: Find the discretisation of X minimising $h(X' \mid I, X)$ where I is the set of dimensions we have already discretised, and $X, X' \in \{X_{1,\dots,d}\} \setminus I$. However, the more random X is, the smaller

$h(X' \mid I, X)$ [20]. In terms of discretisation, this means that when X is discretised into more bins, $h(X' \mid I, X)$ tends to be smaller. Or put differently, this naïve optimisation may prefer solutions with many bins to those with fewer bins, and hence, potentially pick up spurious correlation.

To avoid this issue, we propose to first search for the optimal discretisation at each permissible number of bins. Then, using a *regulariser* we perform model selection to identify the discretisation that best balances between correlation preservation and robustness. Our regulariser takes into account the number of bins and hence alleviates the overfitting issue. The flow is as follow: optimal discretisation first, then model selection.

5.1 Optimal Discretisation Our problem can be stated as: *Given an integer $\lambda \in [1, m]$, find the discretisation of X into λ bins that minimises $h(X' \mid I, X)$ where I is the set of dimensions we have already discretised, and $X, X' \in \{X_{1,\ldots,d}\} \setminus I$.*

Proof of dynamic programming. To prove that dynamic programming is applicable, we prove that the optimal solution to the above problem exhibits optimal substructure.

Formally, let $\mathcal{G}_\lambda$ be the set of possible discretisations on X that produce exactly λ bins. For each $g \in \mathcal{G}_\lambda$, we let $\{b_1^g, \ldots, b_\lambda^g\}$ be the set of bins formed by g. Each bin $b_i^g = (l_i^g, u_i^g]$ where $l_1^g = min(X)$, $u_\lambda^g = max(X)$, and $l_i^g = u_{i-1}^g$ for $i \in [2, \lambda]$. There is a one-to-one correspondence between each discretisation and the set of bins it forms. Thus, we will use both interchangeably.

Note that the dimensions in I have been discretised. Their discrete space consists of hypercubes; each has $|I|$ sides corresponding to $|I|$ dimensions. Let $C_1, \ldots, C_k$ be the non-empty hypercubes. It holds that $k < m$. For each bin b_i^g ($i \in [1, \lambda]$) and hypercube C_j ($j \in [1, k]$), we write $|C_j, b_i^g|$ as the number of data points falling into the $(|I|+1)$-dimensional hypercube made up by extending C_j with b_i^g. Our optimisation problem is equivalent to solving

$$\min_{g \in \mathcal{G}_\lambda} \sum_{i=1}^{\lambda} \sum_{j=1}^{k} \frac{|C_j, b_i^g|}{m} h(X' \mid C_j, b_i^g) \quad . \tag{5.5}$$

Let dsc be the optimal solution and $\{b_1^{dsc}, \ldots, b_\lambda^{dsc}\}$ be its bins. For any bin b, we write $h(X' \mid I, b)$ as $h(X' \mid I)$ computed using only the points falling into b. We have that

$$\begin{aligned} &\sum_{i=1}^{\lambda} \sum_{j=1}^{k} \frac{|C_j, b_i^{dsc}|}{m} h(X' \mid C_j, b_i^{dsc}) \\ &= \frac{|b_\lambda^{dsc}|}{m} \underbrace{\sum_{j=1}^{k} \frac{|C_j, b_\lambda^{dsc}|}{|b_\lambda^{dsc}|} h(X' \mid C_j, b_\lambda^{dsc})}_{h(X' \mid I, b_\lambda^{dsc})} \\ &\quad + \frac{m - b_\lambda^{dsc}}{m} \sum_{i=1}^{\lambda-1} \sum_{j=1}^{k} \frac{|C_j, b_i^{dsc}|}{m - b_\lambda^{dsc}} h(X' \mid C_j, b_i^{dsc}) \quad . \end{aligned} \tag{5.6}$$

must be minimal among all discretisation $g \in \mathcal{G}_\lambda$. We denote the first term on the right hand side of Eq. (5.6) as *Term1* and the second term as *Term2*.

As dsc is optimal, $\{b_1^{dsc}, \ldots, b_{\lambda-1}^{dsc}\}$ is the optimal way to discretize into $\lambda - 1$ bins the values $X \leq l_\lambda^{dsc}$. In other words, these bins are the solution to Eq. (5.5) w.r.t. $\lambda - 1$ and the values $X \leq l_\lambda^{dsc}$. We prove this by contradiction. In particular, we assume that the optimal solution instead is $\{a_1, \ldots, a_{\lambda-1}\} \neq \{b_1^{dsc}, \ldots, b_{\lambda-1}^{dsc}\}$. Then, it holds that

$$\begin{aligned} &\sum_{i=1}^{\lambda-1} \sum_{j=1}^{k} \frac{|C_j, b_i^{dsc}|}{m - b_\lambda^{dsc}} h(X' \mid C_j, b_i^{dsc}) \\ &\qquad > \sum_{i=1}^{\lambda-1} \sum_{j=1}^{k} \frac{|C_j, a_i|}{m - b_\lambda^{dsc}} h(X' \mid C_j, a_i) \quad . \end{aligned}$$

Following Eq. (5.6), this means $\{a_1, \ldots, a_t, b_\lambda^{dsc}\}$ is a better way to discretize X minimising $h(X' \mid I, X)$, which contradicts our assumption on dsc.

Hence, the optimal solution dsc exhibits optimal substructure. This motivates us to build a *dynamic programming* algorithm to solve our problem.

Algorithmic approach. Our method is summarised in Algorithm 1. Here, we first form bins $\{a_1, \ldots, a_\beta\}$ (Line 1). We will explain the rationale of this shortly.

Each term $s[i] = \sum_{j=1}^{i} |a_i|$ is the total support of bins $a_1, \ldots, a_i$. We compute such terms from Lines 6 to 8 for later use in Lines 17 and 18. This step takes $O(\beta)$.

Each term $f[j][i]$ where $1 \leq j \leq i \leq \beta$ is equal to $h(X' \mid I, \bigcup_{k=j}^{i} a_k)$, i.e. $h(X' \mid I)$ computed using data points contained in $\bigcup_{k=j}^{i} a_k$. These terms are analogous to *Term1*. In Lines 9 to 11, we pre-compute them for efficiency purposes. This step in total takes $O(m \log m + m\beta^2)$. Please refer to the online Appendix for the detailed explanation.

Each value $val[\lambda][i]$ where $\lambda \in [1, \beta]$ and $i \in [\lambda, \beta]$ stands for $h(X' \mid I, X)$ computed by optimally merging (discretising) initial bins $a_1, \ldots, a_i$ into λ bins. $b[\lambda][i]$ contains the resulting bins. Our goal is to efficiently compute $val[1 \ldots \beta][\beta]$ and $b[1 \ldots \beta][\beta]$. To do so, from Lines 12 to 14 we first compute $val[1][1 \ldots \beta]$ and $b[1][1 \ldots \beta]$. Then from Lines 15 to 22, we incrementally compute relevant elements of arrays val and b, using the recursive relation described in Eq. (5.6). This is standard dynamic programming. Note that in Line 17, term $\frac{s[i]-s[j]}{s[i]} f[j+1][i]$ corresponds to *Term1* while term $\frac{s[j]}{s[i]} val[\lambda - 1][j]$ corresponds to *Term2*.

The processing from Lines 12 to 22 takes $O(\beta^3)$. As $\beta \ll m$, our algorithm in total takes $O(m \log m + m\beta^2)$.

Remarks. Notice that in our solution, we form initial bins $\{a_1, \ldots, a_\beta\}$ of X where $\beta \ll m$. Ideally, one would start with m bins. However, in the extreme case when all realisations of X are distinct, $h(X' \mid I, X)$ will be zero. This is known as the empty space issue [12]. On the other hand, by pre-partitioning X in to β bins, we ensure that there is

Algorithm 1 UDS Optimal Discretisation

Create initial bins $\{a_1, \ldots, a_\beta\}$ of X
Create a double array $s[1 \ldots \beta]$
Create a double array $f[1 \ldots \beta][1 \ldots \beta]$
Create a double array $val[1 \ldots \beta][1 \ldots \beta]$
Create an array $b[1 \ldots \beta][1 \ldots \beta]$ to store bins
for $i = 1 \rightarrow \beta$ **do**
  $s[i] = \sum_{j=1}^{i} |a_i|$
end for
for $1 \leq j \leq i \leq \beta$ **do**
  $f[j][i] = h(X' \mid I, \bigcup_{k=j}^{i} a_k)$
end for
for $i = 1 \rightarrow \beta$ **do**
  $b[1][i] = \bigcup_{k=1}^{i} a_k$ and $val[1][i] = f[1][i]$
end for
for $\lambda = 2 \rightarrow \beta$ **do**
  for $i = \lambda \rightarrow \beta$ **do**
    $pos = \arg\min_{j \in [1, i-1]} \Omega(j, i, \lambda)$ where $\Omega(j, i, \lambda)$
    $= \left(\frac{s[i]-s[j]}{s[i]} f[j+1][i] + \frac{s[j]}{s[i]} val[\lambda - 1][j] \right)$
    $val[\lambda][i] = \Omega(pos, i, \lambda)$
    Copy all bins in $b[\lambda - 1][pos]$ to $b[\lambda][i]$
    Add $\bigcup_{k=pos+1}^{i} a_k$ to $b[\lambda][i]$
  end for
end for
Return $val[1 \ldots \beta][\beta]$ and $b[1 \ldots \beta][\beta]$

sufficient data for a statistically reliable computation, while also boosting efficiency. Choosing a suitable value for β is a trade-off between accuracy and efficiency.

5.2 Model Selection We propose a regularisation scheme that helps to balance between correlation preservation and robustness. First, we assume that the dimensions of I are respectively discretised into $e_1, \ldots, e_{|I|}$ bins. We pick the best number of bins λ^* as follows,

$$\lambda^* = \arg\min_{\lambda \in [1, \beta]} \frac{h(X'|I,X)}{h(X')} + \frac{H(I,X)}{\log \beta + \sum_{i=1}^{|I|} \log e_i} \tag{5.7}$$

where λ is the number of bins that X is discretised into and $H(I, X)$ is the joint entropy of dimensions in I and *discretised* X. In short, when λ is small, the first term of Eq. (5.7) is large while the second term is small. Conversely, when λ is large, the first term is small while the second term is large. The optimal λ^* yields the best balance between the two terms, i.e. the best balance between the cost of the model and the cost of the data given the model. This will help avoiding choosing many bins when there is no real structure.

6 Related Work

Dependence analysis traditionally deals with two random variables. For this setting, prominent measures include Pearson's correlation, Spearman's correlation, Hilbert-Schmidt independence criterion [9], distance correlation [26], mutual information [5], and maximal information coefficient [24].

To discover multivariate correlations in multi-dimensional data, recently multivariate measures have been proposed. Total correlation [10, 29] is defined based on Shannon entropy. It is a generalisation of mutual information to the multivariate setting. It however tends to give higher dimensional subspaces larger scores, regardless if correlations in such subspaces are strong [5].

Cumulative mutual information (CMI) [16] which uses cumulative entropy is designed specifically for real-valued data. Like total correlation, CMI is biased towards high dimensional subspaces (see Section 3.3).

Quadratic measures of dependency [18, 21, 25] permit empirical computation in closed form. They closely follow the correlation model in Eq. (2.1) and define their scores using multivariate joint distributions. Thus, they are susceptible to the curse of dimensionality. Further, they lack a formal normalisation scheme to address universality.

Recently, Keller et al. [11] propose HICS in the related problem setting. To compute correlation of $X_{1,\ldots,d}$, HICS averages over multiple random runs of the form *diff* $(p(X_i), p(X_i \mid \{X_{1,\ldots,d}\} \setminus \{X_i\}))$ where X_i is selected *randomly* in each run. This causes two issues. First, HICS scores are non-deterministic, making subspace search results potentially unpredictable. Second, by using conditional distributions with $d - 1$ conditions, HICS is also prone to high dimensionality issue.

In earlier work, we proposed MAC [17] – a normalised form of total correlation. The score is based on Shannon entropy over discretised data which MAC obtains by optimising w.r.t. cumulative entropy. With UDS we stay closer to the source, as we define and optimise our score using just cumulative entropy; we only use Shannon entropy to choose the number of bins over which to report the score. Further, whereas MAC needs to optimise the dimension order, UDS avoids this and scales better.

7 Experiments

In this section, we empirically evaluate UDS. In particular, we first study its statistical power on synthetic data sets. Second, as common in subspace search we plug UDS to existing search algorithms [11, 15] to mine correlated subspaces. We evaluate output subspaces both quantitatively using clustering and outlier detection, as well as qualitatively.

We compare to CMI [16], MAC [17], and HICS [11]. As further baseline, we include UDS$_{\neg r}$, a variant of UDS that does *not* use regularisation. For each competitor, we optimise parameter settings according to their respective

papers. For UDS and UDS$_{\neg r}$, the default setting is $\beta = 20$. Like [17, 18, 24], we form initial bins $\{a_1, \ldots, a_\beta\}$ by applying equal-frequency binning. We implemented UDS in Java, and make our code available for research purposes.[1] All experiments were performed single-threaded on i7-4600U CPUs with 16GB RAM. We report wall-clock running times.

7.1 Statistical Power To verify the suitability of UDS to correlation analysis, we first perform statistical tests using synthetic data sets. Here, the null hypothesis is that the data dimensions are statistically independent. To determine the cutoff for testing the null hypothesis, we first generate 100 data sets with *no* correlation. Next, we compute their correlation scores and set the cutoff according to the significance level $\alpha = 0.05$. We then generate 100 data sets with correlation. The power of the measure is the proportion of the 100 new data sets whose correlation scores exceed the cutoff.

We generate each data set with correlation as follows. Let $l = n/2$ where n is the desired dimensionality. We generate $\mathbf{X}_{l\times 1} = \mathbf{A}_{l\times l} \times \mathbf{Z}_{l\times 1}$ where $Z_i \sim \mathit{Gaussian}(0, 1)$ and $\mathbf{A}_{l\times l}$ is fixed with a_{ij} initially drawn from $\mathit{Uniform}[0, 1]$. Here, $\mathbf{X}_{l\times 1}$ and $\mathbf{Z}_{l\times 1}$ are two vectors, each having l dimensions. We let $\{X_1, \ldots, X_l\} = \mathbf{X}_{l\times 1}$. Next, we generate $\mathbf{W}_{l\times 1} = \mathbf{B}_{l\times l} \times \mathbf{X}_{l\times 1}$ where $\mathbf{B}_{l\times l}$ is fixed with b_{ij} initially drawn from $\mathit{Uniform}[0, 0.5]$. Then, using a function f we generate $X_{i+l} = f(W_i) + e_i$ where $i \in [1, l]$, and $e_i \sim \mathit{Gaussian}(0, \sigma)$; we control noise by varying σ. We use four instantiations of f:

$$f_1(x) = 2x + 1 \quad , \qquad f_2(x) = x^2 - 2x \quad ,$$
$$f_3(x) = \log(|x| + 1) \quad , \qquad f_4(x) = \sin(2x) \quad .$$

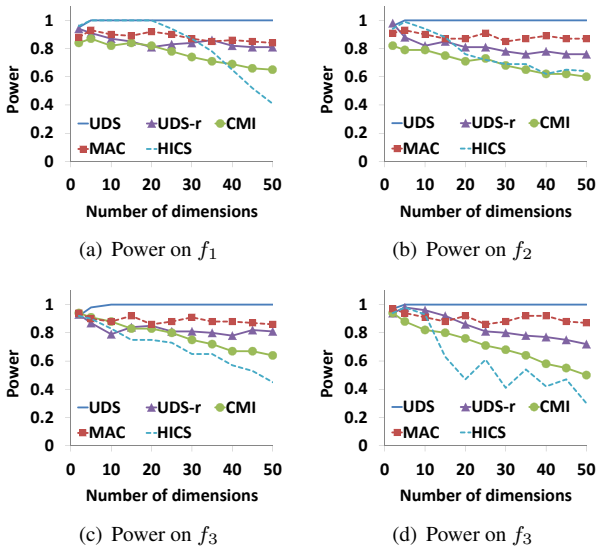
That is, we test with both linear and non-linear correlations.

To study universality, we test with four cases: 1) data sets with as well as without correlation have the same dimensionality n; 2) those with correlation have dimensionality n while those without have dimensionality $n + e$ where e is the number of extra dimensions; 3) those with correlation have dimensionality $n + e$ while those without have dimensionality n; 4) those with as well as without correlation have arbitrary dimensionality. We find 2) and 3) to yield very similar results; hence, we report results of 2) only. For brevity, we further postpone the results of 4) to the online Appendix.

The results for case 1) are in Figure 1. Here, we set $m = 4000$ and vary n. The results for case 2) are in Figure 2. Here, we fix $m = 4000$ and $n = 20$, and vary e.

Going over all results, we see that UDS consistently achieves the best performance in all cases. Moreover, it has from almost perfect to perfect statistical power across different values of data size m, dimensionality n, and the number of extra dimensions e. We outperform MAC most likely because we consistently stick to cumulative entropy instead of transitioning between this entropy notion and Shannon entropy. The clear margin of UDS over UDS$_{\neg r}$ shows the importance of our model selection step (cf., Section 5.2). HICS does not perform so well in high dimensionality perhaps due to its use of high dimensional conditional distributions.

Regarding efficiency, UDS is much faster than MAC and on par with UDS$_{\neg r}$, CMI, and HICS. As UDS clearly outperforms UDS$_{\neg r}$, we skip UDS$_{\neg r}$ in the following.



(a) Power on f_1 (b) Power on f_2

(c) Power on f_3 (d) Power on f_3

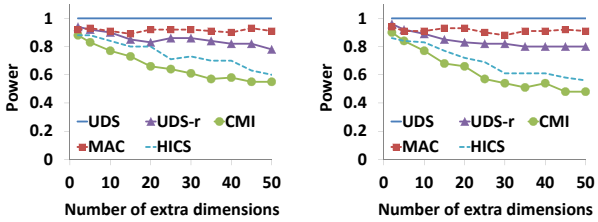
Figure 1: [Higher is better] Statistical power on synthetic data sets for the setting where $m = 4000$ and n is varied.

7.2 Quality of Subspaces – Quantitative Results Here we plug all methods into beam search [11] to find correlated subspaces. As common in subspace search [4, 16, 17], we evaluate quality of output subspaces by each method through clustering, which tends to yield meaningful results on subspaces with high correlations [4, 14].

For each method, we apply DBSCAN [8] – a well-known clustering technique – *on top* of its output subspaces. We follow [1] to aggregate the results of all subspaces. We experiment with 6 real labeled data sets from UCI Repository, regarding their class labels as ground truth. As performance metric, we use F1 measure. The results are in Table 1. We see that UDS performs very well, achieving the best F1 scores on all data sets. This implies that UDS finds better correlated subspaces that help DBSCAN to more accurately discover true clusters.

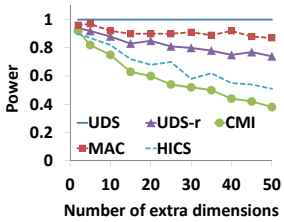
7.3 Quality of Subspaces – Qualitative Results To evaluate the efficacy of UDS in exploratory analysis, we apply it on two real data sets: Communities & Crime from demographic domain [22] and Energy from architecture domain [28]. As these data sets are unlabeled, we cannot assess clustering quality as before. We instead perform subspace

[1] http://eda.mmci.uni-saarland.de/uds/

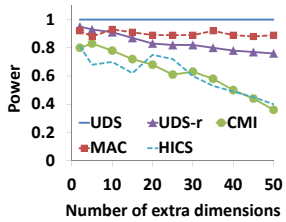


(a) Power on f_1

(b) Power on f_2



(c) Power on f_3



(d) Power on f_3

Figure 2: [Higher is better] Statistical power on synthetic data sets for $m = 4000$, $n = 20$, and varying e.

Data	UDS	CMI	MAC	HICS
Optical	**0.61**	0.40	0.48	0.36
Leaves	**0.70**	0.52	0.61	0.45
Letter	**0.82**	0.64	**0.82**	0.49
PenDigits	**0.85**	0.72	**0.85**	0.71
Robot	**0.54**	0.33	0.46	0.21
Wave	**0.50**	0.24	0.38	0.18
Average	**0.67**	0.48	0.60	0.40

Table 1: [Higher is better] Clustering results (F1 scores) on real-world data sets.

search to detect correlated subspaces, and investigate the discovered correlations. We present some interesting correlations discovered by UDS. All reported correlations are significant at $\alpha = 0.05$ following the testing procedure in [24].

On Communities & Crime, UDS finds a multivariate correlation among % of people in the community with high education, % with low education, % employed as worker, and % employed as manager. This correlation is intuitively understandable. Surprisingly, it is *not* detected by methods other than UDS and MAC. For exposition, we plot some of its 2-D projections in Figure 3. For each correlation pattern in this figure, we plot the function that best fit it – in term of R^2. Two out of three functions are polynomials of degree 5, implying the respective correlation patterns are non-linear.

On Energy, UDS identified a multivariate correlation between outdoor temperature, indoor CO_2 concentration, heating consumption, and drinking water consumption. We plot some of its 2-D projections in Figure 4. The correlation patterns there range from linear (Figure 4(a)) to non-linear (Figures 4(b) and 4(c)). They are also intuitively understandable. Interestingly, no competitor including MAC can detect all of them. For instance, MAC is able to identify the first pattern only. This could be attributed to the fact that MAC computes Shannon entropy over discretised data that it obtains by optimising w.r.t. cumulative entropy.

8 Conclusion

In this paper, we studied the problem of universally, non-parametrically, and efficiently assessing subspace correlations in multivariate data. By universal, we mean that 1) we are able to capture correlation in subspaces of any dimensionality and 2) we allow comparison of correlation scores across different subspaces – regardless how many dimensions they have and what specific statistical properties their dimensions possess. To address all issues, we proposed UDS. In short, we defined UDS based on cumulative entropy. We fulfilled universality by introducing a principled normalisation scheme to bring UDS scores across different subspaces to the same domain. We presented a non-parametric and efficient method to compute UDS on empirical data. Extensive experiments showed that UDS outperformed state of the art in both statistical power and subspace search.

Acknowledgements

The authors are supported by the Cluster of Excellence "Multimodal Computing and Interaction" within the Excellence Initiative of the German Federal Government.

References

[1] I. Assent et al. DUSC: Dimensionality unbiased subspace clustering. In *ICDM*, pages 409–414, 2007.

[2] L. Breiman and J. H. Friedman. Estimating optimal transformations for multiple regression and correlation. *J. Am. Stat. Assoc.*, 80(391):580–598, 1985.

[3] P. Chanda et al. On mining statistically significant attribute association information. In *SDM*, pages 141–152, 2010.

[4] C. H. Cheng et al. Entropy-based subspace clustering for mining numerical data. In *KDD*, pages 84–93, 1999.

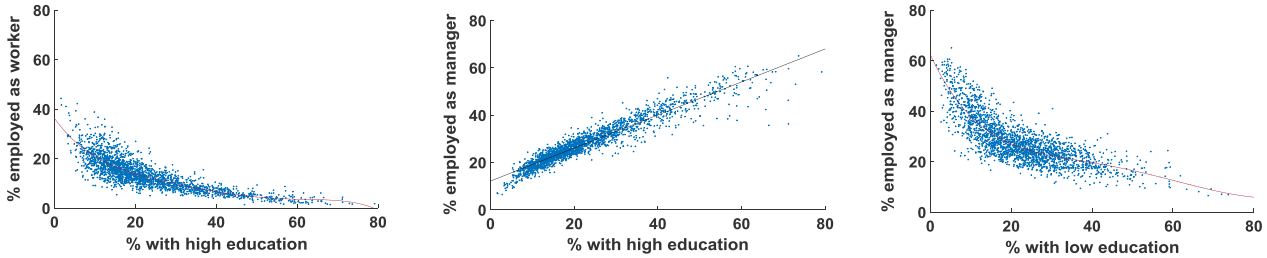
[5] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience New York, 2006.

[6] A. D. Crescenzo and M. Longobardi. On cumulative entropies. *J. Stat. Plan. Inference*, 139(2009):4072–4087, 2009.

[7] N. Drissi et al. Generalized cumulative residual entropy for distributions with unrestricted supports. *Res. Lett. Signal Process.*, 2008:1–5, 2008.

[8] M. Ester et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[9] A. Gretton et al. Measuring statistical dependence with Hilbert-Schmidt norms. In *ALT*, pages 63–77, 2005.

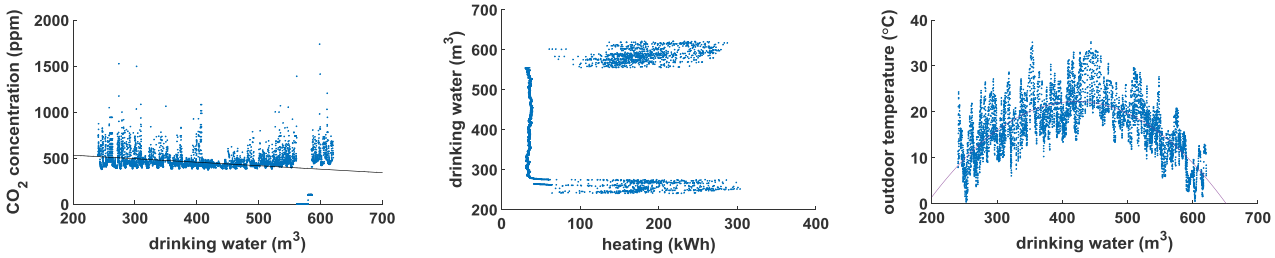


(a) % high education vs. % employed as worker

(b) % high education vs. % employed as manager

(c) % low education vs. % employed as manager

Figure 3: Examples of correlation discovered by UDS on demographic data. Of the competitors, only MAC discovers these patterns.



(a) water consumption vs. CO_2 concentration

(b) heating consumption vs. water consumption

(c) water consumption vs. outdoor temperature

Figure 4: Example correlations discovered by UDS on climate data. Of the competitors, only MAC discovers the first pattern.

[10] T. S. Han. Multiple mutual informations and multiple interactions in frequency data. *Information and Control*, 46(1):26–45, 1980.

[11] F. Keller et al. HiCS: High contrast subspaces for density-based outlier ranking. In *ICDE*, pages 1037–1048, 2012.

[12] J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer, New York, 2007.

[13] J. H. Macke et al. Generating spike trains with specified correlation coefficients. *Neural Computation*, 21(2):397–423, 2009.

[14] E. Müller et al. Evaluating clustering in subspace projections of high dimensional data. *PVLDB*, 2(1):1270–1281, 2009.

[15] H. V. Nguyen et al. 4S: Scalable subspace search scheme overcoming traditional apriori processing. In *BigData Conference*, pages 359–367, 2013.

[16] H. V. Nguyen et al. CMI: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In *SDM*, pages 198–206, 2013.

[17] H. V. Nguyen et al. Multivariate maximal correlation analysis. In *ICML*, pages 775–783, 2014.

[18] H. V. Nguyen et al. Unsupervised interaction-preserving discretization of multivariate data. *Data Min. Knowl. Discov.*, 28(5-6):1366–1397, 2014.

[19] H. V. Nguyen and J. Vreeken. Non-parametric jensen-shannon divergence. In *ECML/PKDD*, 2015.

[20] M. Rao et al. Cumulative residual entropy: A new measure of information. *IEEE Trans. Inf. Theory*, 50(6):1220–1228, 2004.

[21] M. Rao et al. A test of independence based on a generalized correlation function. *Signal Processing*, 91(1):15–27, 2011.

[22] M. Redmond and A. Baveja. A data-driven software tool for enabling cooperative information sharing among police departments. *Eur. J. Oper. Res.*, 141(3):660–678, 2002.

[23] A. Renyi. On measures of dependence. *Acta Mathematica Academiae Scientiarum Hungarica*, 10(3-4):441–451, 1959.

[24] D. N. Reshef et al. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011.

[25] S. Seth et al. A unified framework for quadratic measures of independence. *IEEE Trans. Signal Process.*, 59(8):3624–3635, 2011.

[26] G. J. Székely and M. L. Rizzo. Brownian distance covariance. *Ann. Appl. Stat.*, 3(4):1236–1265, 2009.

[27] J. Vreeken. Causal inference by direction of information. In *SDM*, pages 909–917, 2015.

[28] A. Wagner et al. Performance analysis of commercial buildings – results and experiences from the german demonstration program 'energy optimized building (EnOB)'. *Energy and Buildings*, 68:634–638, 2014.

[29] S. Watanabe. Information theoretical analysis of multivariate correlation. *IBM J. Res. Dev.*, 4:66–82, 1960.

A Proofs

Proof. [Lemma 3.1] By definition, we have

$$\begin{aligned}&\text{CMI}(X_{1,\ldots,d})\\&= \max_{\sigma\in\mathcal{F}_d}\sum_{i=2}^{d} h(X_{\sigma(i)}) - h(X_{\sigma(i)} \mid X_{\sigma(1),\ldots,\sigma(i-1)}).\end{aligned}$$

For each $\sigma \in \mathcal{F}_d$,

$$\begin{aligned}&\sum_{i=2}^{d} h(X_{\sigma(i)}) - h(X_{\sigma(i)} \mid X_{\sigma(1),\ldots,\sigma(i-1)})\\&\leq \sum_{i=2}^{d} h(X_{\sigma(i)}) - h(X_{\sigma(i)} \mid X_{\sigma(1),\ldots,\sigma(i-1)})\\&\qquad + h(X_{d+1}) - h(X_{d+1} \mid X_{\sigma(1),\ldots,\sigma(i)}).\end{aligned}$$

This holds because from Theorem 3.1,

$$h(X_{d+1}) \geq h(X_{d+1} \mid X_{\sigma(1),\ldots,\sigma(i)}).$$

Hence, for each $\sigma \in \mathcal{F}_d$,

$$\begin{aligned}&\sum_{i=2}^{d} h(X_{\sigma(i)}) - h(X_{\sigma(i)} \mid X_{\sigma(1),\ldots,\sigma(i-1)})\\&\leq \max_{\sigma\in\mathcal{F}_{d+1}}\sum_{i=2}^{d+1} h(X_{\sigma(i)}) - h(X_{\sigma(i)} \mid X_{\sigma(1),\ldots,\sigma(i-1)}).\end{aligned}$$

In other words, $\text{CMI}(X_{1,\ldots,d}) \leq \text{CMI}(X_{1,\ldots,d+1})$.

Proof. [Lemma 3.2] From Theorem 3.1, it holds that

$$\begin{aligned}&\sum_{i=2}^{d} h(X_{\sigma(i)}) - h(X_{\sigma(i)} \mid X_{\sigma(1),\ldots,\sigma(i-1)})\\&\leq \sum_{i=2}^{d} h(X_{\sigma(i)})\\&\leq \max_{\sigma\in\mathcal{F}_d}\sum_{i=2}^{d} h(X_{\sigma(i)}).\end{aligned}$$

Hence, we arrive at: $\text{CMI}(X_{1,\ldots,d}) \leq \max_{\sigma\in\mathcal{F}_d}\sum_{i=2}^{d} h(X_{\sigma(i)})$.

Proof. [Lemma 4.1] The result follows from the proof of Lemma 3.1. Based on, Theorem 3.1, the equality holds iff for each $i \in [2,d]$, $X_{\sigma(i)}$ is a function of $X_{\sigma(1),\ldots,\sigma(i-1)}$. This means that

- $X_{\sigma(2)}$ is a function of $X_{\sigma(1)}$.
- $X_{\sigma(3)}$ is a function of $X_{\sigma(1)}$ and $X_{\sigma(2)}$.
- ...
- $X_{\sigma(d)}$ is a function of $X_{\sigma(1),\ldots,\sigma(d-1)}$.

This is equivalent to that $X_{\sigma(2),\ldots,\sigma(d)}$ are functions of $X_{\sigma(1)}$.

Proof. [Lemma 4.2] That $0 \leq \text{UDS}(X_{1,\ldots,d}) \leq 1$ follows from Theorem 3.1 and Lemma 4.1.

$\text{UDS}(X_{1,\ldots,d}) = 0$ iff for each $\sigma \in \mathcal{F}_d$,

$$\sum_{i=2}^{d} h(X_{\sigma(i)}) - h(X_{\sigma(i)} \mid X_{\sigma(1),\ldots,\sigma(i-1)}) = 0.$$

This means that for each $i \in [2,d]$, $X_{\sigma(i)}$ is independent of $X_{\sigma(1),\ldots,\sigma(i-1)}$, which is equivalent to $X_{\sigma(1),\ldots,\sigma(d)}$ are independent.

$\text{UDS}(X_{1,\ldots,d}) = 0$ iff there exists $\sigma \in \mathcal{F}_d$ such that

$$\sum_{i=2}^{d} h(X_{\sigma(i)}) - h(X_{\sigma(i)} \mid X_{\sigma(1),\ldots,\sigma(i-1)}) = \sum_{i=2}^{d} h(X_{\sigma(i)}).$$

Following Lemma 4.1, this means that $X_{\sigma(2),\ldots,\sigma(d)}$ are functions of $X_{\sigma(1)}$.

LEMMA A.1. *It holds that:*
- $0 \leq \text{UDS}_{pr}(X_{1,\ldots,d}) \leq 1$.
- $\text{UDS}_{pr}(X_{1,\ldots,d}) = 0$ *iff* $X_{1,\ldots,d}$ *are independent.*
- $\text{UDS}_{pr}(X_{1,\ldots,d}) = 1$ *iff* $X_{\sigma(2),\ldots,\sigma(d)}$ *are functions of* $X_{\sigma(1)}$.

Proof. That $0 \leq \text{UDS}_{pr}(X_{1,\ldots,d}) \leq 1$ follows from Theorem 3.1 and Lemma 4.1.

$\text{UDS}_{pr}(X_{1,\ldots,d}) = 0$ iff

$$\sum_{i=2}^{d} h(X_{\sigma(i)}) - h(X_{\sigma(i)} \mid X_{\sigma(1),\ldots,\sigma(i-1)}) = 0.$$

This means that for each $i \in [2,d]$, $X_{\sigma(i)}$ is independent of $X_{\sigma(1),\ldots,\sigma(i-1)}$, which is equivalent to $X_{\sigma(1),\ldots,\sigma(d)}$ are independent.

$\text{UDS}_{pr}(X_{1,\ldots,d}) = 1$ iff

$$\sum_{i=2}^{d} h(X_{\sigma(i)}) - h(X_{\sigma(i)} \mid X_{\sigma(1),\ldots,\sigma(i-1)}) = \sum_{i=2}^{d} h(X_{\sigma(i)}).$$

Following Lemma 4.1, this means that $X_{\sigma(2),\ldots,\sigma(d)}$ are functions of $X_{\sigma(1)}$.

B Detailed Complexity Analysis

We pre-sort the values of $X_1,\ldots,X_n$. For each dimension X, we maintain a *rank index* structure $\mathbf{RI}_X$ where $\mathbf{RI}_X[i]$ is the row index of the i^{th} smallest value of X.

When computing $h(X' \mid I, X)$, we pre-partition X in to bins $\{a_1,\ldots,a_\beta\}$. Recall that $C_1,\ldots,C_k$ are the non-empty hypercubes of I. To efficiently compute $f[j][i]$ where

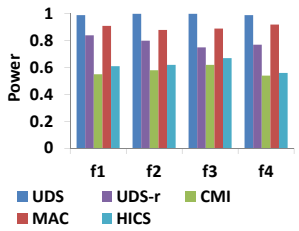


Figure 5: [Higher is better] Statistical power on synthetic data sets for the setting where $m = 4000$ and the dimensionality of each data set created is randomly drawn from $[2, 50]$.

$1 \leq j \leq i \leq \beta$, we have to make sure that values of X' within each combination (C_y, a_z) for $y \in [1, k]$ and $z \in [1, \beta]$ are already sorted. We achieve this by using $\mathbf{RI}_{X'}$. In particular, we loop through each row index, starting from 0. For each index i, we obtain the real row index which is $\mathbf{RI}_{X'}[i]$. Hence, we obtain the respective row. For this row, we retrieve the respective combination (C_y, a_z) that it has. Then, we add X' value of the row to the list of X' values of (C_y, a_z). In this way, we have values of X' sorted in *all* combinations (C_y, a_z).

Now, for each $j \in [1, \beta]$, we run i from j to β. If $i = j$, $f[j][i]$ is in fact $h(X' \mid I)$ computed using the data points in bin a_i. If $i > j$, we compute $f[j][i]$ by merging lists of sorted values X' of $(C_1, a_i), \ldots, (C_k, a_i)$ with the respective list of $(C_1, \bigcup_{z=j}^{i-1} a_z), \ldots, (C_k, \bigcup_{z=j}^{i-1} a_z)$. That is, we merge the list of (C_y, a_i) with that of $(C_y, \bigcup_{z=j}^{i-1} a_z)$. The merge is done efficiently using merge procedure of the well-known merge sort.

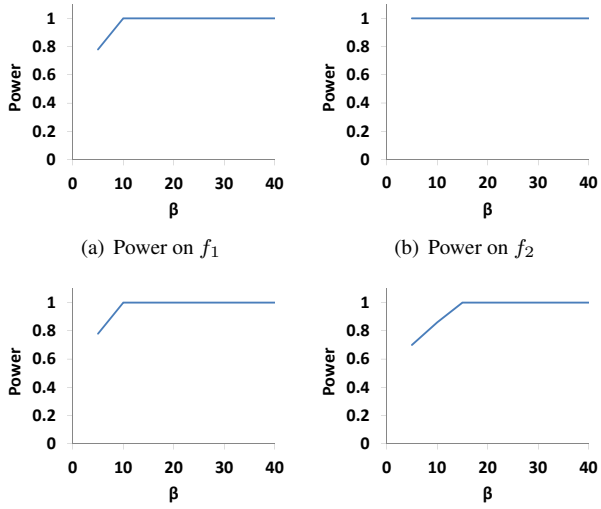
Therefore, computing $f[j][i]$ where $1 \leq j \leq i \leq \beta$ in total takes $O(m \log m + m\beta^2)$.

C Additional Results on Statistical Power

The results for case 4), where data sets with and without correlation have arbitrary dimensionality, are in Figure 5. Here, $m = 4000$ and the dimensionality of each data set created is randomly drawn from $[2, 50]$. We see that UDS consistently achieves the best statistical power across f_1, f_2, f_3, and f_4. This means that it is able to universally quantify correlations of subspaces with different dimensionality.

D Sensitivity to β

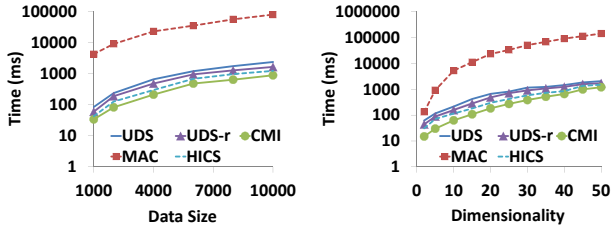
To assess sensitivity of UDS to β, we use the setting of case 1) where data sets with as well as without correlation have the same dimensionality n. In particular, we generate data sets with $m = 4000$ and $n = 20$. We vary β from 5 to 40 with step size being 5. The results are in Figure 6. We see that UDS is very stable for $\beta > 10$. This implies that it allows easy parameterization.



(a) Power on f_1 (b) Power on f_2

(c) Power on f_3 (d) Power on f_4

Figure 6: [Higher is better] Sensitivity to β: Statistical power on synthetic data sets for the setting where $m = 4000$ and $n = 20$.



(a) Runtime vs. m (b) Runtime vs. n

Figure 7: [Lower is better] Scalability to data size m and dimensionality n. The default setting is $m = 4000$ and $n = 20$. The runtime axis is in *log scale*.

E Efficiency Results

For efficiency to data size m, we generate data sets with dimensionality $n = 20$ and m varied. For efficiency to dimensionality n, we generate data sets with size $m = 4000$ and n varied. The results are in Figure 7. We see that UDS scales much better than MAC and on par with other methods. MAC is inefficient because it has to compute all pairwise correlations of a subspace before outputting its final score.

F Results on Outlier Detection

Besides clustering, we also use outlier detection to test quality of correlated subspaces found by each method. Again, we plug all methods into beam search to find correlated subspaces. As common in subspace search [11], for each method we apply LOF, a well-known technique for outlier detection, on top of its output subspaces. Outlier detection

Data	UDS	CMI	MAC	HICS
Ann-Thyroid	**0.98**	0.96	0.96	0.95
Satimage	**0.98**	0.74	0.95	0.86
Segmentation	**0.54**	0.39	0.51	0.49
Wave Noise	**0.51**	0.50	0.50	0.48
WBC	**0.50**	0.47	0.48	0.47
WBCD	**0.99**	0.92	**0.99**	0.91
Average	**0.75**	0.66	0.73	0.69

Table 2: [Higher is better] Outlier detection results (AUC scores) on real-world data sets.

also tends to yield meaningful results on subspaces with high correlations [11, 16].

To show that UDS can work with various data sets, we pick another 6 real labeled data sets – also from UCI Repository – for testing purposes. For each of these data sets, we follow standard procedure in the literature and create outliers by randomly taking 10% of the smallest class. As performance metric, we use AUC (Area under ROC Curve). The results are in Table 2. We see that UDS consistently achieves the best AUC scores on all data sets. This implies that it finds better correlated subspaces that help LOF to more accurately identify true outliers.