Uniform Partitioning of Data Grid for Association Detection

Ali Mousavi, Richard G. Baraniuk, *Fellow, IEEE*

Abstract—Inferring appropriate information from large datasets has become important. In particular, identifying relationships among variables in these datasets has far-reaching impacts. In this paper, we introduce the uniform information coefficient (UIC), which measures the amount of dependence between two multidimensional variables and is able to detect both linear and non-linear associations. Our proposed UIC is inspired by the maximal information coefficient (MIC) [1]; however, the MIC was originally designed to measure dependence between two one-dimensional variables. Unlike the MIC calculation that depends on the type of association between two variables, we show that the UIC calculation is less computationally expensive and more robust to the type of association between two variables. The UIC achieves this by replacing the dynamic programming step in the MIC calculation with a simpler technique based on the uniform partitioning of the data grid. This computational efficiency comes at the cost of not maximizing the information coefficient as done by the MIC algorithm. We present theoretical guarantees for the performance of the UIC and a variety of experiments to demonstrate its quality in detecting associations.

1 Introduction

One of the challenging issues for data scientists is to infer useful information from large datasets containing hundreds of variables which some of them may have interesting but unexplored relationships with each other. This is due to the examples of massive datasets in different areas such as: social networks, astronomy, genomics, medical records, and political science. Hence, it is an important problem to design algorithms that are able to discover associations between different variables in a large dataset.

Measuring the amount of dependence between variables has been extensively studied in the literature and several methods have been proposed for it. Classical works are but not limited to the Pearson correlation coefficient (PCC) [1], correlation ratio [2]–[4], maximal correlation [5], and Spearman correlation coefficient [6].

Mutual Information [7] is a measure that we can use for the quantification of dependency between two variables and has been a focus of several recent works [1], [8]. There are different methods (e.g., kernel density estimation) for estimating mutual information [9]–[19]. A recent line of work for computationally efficient estimation of mutual information is based on the k-nearest neighbors graph or minimal spanning trees of the graph of data points. We refer interested readers to [20]–[22] and references therein.

The *Maximal Information Coefficient (MIC)* [1] is recently proposed for quantifying dependency between two one-dimensional random variables. It is mainly based on calculating the largest possible mutual information between the two variables. The probability distribution corresponding to each variable comes from a two-dimensional grid that the MIC algorithm imposes on the dataset. The MIC bins the dataset in a two-dimensional grid by equipartitioning one axis and using dynamic programming to partition the other axis in order to maximize the mutual information between the two variables. It uses dynamic programming [23] as an optimization method to break the problem of finding the optimal data grid into a sequence of simpler problems. It has two main properties that make it superior in comparison with the aforementioned measures. First, it has *generality* meaning that if the sample size is large enough, it is able to detect different kinds of associations rather than specific types. Second, it is an *equitable* measure meaning that it gives similar scores to equally noisy associations no matter what type the association is.

However, the MIC has three major problems that motivated this work. First, MIC's computational cost grows rapidly as a function of the dataset size. Second, compared to other measures of dependency, the MIC has shown lower statistical power in detecting associations for small size datasets. Third, using MIC for detecting associations between multidimensional variables is computationally expensive. Since MIC's computational cost may become infeasible, Reshef et al. in [1] have applied a heuristic so as not to compute the mutual information for all possible grids. This heuristic application may result in finding a local maximum of mutual information. Authors in [8] introduced the ChiMIC as a new method to compute the MIC values while resolving the aforementioned drawbacks. However, their approach is also limited to one-dimensional variables and as we show later is computationally expensive as well.

In this paper, we develop a new measure of dependency inspired by the MIC. Our measure is based on replacing the dynamic programming application used in the computation

This work was supported in part by NSF grants CCF-1911094, IIS-1838177, and IIS-1730574; ONR grants N00014-18-12571 and N00014-17-1-2551; AFOSR grant FA9550-18-1-0478; DARPA grant G001534-7500; and a Vannevar Bush Faculty Fellowship, ONR grant N00014-18-1-2047.

A. Mousavi is with Google Research, Kirkland, WA 98033 USA and R. G. Baraniuk is with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 USA
Email: alimous@google.com, richb@rice.edu

[1] $\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}$ where $\mathrm{cov}(X,Y)$ is the covariance between X and Y and σ_X and σ_Y are the standard deviation of X and Y, respectively.

of the MIC with uniformly binning the data grid. The trade-off here is that our proposed method is *time efficient* at the cost of not maximizing the mutual information between the two variables. However, with theoretical guarantees, we show that our proposed method is able to detect both functional and non-functional associations between different variables, similar to the MIC while more *time efficiently*. In addition and from the statistical power point of view, we show through a series of experiments that our proposed measure is more powerful than the MIC and gives significantly fewer false positives in detecting associations.

In a conference version of this manuscript [24], we introduced our efficient algorithm for two one-dimensional variables. The current manuscript generalizes our algorithm for detecting associations between two multidimensional variables $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{R}^m$ and $\mathbf{y} = (y_1, \ldots, y_q) \in \mathbb{R}^q$. We should note that detecting associations between two multidimensional variables is also possible by using the original MIC algorithm. Let $\mathbf{x} = (x_1, \ldots, x_m)$ and $\mathbf{y} = (y_1, \ldots, y_q)$ be the two variables that we are interested in detecting associations between them. We can use MIC to check if there is any association between x_i ($1 \leq i \leq m$) and y_j ($1 \leq j \leq q$). While this method needs mq times of running the original MIC algorithm to measure the dependency, our algorithm finds associations with a one-time uniformly partitioning the multidimensional data grid. Therefore, our algorithm is significantly computationally more efficient.

While in this paper we assume that the dataset is fixed, there are important scenarios that we have a dynamic environment (e.g., in recommendation systems) and are interested to detect associations between different variables. One prominent example is the class of networked bandit problems where the goal is to recommend items to users not only based on their individual activity, but also based on the underlying network of relationships between different users. Recent works such as [25] and references therein have explored the idea of graph clustering for this class of problems. Another example is [26] where authors have introduced context aware clustering of bandits for collaborative recommendation tasks. Similar to our work in using a k-nearest neighbors style algorithm for detecting noisy associations, [26] estimates user neighborhoods for collaborative recommendation tasks. Finally, authors in [27] have introduced the idea of using data smashing principles to quantify the association between two data streams.

The rest of this paper is organized as follows. In Section 2, we review the MIC and the algorithm used to compute it from [1]. In Section 3, we introduce our new measure of dependency that is a modification to the MIC. We present experimental results in Section 4. Finally, Section 5 includes the conclusions of the paper and we have included the proofs of our main results in the Appendix.

2 Background

2.1 MIC Definition and Properties

For any finite dataset D which contains ordered pairs of two one-dimensional random variables, one can partition the first element, i.e., x-value of these pairs into ℓ_x bins and similarly partition the second element or y-value of these pairs into ℓ_y bins. As a result of this partitioning, we will have an ℓ_x-by-ℓ_y grid G. Each cell of this grid may or may not contain some sample points from the set D. This grid induces a probability distribution on the cells of G, where the corresponding probability of each cell is equal to the portion of sample points located in that cell. That is,

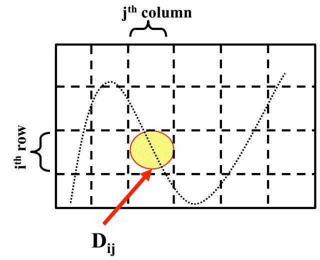


Fig. 1. Partitioning of D into ℓ_x columns and ℓ_y rows. D_{ij} denotes the set of sample points located in the i-th row and the j-th column. Figure is adopted from the conference version of this manuscript [24].

$$p_{ij} = \frac{|D_{ij}|}{|D|}, \tag{1}$$

where p_{ij} denotes the probability corresponding to the cell located at the i^{th} row and the j^{th} column and $|D_{ij}|$ denotes the number of sample points falling into the i-th row and the j-th column. See Figure 1 for a graphical view of the grid G. It is obvious that for each (ℓ_x, ℓ_y), we will have a grid that induces a new probability distribution and hence results in a different mutual information between the two variables.

Let $I^*_{D|G}(P;Q) = \max_G I_{D|G}(P;Q)$ be the largest possible mutual information achievable by an ℓ_x-by-ℓ_y grid G on a set D of sample points. P and Q are the partitions of x-axis and y-axis of grid G, respectively. In order to have a fair comparison among different grids, the computed values of mutual information should be normalized. Since $I(P;Q) = H(Q) - H(Q|P) = H(P) - H(P|Q)$, we divide $I^*_{D|G}(P;Q)$ by $\log(\min(\ell_x, \ell_y))$. Therefore, we have

$$0 \leq \frac{I^*_{D|G}(P;Q)}{\log(\min(\ell_x, \ell_y))} \leq 1. \tag{2}$$

This inequality motivates the definition of the MIC as a measure of dependency between two variables. For a dataset D containing n samples of two one-dimensional variables,

$$\mathrm{MIC}(D) = \max_{\ell_x \ell_y < B(n)} \frac{I^*_{D|G}(P;Q)}{\log(\min(\ell_x, \ell_y))}, \tag{3}$$

where $B(n) = n^{0.6}$ or more generally $\omega(1) \leq B(n) \leq O(n^{1-\epsilon})$ [1] where $\omega(.)$ and $O(.)$ are standard time complexity notations. According to this definition, the MIC has the following properties:

- $0 \leq \mathrm{MIC}(D) \leq 1$.
- $\mathrm{MIC}(x, y) = \mathrm{MIC}(y, x)$.
- MIC is invariant under any order-preserving transformation applied to the dataset D.
- MIC is not invariant under the rotation of coordinate axes, e.g., if $y = x$, then $\mathrm{MIC}(D) = 1$. However, after a 45° clockwise rotation of the coordinate axes, we have $y = 0$ and hence $\mathrm{MIC}(D) = 0$.

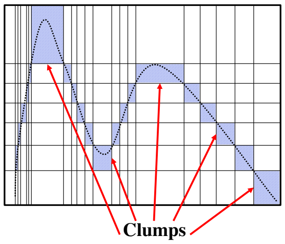


Fig. 2. *OptimizeXAxis* [1] considers only consecutive points falling into the same row and draw partitions between them. The set of consecutive points falling into the same row is called a *clump*.

2.2 MIC Calculation Algorithm

Here we only review the *OptimizeXAxis* algorithm which is used in the computation of the largest mutual information achievable by an ℓ_x-by-ℓ_y grid and refer the interested readers to [1] for the full description of the MIC algorithm. Any ℓ_x-by-ℓ_y grid imposes two sets of partitions on x-values (columns of grid) and y-values (rows of grid). We indicate columns of the grid by $\langle c_1, c_2, \ldots, c_{\ell_x}\rangle$ where c_i denotes the endpoint (largest x-value) of the i-th column.

Since $I(P,Q)$ is upper-bounded by $H(P)$ and $H(Q)$, in order to maximize it, one can equipartition either the x-axis or y-axis, i.e., impose a discrete uniform distribution on either Q or P. Without loss of generality, we consider the version of the algorithm that equipartitions the y-axis. However, it is obvious that we should check both of the cases (equipartitioning either the x-axis or y-axis) separately for each ℓ_x-by-ℓ_y grid and choose the maximum resulting mutual information.

Let $H(P)$ denote the entropy of distribution imposed by m sample points ($m \leq |D| = n$) on the partition of x-axis. Similarly, let $H(Q)$ denote the entropy of distribution imposed by m sample points ($m \leq |D| = n$) on the partition of y-axis. Since we have assumed that the y-axis is equipartitioned, $H(Q)$ is constant and equal to $\log(|Q|)$. Finally, let $H(P,Q)$ denote the entropy of distribution imposed by m sample points ($m < |D| = n$) on the cells of grid G which has x-axis partition P and y-axis partition Q. Since $I(P;Q) = H(Q) - H(Q|P)$ and we have already maximized $H(Q)$ by equipartitioning the y-axis, to achieve the highest mutual information, we have to minimize the $H(Q|P)$. An alternative formula for the mutual information is $I(P;Q) = H(Q) + H(P) - H(P,Q)$. Since $H(Q)$ is constant, the *OptimizeXAxis* only needs to maximize $H(P) - H(P,Q)$. The following theorem [1] is the key to solve this problem.

Theorem 2.1. *For a dataset D of size n and a fixed row partition Q, and for every $m, l \in \mathbb{N}$, if we define $F(m,l) = \max_{D(1:m),|P|=l}\{H(P) - H(P,Q)\}$ then for $l > 1$ and $1 < m \leq n$ we would have the following recursive equation*

$$F(m,l) = \max_{1\leq i<m}\{\frac{i}{m}F(i,l-1) - \frac{m-i}{m}H(\langle i,m\rangle,Q)\}. \quad (4)$$

Proof. See proposition 3.2. in the supplementary file of [1]. □

The *OptimizeXAxis* algorithm (Alg. 2 in the supplementary material of [1]) uses dynamic programming technique motivated by Theorem 2.1 to minimize the $H(Q|P)$. It ensures $F(n,l)$ that is the desired partition of dataset D (which has n sample points) having l columns imposing partition P over x-axis. In order to minimize the $H(Q|P)$, *OptimizeXAxis* considers only consecutive points falling into the same row and draw partitions between them. The set of consecutive points falling into the same row is called *clump*. See Figure 2 for a graphical view of clump.

There are three major drawbacks in *OptimizeXAxis* Algorithm if one wants to use it for detecting associations between two variables. First, it is computationally expensive. If there exists k clumps in the given partition of an ℓ_x-by-ℓ_y grid, the runtime of this algorithm would be $O(k^2)$. If there is a functional association between two variables, then the number of clumps in the corresponding grid is pretty small. However, for noisy or random datasets it is easy to imagine that the number of clumps is very large and hence, the computational complexity of the *OptimizeXAxis* Algorithm would be large.

The second drawback of *OptimizeXAxis* Algorithm is that compared to other statistical measures and for at least small size datasets, it gives a higher false positive rate in detecting associations. In other words, it shows lower statistical power.

The third drawback is that *OptimizeXAxis* Algorithm is exclusively designed for detecting associations between two one-dimensional variables. If we want to use *OptimizeXAxis* Algorithm for association detection in a multidimensional setting, we need to break the problem into one-dimensional setting and check each pair of entries from the two variables separately as we discussed in the introduction. In other words, if the first variable is m-dimensional and the second variable is q-dimensional, we need to run *OptimizeXAxis* Algorithm for mq times that could be computationally expensive. We will illustrate this with examples in Section 4.

3 Algorithm

In the following, we introduce the uniform information coefficient (UIC) as an efficient alternative to the MIC that does not have any of the aforementioned drawbacks. We first describe our algorithm in the noiseless setting and then show how one can use a *k-nearest neighbor* style method to reduce the noise and detect associations more effectively in the noisy setting.

In order to emphasize on the multidimensionality of data points, in the rest of this paper we denote our dataset by $\mathbf{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$ where $\mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$ and $\mathbf{y} = (y^{(1)}, y^{(2)}, \ldots, y^{(q)})$. Here n shows the number of samples and the superscript j as in $x^{(j)}$ denotes the j-th component of $\mathbf{x}$.

3.1 Noiseless Setting

In this section, we describe the UIC calculation algorithm for the noiseless setting. The algorithm we propose in here for replacing the *OptimizeXAxis* Algorithm is *uniform partitioning* (Algorithm 1). Let $y^{(j)}_{\min} = \min_i y^{(j)}_i$ and $y^{(j)}_{\max} = \max_i y^{(j)}_i$ for every $1 \leq j \leq q$ and similarly $x^{(j)}_{\min} = \min_i x^{(j)}_i$ and $x^{(j)}_{\max} = \max_i x^{(j)}_i$ for every $1 \leq j \leq m$. We then partition both $\mathbf{x}$ and $\mathbf{y}$ axes such that all the axes corresponding to $\mathbf{x}$ have length $\frac{x^{(j)}_{\max}-x^{(j)}_{\min}}{\ell_x}$ for every $1 \leq j \leq m$ and similarly all the axes corresponding to $\mathbf{y}$ have length $\frac{y^{(j)}_{\max}-y^{(j)}_{\min}}{\ell_y}$ for every $1 \leq j \leq q$. We call this new measure, that is derived by replacing the *OptimizeXAxis* Algorithm with Algorithm 1, by the uniform information coefficient (UIC). The following proposition shows that when there exists a functional association between two variables (with finite gradient), the UIC will approach 1 as the sample size grows. Without loss of generality, we do all the proofs in the case that $(\mathbf{x}, \mathbf{y}) \in [0,1]^m \times [0,1]^q$. These proofs could be easily generalized to other cases where the support sets of $\mathbf{x}$ and $\mathbf{y}$ have finite volume.

Algorithm 1 UniformPartition(ℓ_x, ℓ_y)

```
Inputs: Dataset $\mathbf{D}$
Parameters: $\ell_x$ and $\ell_y$ are integers greater than 1
Output: Returns a score $I^*$ which is the value of $I(\mathbf{P};\mathbf{Q})$ where $\mathbf{P}$ and $\mathbf{Q}$ are distributions from uniform partitioning of axes corresponding to $\mathbf{X}$ and $\mathbf{Y}$.
$\mathbf{P} \leftarrow$ Uniform partition of $\mathbf{X}$-axes by $\ell_x$ columns such that for every $1 \leq j \leq m$, $\mathbf{X}^{(j)}$ partitions has length $\frac{x^{(j)}_{\max}-x^{(j)}_{\min}}{\ell_x}$
$\mathbf{Q} \leftarrow$ Uniform partition of $\mathbf{Y}$-axes by $\ell_y$ columns such that for every $1 \leq j \leq q$, $\mathbf{Y}^{(j)}$ partitions has length $\frac{y^{(j)}_{\max}-y^{(j)}_{\min}}{\ell_y}$
$I^* = \frac{H(\mathbf{P})+H(\mathbf{Q})-H(\mathbf{P},\mathbf{Q})}{\log(\min(\ell_x,\ell_y))}$
return $I^*$
```

Proposition 3.1. *If* $\mathbf{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ *where* $\mathbf{y} = h(\mathbf{x})$, $h : \mathbb{R}^m \to \mathbb{R}^q$ *and* $0 < |\nabla h^{(j)}(\mathbf{x})| < \infty$ *for* $1 \leq j \leq q$, *then* $\lim_{n\to\infty}$ *UIC*$(\mathbf{D}) = 1$.

Proof. See Appendix A. □

If $\mathbf{x}$ and $\mathbf{y}$ are independent, then according to the following Proposition we have $\lim_{n\to\infty}$ UIC$(\mathbf{D}) = 0$.

Proposition 3.2. *If* $\mathbf{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ *where* $\mathbf{x} \perp\!\!\!\perp \mathbf{y}$, *then* $\lim_{n\to\infty}$ *UIC*$(\mathbf{D}) = 0$.

Proof. See Appendix B. □

3.2 Noisy Setting

In this section we study the performance of the UIC in the noisy setting. We first give a lower-bound on its value when the two variables $\mathbf{x}$ and $\mathbf{y}$ have a noisy functional association in which the noise is bounded. After that, we study the case of unbounded noise.

For the bounded noise case, without loss of generality, we assume that $\mathbf{x}^{(j)} \sim U[0,1]$ for $1 \leq j \leq m$ and the noise has a uniform distribution. Specifically, we assume that sample points $(\mathbf{x}_i, \mathbf{y}_i)$ have the form $(\mathbf{x}_i, h(\mathbf{x}_i) + \mathbf{z}_\epsilon)$ where $\mathbf{z}_\epsilon^{(j)} \sim U[-\epsilon, \epsilon]$ for $1 \leq j \leq q$.

For every j where $1 \leq j \leq q$, we define $\mathbf{y}^{(j)}_{\text{mid}} = \frac{\mathbf{y}^{(j)}_{\max}+\mathbf{y}^{(j)}_{\min}}{2}$. In Algorithm 1, we only divide the $\mathbf{y}$-axis corresponding to $\mathbf{y}^{(j)}$ into two sections by drawing a hyperplane at $\mathbf{y}^{(j)}_{\text{mid}}$. If we call this partition of $\mathbf{y}$ by $\mathbf{Q}^{(j)}$, we can define the corresponding entropy and denote it by $H(\mathbf{Q}^{(j)})$.

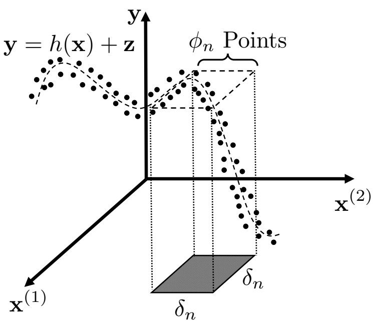


Fig. 3. Using *k-nearest neighbors* method to reduce the effect of noise in noisy relationships. We replace each point with the average of its neighbors in its δ_n-neighborhood.

Among all possibilities of partitioning axes corresponding to $\mathbf{y}$ entries, we consider the one that gives us $j^* = \arg\max_j H(\mathbf{Q}^{(j)})$ (equivalently $H(\mathbf{Q}^*) = \max_j H(\mathbf{Q}^{(j)})$) and accordingly partition axes corresponding to $\mathbf{y}$ into two regions by drawing a hyperplane at $\mathbf{y}^*_{\text{mid}}$.

In addition, similar to Algorithm 1 we divide the axes corresponding to $\mathbf{x}$ into ℓ_x columns each having the length $\frac{1}{\ell_x}$ (since $\mathbf{x}^{(j)} \sim U[0,1]$ for $1 \leq j \leq m$). Let $\mathbf{D}_1 = \{(\mathbf{x}_i, \mathbf{y}_i) | \mathbf{y}_i^{(j^*)} < \mathbf{y}^*_{\text{mid}}\}$ and $\mathbf{D}_2 = \{(\mathbf{x}_i, \mathbf{y}_i) | \mathbf{y}_i^{(j^*)} > \mathbf{y}^*_{\text{mid}}\}$. Similar to the noiseless setting, we use $\mathbf{P}$ and $\mathbf{Q}$ to denote the partition of the axes corresponding to $\mathbf{x}$ and $\mathbf{y}$, respectively. Having this setting and notations in mind, the following Corollary gives a simple lower-bound for the UIC($\mathbf{D}$) in this case.

Corollary 3.3. *Let r be the number of subspaces in $\mathbf{P}$ in which there exists a sample point $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ such that $|\hat{\mathbf{y}}^{(j^*)} - \mathbf{y}^*_{\text{mid}}| \leq \boldsymbol{\epsilon}$. Then, UIC($\mathbf{D}$) is lower-bounded by*

$$\frac{|\mathbf{D}|\log(|\mathbf{D}|) - |\mathbf{D}_1|\log(|\mathbf{D}_1|) - |\mathbf{D}_2|\log(|\mathbf{D}_2|)}{|\mathbf{D}|} - \frac{r}{\ell_x^m}.$$

Proof. See Appendix C □

The main issue with generalizing this lower-bounding idea to other noise distributions is that noise values could be unbounded. Hence, we use the idea of replacing a data point with the average of its *k-nearest neighbors*. In other words, for every $\mathbf{x}_i$ in our dataset, we replace the noisy $\mathbf{y}_i = h(\mathbf{x}_i) + \mathbf{z}_i$ with $\bar{\mathbf{y}}_i = \frac{\sum_{j=i_1}^{i_k} h(\mathbf{x}_j)+\mathbf{z}_j}{k}$, where $\{\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_k}\}$ are the *k-nearest neighbors* to $\mathbf{x}_i$ in the Euclidean space. This idea will help us to reduce the effect of noise so as to come up with a consistent version of the association detector. We study this idea for the case that noise is drawn from a Gaussian distribution $N(\mathbb{0}, \sigma^2\mathbb{I})$.

For each sample point, we consider its δ_n-neighborhood (we use subscript n to show the dependency on the size of the dataset n). We replace each data point with the average of sample points located in its δ_n-neighborhood (Figure 3). The following lemma characterizes the fraction of sample points in this neighborhood.

Lemma 3.4. *Let $\mathbf{x}^{(t)}$ be uniformly distributed for $1 \leq t \leq m$, i.e., $\mathbf{x}^{(t)} \sim U[0,1]$ and $\mathbf{z} \sim N(\mathbb{0}, \sigma^2\mathbb{I})$. If $(\mathbf{x}_i, \mathbf{y}_i)$ denote the i-th data point in $\mathbf{D}$ where $\mathbf{y}_i = h(\mathbf{x}_i) + \mathbf{z}_i$ and $\mathcal{N}_j = \{\mathbf{x}_i \in$*

$\mathbf{D}|\ (\forall t, 1 \leq t \leq m)\ |\mathbf{x}_i^{(t)} - \mathbf{x}_j^{(t)}| < \delta_n\}$, *then* $\lim_{n\to\infty} \frac{|\mathcal{N}_j|}{n} = (2\delta_n)^m$.

Proof. See Appendix D. □

Lemma 3.4 lets us estimate the number of data points in δ_n-neighborhood of every sample as $\phi_n = n(2\delta_n)^m$ in the asymptotic setting.

Assume that each of $h(\cdot)$'s components is a Lipschitz continuous function of order β, i.e., for all $1 \leq t \leq q$ we have

$$|h^{(t)}(\mathbf{v}) - h^{(t)}(\mathbf{w})| \leq k\|\mathbf{v} - \mathbf{w}\|_2^{\beta},$$

where k is a constant which depends on the function $h(\cdot)$. If we estimate and replace the $\mathbf{y}$-value of each noisy sample point with the average of sample points in its δ_n-neighborhood denoted by $\bar{h}(\cdot)$, in the case of Gaussian noise ($\mathbf{z} \sim N(\mathbb{0}, \sigma^2\mathbb{I})$) we can bound the estimation risk as

$$\begin{aligned}
\Delta_n &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\|\bar{h}(\mathbf{x}_i) - h(\mathbf{x}_i)\|_2^2\right] \qquad (5)\\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{q} \mathbb{E}\left[\left(\bar{h}^{(j)}(\mathbf{x}_i) - h^{(j)}(\mathbf{x}_i)\right)^2\right]\\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{q} \mathbb{E}\left[\left(\frac{\sum_{r_i=s_1}^{s_{\phi_n}} (h^{(j)}(\mathbf{x}_{r_i}) + \mathbf{z}_{r_i}^{(j)})}{\phi_n} - h^{(j)}(\mathbf{x}_i)\right)^2\right]\\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{q} \mathbb{E}\left[\left(\frac{\sum_{r_i=s_1}^{s_{\phi_n}} (h^{(j)}(\mathbf{x}_{r_i}) - h^{(j)}(\mathbf{x}_i)) + \mathbf{z}_{r_i}^{(j)}}{\phi_n}\right)^2\right]\\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{q} \mathbb{E}\left[\left(\frac{\sum_{r_i=s_1}^{s_{\phi_n}} (h^{(j)}(\mathbf{x}_{r_i}) - h^{(j)}(\mathbf{x}_i))}{\phi_n}\right)^2\right]\\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{q} \mathbb{E}\left[\left(\frac{\sum_{r_i=s_1}^{s_{\phi_n}} \mathbf{z}_{r_i}^{(j)}}{\phi_n}\right)^2\right]\\
&\leq \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{q} \mathbb{E}\left[\left(\frac{\sum_{r_i=s_1}^{s_{\phi_n}} k\|\mathbf{x}_{r_i} - \mathbf{x}_i\|_2^{\beta}}{\phi_n}\right)^2\right] + q\frac{\sigma^2}{\phi_n}\\
&\leq \frac{qk^2m^{\beta}\phi_n^{\frac{2\beta}{m}}}{2^{2\beta}n^{\frac{2\beta}{m}}} + q\frac{\sigma^2}{\phi_n}.
\end{aligned}$$

In order to minimize the estimation risk we can take derivative with respect to ϕ_n and set it to 0. Therefore, the ϕ_n which minimizes the estimation risk is

$$\phi_n^* = \sqrt[\frac{2\beta}{m}+1]{\frac{2^{2\beta}\sigma^2}{k^2m^{\beta}}}\, n^{1-\frac{1}{\frac{2\beta}{m}+1}}. \qquad (6)$$

We use this ϕ_n^* later to to bound the noise. The following well-known lemma gives a tail bound for the maximum of Gaussian variables, i.e., noise values in here.

Lemma 3.5. *If* $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$ *are i.i.d. drawn from* $N(\mathbb{0}, \sigma^2\mathbb{I})$, *then* $\mathbb{P}\{\max_{1\leq i\leq n} |\mathbf{z}_i^{(j)}| > t\} \leq 2ne^{\frac{-t^2}{2\sigma^2}}$ *for all* $1 \leq j \leq q$.

By using the k-nearest neighbors method, each $\mathbf{z}_i^{(j)}$ is replaced by $\bar{\mathbf{z}}_i^{(j)}$ which is the average of ϕ_n i.i.d. noise values and hence its variance is reduced by the factor of ϕ_n. This idea motivates the following corollary which lets us to reduce the effect of noise.

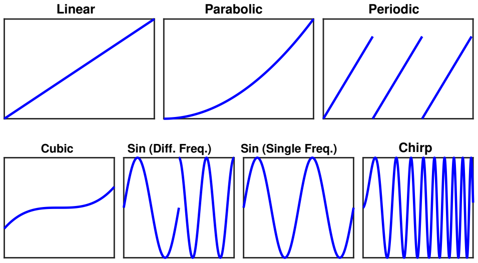


Fig. 4. Functional associations we have used to test the MIC and UIC. Corresponding results are available in Tables 1 and 2.

Corollary 3.6. *By using the k-nearest neighbors method,* $\bar{\mathbf{z}}_i^{(j)} = \frac{\sum_{k=s_1}^{s_{\phi_n}} \mathbf{z}_k^{(j)}}{\phi_n}$, *and as a result* $\lim_{n\to\infty} \max_{1\leq i\leq n} |\bar{\mathbf{z}}_i^{(j)}| = 0$ *for* $1 \leq j \leq q$.

Proof. See Appendix E. □

In the next section we show how the UIC works in practice comparing to the MIC

4 Experimental Results

In this section, we study the performance of our proposed measure. We first focus on one-dimensional variables and show how it works for functional and non-functional associations. Next, we show results from experiments on one-dimensional variables having noisy associations and compare our proposed method with the MIC from statistical power point of view. After that we move to multidimensional variables and study the performance of our proposed measure in the multidimensional setting. We refer interested readers to [1] for a comprehensive comparison between the MIC and other measures we are not reporting them here.

As we mentioned previously, since the computational complexity of the dynamic programming step in the *OptimizeXAxis* Algorithm of the MIC is considerably large, the authors in [1] use a heuristic approximation by restricting the number of clumps. The implementation of this method is available in [1]. In all of our experimental results, we have used this implementation for calculating the MIC values. In addition to the MIC [1], we have also compared our results with a recent modification of the MIC that is called ChiMIC [8]. For the ChiMIC, we have used the MATLAB implementation provided by the authors. Compared to the conference version of this manuscript [24], for calculating the UIC we have used an improved implementation of Algorithm 1. The Python implementation of our work is available online [2]. However, when we compare against other methods, we use a MATLAB implementation to have a fair comparison between different computational complexities.

All values reported in this section are the average of 100 Monte-Carlo sample experiments. We should also mention that entropy calculation in this section is based on the probability assignment we described in (1).

[2] https://github.com/alimousavi1988/UIC

TABLE 1: Values and runtime (in sec) of MIC(D) and UIC(D) for different functional associations in Figure 4. In this set of experiments $|D| = 200$.

Association	$n = 200$					
	MIC	Time	ChiMIC	Time	UIC	Time
Linear	1	0.18	1	0.005	1	0.001
Parabolic	1	0.18	1	0.005	1	0.001
Periodic	1	0.25	1	0.01	0.92	0.001
Cubic	1	0.19	1	0.005	0.95	0.001
Sin (Diff. Freq.)	1	0.22	1	0.01	0.72	0.001
Sin (Single. Freq.)	1	0.39	1	0.01	0.95	0.001
Chirp	0.76	0.4	0.38	0.01	0.31	0.001

TABLE 2: Values and runtime (in sec) of MIC(D) and UIC(D) for different functional associations in Figure 4. In this set of experiments $|D| = 5000$.

Association	$n = 5000$					
	MIC	Time	ChiMIC	Time	UIC	Time
Linear	1	0.75	1	0.7	1	0.18
Parabolic	1	0.76	1	0.7	1	0.18
Periodic	1	9.00	1	0.7	0.998	0.18
Cubic	1	0.77	1	2.3	0.995	0.18
Sin (Diff. Freq.)	1	9.54	1	2.7	0.995	0.18
Sin (Single. Freq.)	1	9.46	1	2.8	0.998	0.18
Chirp	1	9.81	1	3.2	0.86	0.18

4.1 Noiseless Setting

Figure 4 shows functional associations between two one-dimensional variables on which we have tested the performance of the MIC, ChiMIC, and UIC. These associations are linear, quadratic, periodic, cubic, multi-frequency sinusoidal with discontinuity ($\sin(4\pi x)$ for $0 \leq x \leq 0.5$ and $\sin(8\pi x + \pi/2)$ for $0.5 \leq x \leq 1$), single frequency sinusoidal ($\sin(4\pi x)$ for $0 \leq x \leq 1$), and chirp ($\sin(x^{1.6})$ for $0 \leq x \leq 4\pi$). Table 1 summarizes results for the case that there are 200 sample points. For all functional associations except the chirp function MIC(D) = 1 and ChiMIC(D) = 1 in Table 1. However, UIC(D) = 1 only for linear and parabolic associations.

The differences between the MIC and UIC are reasonable for all cases except for the multi-frequency sinusoidal and chirp associations. The main issue with these functions is their high-frequency components. Due to the uniform partitioning of the data grid, the UIC is able to fully capture high-frequency components in functional associations when the grid is fine enough. As a result, as we increase the number of samples and have finer grids, the UIC value catches up with the MIC value. As an example, in Table 2 that we have increased the number of samples to 5000, the difference between the UIC and MIC is almost zero for multi-frequency sinusoidal and significantly smaller compared to Table 1 for the chirp function. In addition, for other functional associations in Table 2, the UIC value is almost equal to the MIC value because of the increased sample size and having finer grids. If we compare the running time values in Tables 1 and 2 we can see that:

- There is a tradeoff between accuracy and time complexity. The UIC is less computationally expensive compared to the MIC and ChiMIC at the cost of not maximizing the mutual information. This is mainly due to using uniform partitioning rather than dynamic programming. In addition, this is the main reason why UIC's outputs do not match the MIC's outputs in the small-sample regime.
- The running time of the UIC algorithm is independent of the type of association while this is not the case for the MIC and ChiMIC. The major reason for this difference is that unlike the UIC, the construction of the initial data grid in the MIC and ChiMIC calculation depends on the type of association.

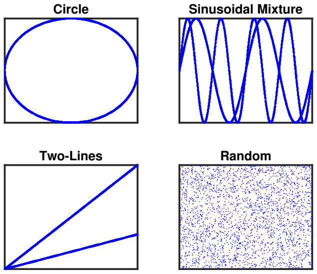


Fig. 5. Non-functional associations we have used to test the MIC and UIC. Corresponding results are available in Tables 3 and 4.

TABLE 3: Values and run time (in sec) for calculation of MIC(D) and UIC(D) for different non-functional relationships in Figure 5. For this set of experiments, $|D| = 200$.

Association	$n = 200$					
	MIC	Time	ChiMIC	Time	UIC	Time
Circle	0.63	0.21	0.6	0.01	0.23	0.001
Sinusoidal Mixture	0.58	0.22	0.55	0.01	0.49	0.001
Two Lines	0.83	0.21	0.82	0.01	0.68	0.001
Random	0.18	0.45	0.06	0.01	0.06	0.001

Tables 3 and 4 summarize results for non-functional associations presented in Figure 5. As we mentioned before, unlike the MIC that uses dynamic programming to strictly maximizes the information coefficient (IC) for each grid size (ℓ_x, ℓ_y), the UIC's algorithm does not strictly maximize the IC for each grid size (ℓ_x, ℓ_y) and instead uses uniform partitioning. In general, this point results in having smaller values for the UIC compared to the MIC. As a result, for detecting associations that we expect the value of the MIC to be significantly small (e.g., random samples or $x \perp\!\!\!\perp y$), the UIC works better than the MIC since it outputs smaller values than the MIC in general.

As an example, the ideal MIC and UIC for random sample points is 0; however, as we can see MIC(D)=0.18 and UIC(D)=0.06 when $n = 200$ and MIC$(D) = 0.07$ and

TABLE 4: Values and run time (in sec) for calculation of MIC(D) and UIC(D) for different non-functional relationships in Figure 5. For this set of experiments, $|D| = 5000$.

Association	$n = 5000$					
	MIC	Time	ChiMIC	Time	UIC	Time
Circle	0.71	16.92	0.7	2.3	0.58	0.18
Sinusoidal Mixture	0.66	16.76	0.64	2.5	0.59	0.18
Two Lines	0.83	9.00	0.8	2.6	0.83	0.18
Random	0.07	19.95	0.01	2.8	0.01	0.18

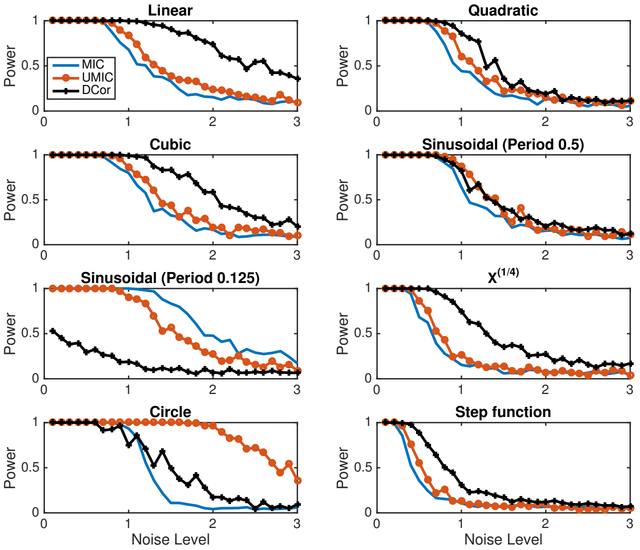


Fig. 6. Power of the MIC and UIC as a function of the level of noise added, for eight different types of associations. We have used 500 Monte-Carlo samples to compute the power in each plot.

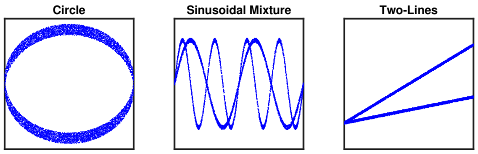


Fig. 7. Noisy non-functional associations we have used to test the MIC and UIC. Corresponding results are available in Tables 5 and 6.

UIC$(D) = 0.01$ when $n = 5000$. This issue is related to one of the criticisms made about the MIC in the literature [28]. One of the drawbacks of using the MIC is that as a statistical test it has a lower power compared to other measures of dependency such as distance correlation [28]. In other words, it gives more false positives in detecting associations. However, according to our experimental results and Proposition 3.2 this issue is alleviated for the UIC.

Figure 6 compares the MIC, UMIC, and distance correlation [29] from the statistical power point of view. As shown in Figure 6, we have considered eight different types of associations with different values of added noise. For each association, we have considered 500 null datasets (i.e., datasets with $\mathbf{x}$ and $\mathbf{y}$ being independent) to estimate our rejection regions for an alternative with level 0.05. Once we had the rejection regions, we used another 200 sample datasets to estimate the power for each association. As we can see in Figure 6, the UIC has larger power than the MIC in all cases except for the high-frequency sinusoidal association. Furthermore, the UIC and distance correlation have comparable performance. In five cases distance correlation outperforms the UIC; while in the other three cases the UIC outperforms distance correlation. Figure 6 basically shows that by using the UIC rather than the MIC we will have significantly fewer false positives in detecting associations between variables of our datasets.

TABLE 5: Values and run time (in sec) for calculation of MIC(D) and UIC(D) for different noisy non-functional relationships in Figure 5. For this set of experiments, $|D| = 200$.

Association	$n = 200$					
	MIC	Time	ChiMIC	Time	UIC	Time
Circle	0.54	0.23	0.53	0.015	0.22	0.001
Sinusoidal Mixture	0.57	0.22	0.54	0.015	0.47	0.001
Two Lines	0.80	0.22	0.75	0.012	0.65	0.001

TABLE 6: Values and run time (in sec) for the calculation of MIC(D) and UIC(D) for different noisy non-functional relationships in Figure 5. In these experiments, $|D| = 5000$.

Association	$n = 5000$					
	MIC	Time	ChiMIC	Time	UIC	Time
Circle	0.51	17.31	0.49	3.9	0.39	0.18
Sinusoidal Mixture	0.64	17.12	0.62	3.9	0.55	0.18
Two Lines	0.74	10.92	0.74	3.7	0.72	0.18

Tables 3 and 4 report running time of the MIC and UIC calculation for non-functional associations as well. In the case of non-functional associations, we have more clumps in the initial grid of sample points for the calculation of the MIC. Hence, if we compare Table 4 with Table 2 we can see that compared to functional associations, it is computationally more expensive to compute the MIC for nonfunctional associations. On the other hand and as we previously mentioned, if we relax the dynamic programming step with uniform partitioning in calculating the IC, we come up with the UIC that approximates the MIC but at the same time is computationally more efficient as a result of this trade-off. Since the UIC is dealing with uniform partitioning of the grid of sample points instead of applying dynamic programming, the type of an association between two variables does not have an impact on its running time.

4.2 Noisy Setting

Tables 5 and 6 summarize results for noisy non-functional associations presented in Figure 7. Associations in Figure 7 are similar to the ones in Figure 5 except for the fact that we have added noise drawn from the uniform distribution $U[-0.05, 0.05]$ to their sample points. Comparing Table 5 with Table 3 and Table 6 with Table 4, we can see that the range of decrease for different associations is almost the same for both MIC and UIC. Tables 5 and 6 show

TABLE 7: UIC$(\mathbf{D})$ for different functional and non-functional associations denoted in Figure 8. In these experiments, $|\mathbf{D}| = 5000$ and $|\mathbf{D}| = 10^5$.

Association	UIC	
	$n = 5000$	$n = 10^5$
Linear	0.89	0.95
Parabolic	0.83	0.87
Cubic	0.88	0.95
Sinusoidal	0.78	0.91
Multi-Frequency Sinusoidal	0.43	0.76
Multi-Line	0.68	0.86
Independent	0.02	0
Spherical	0.44	0.54
Two-Plates	0.49	0.58
Two-Sinusoidal	0.18	0.36

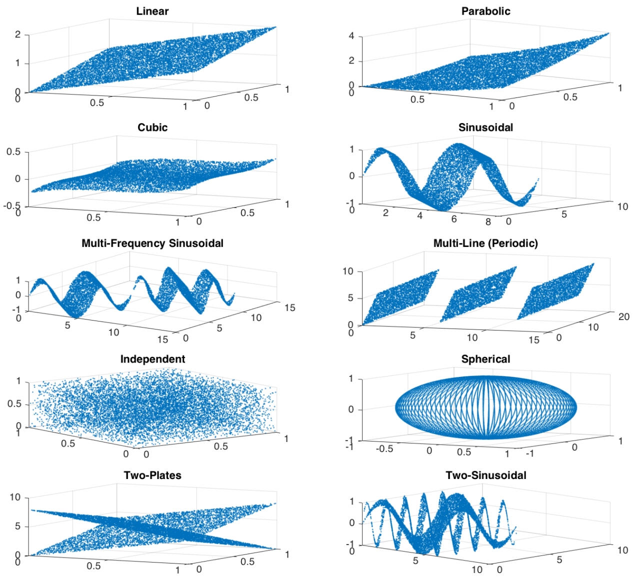


Fig. 8. Different types of multidimensional associations we have used to test the UIC. Corresponding results are available in Tables 7 and 8.

TABLE 8: UIC($\mathbf{D}$), MIC($\mathbf{D}$), and ChiMIC($\mathbf{D}$) and their runtime (in sec) for different functional and non-functional associations denoted in Figure 8. In these experiments, $|\mathbf{D}| = 10^5$.

Association	$n = 10^5$					
$\mathbf{x} \in \mathbb{R}^2, y \in \mathbb{R}$	**UIC**	**Time**	**MIC**	**Time**	**ChiMIC**	**Time**
Linear	0.95	164	0.37	7796	0.36	6972
Parabolic	0.87	165	0.38	7722	0.37	6808
Cubic	0.95	164	0.29	7735	0.29	6717
Sinusoidal	0.91	165	0.01	9476	0	2316
Multi-Frequency Sinusoidal	0.76	165	0.02	9372	0.01	5714
Multi-Line	0.86	165	0.28	7812	0.28	6840
Independent	0	164	0.01	9466	0	3222
Spherical	0.54	165	0.12	9034	0.12	7514
Two-Plates	0.58	165	0.07	9344	0.07	7580
Two-Sinusoidal	0.36	165	0.11	9078	0.1	7528

the running time for calculation of the MIC, ChiMIC, and UIC for noisy associations presented in Figure 7 as well. Since adding noise to sample points increases the number of clumps in the initial grid of sample points, we observe that compared to noiseless non-functional associations (as reported in Table 4), it takes more time to compute the MIC and ChiMIC for noisy non-functional associations (as reported in Table 6). However, since the UIC calculation algorithm performs uniform partitioning on sample points, its computational complexity is robust to adding noise to associations. We should note that as we can see in Tables 3 and 5, adding noise does not have a significant impact on running time of the MIC and ChiMIC calculation algorithm when the number of sample points is rather small.

4.3 Multidimensional Variables

We now move to experimental results for multidimensional variables. Figure 8 shows multidimensional associations on which we have tested the performance of the UIC. In all the cases $\mathbf{x} \in \mathbb{R}^2$ is a two-dimensional variable and $\mathbf{y} \in \mathbb{R}$ is a one-dimensional variable. Extending experimental results to higher dimensions is straightforward and we have chosen this setting for the sake of visualization. Figure 8 contains functional associations (linear, parabolic, cubic, sinusoidal, multi-frequency sinusoidal, periodic multi-line), non-functional associations (spherical, two-plates, two-sinusoidal), and independent variables.

Table 7 reports the UIC value corresponding to different associations in Figure 8 for two sets of experiments where $|\mathbf{D}| = 5000$ and $|\mathbf{D}| = 10^5$. As we can see in Table 7, the value of the UIC for functional associations approaches to 1 as we increase the sample size. Among the functional associations of Figure 8, the UIC shows the weakest association for the multi-frequency sinusoidal relationship. This corresponds to the same issue we previously mentioned about the one-dimensional multi-frequency sinusoidal function in Figure 4. As we increase the sample size we can observe a rapid growth in the value of the UIC for the multi-frequency sinusoidal function (Table 7). For the case of independent variables, the UIC is equal to 0.02 for $|\mathbf{D}| = 5000$ and 0 for $|\mathbf{D}| = 10^5$. In other words, as we increase the sample size, the UIC shows perfect independence. For other non-functional associations the UIC is equal to a value that shows neither perfect independence nor a perfect functional association, as reported in Table 7.

As we mentioned previously, the MIC and ChiMIC are exclusively designed for one-dimensional variables. If we want to use them for a multidimensional setting, one way is to compare all possible pairs of coordinates between the two variables. In other words, if we need to check the existence of any association between $\mathbf{x} = (x_1, \ldots, x_m)$ and $\mathbf{y} = (y_1, \ldots, y_q)$, we should run the MIC and ChiMIC between all x_is ($1 \leq i \leq m$) and y_js ($1 \leq j \leq q$). The problem with this approach is that it needs mq times of running the original MIC algorithm and hence, could be computationally expensive. On the other hand, the UIC calculation algorithm could be easily generalized for a multidimensional setting. Therefore, if $\mathbf{x} = (x_1, \ldots, x_m)$ and $\mathbf{y} = (y_1, \ldots, y_q)$, we do not need to check the UIC between each x_i ($1 \leq i \leq m$) and y_j ($1 \leq j \leq q$) separately and can instead directly calculate the UIC between $\mathbf{x}$ and $\mathbf{y}$.

Table 8 shows examples of this setting on the associations mentioned in Figure 8. If we use the UIC, it gives us a computationally efficient response which takes almost 165 seconds independent of the type of association. However, for the MIC and ChiMIC we need to first measure the association between (x_1, y) and (x_2, y) separately since they do not support the multidimensional setting. Once we have these two separate measures we can combine them (e.g., by taking an average) to come up with a measure which shows how associated y is to $\mathbf{x} = (x_1, x_2)$. In our examples, since both x_1 and x_2 equally impact y, we have considered a simple averaging of $\text{MIC}(x_1, y)$ and $\text{MIC}(x_2, y)$ and reported the numbers in Table 8. By comparing the results in Table 8, we can observe two major points. First, the UIC is significantly more efficient compared to the MIC and ChiMIC in the multidimensional setting. Second, the MIC and ChiMIC are not perfect measures for the multidimensional setting. Since y is affected by both x_1 and x_2, MIC and ChiMIC cannot recognize the perfect relationship between $\mathbf{x} = (x_1, x_2)$ and y by simply considering the relationship

TABLE 9: UIC($\mathbf{D}$) and its runtime (in sec) for several associations where both $\mathbf{x}$ and $\mathbf{y}$ are multidimensional. The UIC value converges to 1 for functional associations as the sample size increases.

Association	$n = 10^5$		$n = 10^6$		$n = 10^7$		$n = 10^8$	
$\mathbf{x} \in \mathbb{R}^5$, $\mathbf{y} \in \mathbb{R}^5$	**UIC value**	**runtime**	**UIC value**	**runtime**	**UIC value**	**runtime**	**UIC value**	**runtime**
Independent $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5), \quad x_i \sim U[0,1]$ $\mathbf{y} = (y_1, y_2, y_3, y_4, y_5), \quad y_i \sim U[0,1]$	0.001	0.2	0.001	4.8	0.001	134.5	0.001	3240.8
Linear $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5), \quad x_i \sim U[0,1]$ $\mathbf{y} = (2x_1, 2x_2, 2x_3, 2x_4, 2x_5)$	1	0.2	1	4.8	1	128.9	1	3268.5
Cubic $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5), \quad x_i \sim U[0,1]$ $\mathbf{y} = (x_1^3, x_2^3, x_3^3, x_4^3, x_5^3)$	0.25	0.2	0.42	4.9	0.65	131.2	1	3296.1
Mixed $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5), \quad x_i \sim U[0,1]$ $\mathbf{y} = (x_1, x_2^2, x_3^3, \sin(x_4), \cos(x_5))$	0.55	0.2	0.65	4.8	0.85	130.9	0.96	3306.2

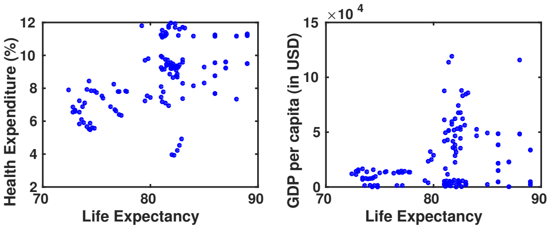


Fig. 9. Life expectancy vs. GDP per capita and government health expenditure for developed countries in 2010-2015. Both plots show a positive correlation meaning that as GDP per capita and health expenditure grow, life expectancy increases as well.

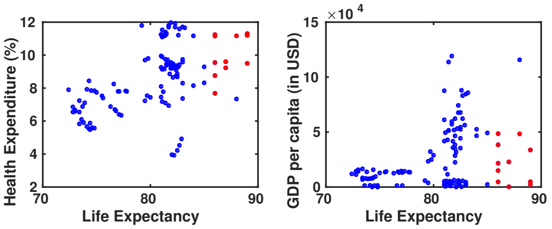


Fig. 10. Red circles show developed countries having the largest life expectancy but not the largest GDP per capita. Instead their governments' health expenditure are among the largest ones.

between (y, x_1) and (y, x_2).

Finally, we study the case where both $\mathbf{x}$ and $\mathbf{y}$ are multidimensional. Table 9 shows the UIC values for several associations in which $\mathbf{x} \in \mathbb{R}^5$ and $\mathbf{y} \in \mathbb{R}^5$. In Table 9 we have mentioned the details of each association since we are not able to fully visualize them in 3D. In particular, we have considered several functional associations (linear, cubic, mixed) and independent variables. In these experiments, we change $|\mathbf{D}|$ from 10^5 to 10^8. According to Table 9, as we increase the sample size the UIC values of the cubic and mixed associations converge to 1. For the linear relationship, the UIC value is equal to 1 constantly. In addition, for independent variables the UIC value is almost 0 constantly and independent of the sample size.

4.4 Real-World Dataset

We consider a public dataset from World Health Organization (WHO) [3] which contains potential factors affecting the life expectancy. From this dataset, we focus on the case of developed countries over five years (2010-2015)[4]. We are interested in observing the association in two cases: first, between the life expectancy and GDP per capita; second, between the life expectancy and government expenditure on health as a percentage of total government expenditure.

TABLE 10: Association between the life expectancy and GDP and health expenditure. Unlike the UIC, MIC and ChiMIC weigh both associations almost equally important.

Life Expectancy vs.	UIC	MIC	ChiMIC
GDP per Capita	0.26	0.54	0.39
Health Expenditure	0.53	0.55	0.40

Figure 9 shows 120 data points in each plot and Table 10 shows the MIC, ChiMIC, and UIC values for these two plots. The MIC and ChiMIC values are almost the same for both plots. However, the UIC value for the relationship between the life expectancy and health expenditure is significantly larger than the one between the life expectancy and GDP per capita. Therefore, compared to the UIC, the MIC and ChiMIC are treating these associations differently. The MIC and ChiMIC describe both associations as almost equally strong while the UIC describes the association between the life expectancy and health expenditure as the stronger one.

While there is a positive correlation between the GDP per capita and life expectancy, there are several developed countries that have the largest life expectancies while their GDP per capita are not among the largest ones. These countries are determined by red circles in the right plot of Figure 10. If we observe the government health expenditure of these countries in the left plot of Figure 10, we notice that their governments health expenditure are among the largest ones as well. This shows that in the WHO dataset there exist developed countries that their life expectancies and health expenditures are among the largest ones but their GDP per capita are not. Therefore, for this dataset one can expect a stronger association between the life expectancy and government health expenditure compared to the GDP per capita. While the UIC results in Table 10 confirm this observation, the MIC and ChiMIC on the other hand, weigh both associations equally. This is another example that shows how our measure could outperform the MIC.

[3] https://www.kaggle.com/kumarajarshi/life-expectancy-who

[4] WHO_Dataset.csv in https://github.com/alimousavi1988/UIC

5 Conclusions

In this paper we introduced the UIC that is a novel measure of association between two multidimensional variables and is able to detect both linear and non-linear associations. While our proposed measure is inspired by the MIC [1], it is different in several ways. The MIC uses dynamic programming to find an optimal grid of data while our approach uses uniform partitioning. This makes our approach computationally more efficient and robust to the type of associations at the cost of not strictly maximizing the mutual information as done by the MIC. Furthermore, we showed that uniform partitioning allows us to simply extend our approach to multidimensional variables. For future work, we plan to study learning-based approaches for detecting associations. In particular, we leave studying neural mutual information estimators [30] and their out of distribution generalization [31] as avenues for the future research.

References

[1] D. Reshef, Y. Reshef, H. Finucane, S. Grossman, G. McVean, P. Turnbaugh, E. Lander, M. Mitzenmacher, and P. Sabeti, "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518–1524, 2011.

[2] H. Cramer, *Mathematical Methods of Statistics*. Princeton Univ Pr, 1999, vol. 9.

[3] A. Kolmogorov, "Grundbegriffe der wahrscheinlichkeitsrechnung," 1933.

[4] A. Rényi, "New version of the probabilistic generalization of the large sieve," *Acta Mathematica Hungarica*, vol. 10, no. 1, pp. 217–226, 1959.

[5] L. Breiman and J. Friedman, "Estimating optimal transformations for multiple regression and correlation," *J. Amer. Statist. Assoc.*, pp. 580–598, 1985.

[6] W. Pirie, "Spearman rank correlation coefficient," *Encyclopedia of Statistical Sciences*, 1988.

[7] T. Cover and J. Thomas, *Elements of information theory*. Wiley Online Library, 1991, vol. 6.

[8] Y. Chen, Y. Zeng, F. Luo, and Z. Yuan, "A new algorithm to optimize maximal information coefficient," *PloS one*, vol. 11, no. 6, p. e0157567, 2016.

[9] Y. Moon, B. Rajagopalan, and U. Lall, "Estimation of mutual information using kernel density estimators," *Phys. Rev.*, vol. 52, no. 3, pp. 2318–2321, 1995.

[10] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev.*, vol. 69, no. 6, pp. 066–138, 2004.

[11] L. Paninski, "Estimation of entropy and mutual information," *Neural Comput.*, vol. 15, no. 6, pp. 1191–1253, 2003.

[12] S. Gao, G. V. Steeg, and A. Galstyan, "Estimating mutual information by local gaussian approximation," *arXiv preprint arXiv:1508.00536*, 2015.

[13] R. Moddemeijer, "On estimation of entropy and mutual information of continuous distributions," *Signal Processing*, vol. 16, no. 3, pp. 233–248, 1989.

[14] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, vol. 11, Apr 1986, pp. 49–52.

[15] Y. L. Chow, "Maximum mutual information estimation of hmm parameters for continuous speech recognition using the n-best algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, Apr 1990, pp. 701–704 vol.2.

[16] S. Khan, S. Bandyopadhyay, A. R. Ganguly, S. Saigal, D. J. Erickson, V. Protopopescu, and G. Ostrouchov, "Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data," *Phys. Rev.*, vol. 76, pp. 026 209–1–026 209–15, Aug 2007.

[17] D. Pál, B. Póczos, and C. Szepesvári, "Estimation of rényi entropy and mutual information based on generalized nearest-neighbor graphs," in *Proc. Adv. in Neural Processing Systems (NIPS)*, 2010, pp. 1849–1857.

[18] S. Gao, G. V. Steeg, and A. Galstyan, "Efficient estimation of mutual information for strongly dependent variables," in *Proc. Int. Conf. Art. Intell. Stat. (AISTATS)*, 2015, pp. 277–286.

[19] K. R. Moon, K. Sricharan, and A. O. Hero, "Ensemble estimation of mutual information," *arXiv preprint arXiv:1701.08083*, 2017.

[20] M. Noshad, K. R. Moon, S. Yasaei Sekeh, and A. O. Hero, "Direct estimation of information divergence using nearest neighbor ratios," in *Proc. Int. Symposium Info. Theory (ISIT)*, 2017, pp. 903–907.

[21] M. Noshad, Y. Zeng, and A. O. Hero, "Scalable mutual information estimation using dependence graphs," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP),*, 2019, pp. 2962–2966.

[22] S. Yasaei Sekeh and A. O. Hero, "Geometric estimation of multivariate dependency," *arXiv preprint arXiv:1905.08594*, 2019.

[23] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[24] A. Mousavi and R. G. Baraniuk, "An information-theoretic measure of dependency among variables in large datasets," in *Proc. Allerton Conf. Communication, Control, and Computing*. IEEE, 2015, pp. 650–657.

[25] S. Li, "The art of clustering bandits," Ph.D. dissertation, Università degli Studi dell'Insubria, 2016.

[26] C. Gentile, S. Li, P. Kar, A. Karatzoglou, G. Zappella, and E. Etrue, "On context-dependent clustering of bandits," in *Proc. Int. Conf. Machine Learning*, 2017, pp. 1253–1262.

[27] I. Chattopadhyay and H. Lipson, "Data smashing: uncovering lurking order in data," *J. Royal Soc. Interface*, vol. 11, no. 101, 2014.

[28] N. Simon and R. Tibshirani, "Comment on detecting novel associations in large data sets by reshef et al, science dec 16, 2011," *arXiv preprint arXiv:1401.7645*, 2014.

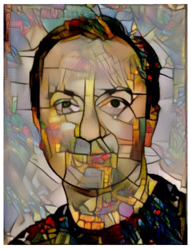
[29] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *Ann. Stat.*, vol. 35, no. 6, pp. 2769–2794, 2007.

[30] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "Mutual information neural estimation," in *Proc. Int. Conf. Machine Learning*, 2018, pp. 531–540.

[31] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, R. L. Priol, and A. Courville, "Out-of-distribution generalization via risk extrapolation (rex)," *arXiv preprint arXiv:2003.00688*, 2020.

Ali Mousavi Ali Mousavi is currently an AI resident in Google Brain. He received his B.Sc. degree in Electrical Engineering from the Sharif University of Technology in 2011 and the M.Sc. and Ph.D. degree in Electrical and Computer Engineering from Rice University in 2014 and 2018, respectively. His research interests include machine learning applications. He has received the Rice University George R. Brown School of Engineering Fellowship, Ken Kennedy Institute Schlumberger Graduate Fellowship, and the Society of Iranian-American Women for Education (SIAWE) Fellowship.

My name is Ali Mousavi



Richard G. Baraniuk Richard G. Baraniuk is the Victor E. Cameron Professor of electrical and computer engineering at Rice University, Houston, TX, USA, and the Founder and Director of OpenStax (openstax.org), Houston, TX, USA. His research interests include the new theory, algorithms, and hardware for sensing, signal processing, and machine learning. He is a Fellow of the AAAS and has received national young investigator awards from the NSF and ONR; the Rosenbaum Fellowship from the Isaac Newton Institute of Cambridge University; the ECE Young Alumni Achievement Award from the University of Illinois; the IEEE Signal Processing Society Best Paper, Best Column, Education, and Technical Achievement Awards; and the IEEE James H. Mulligan, Jr. Medal.

Appendix A
Proof of Proposition 3.1

Proof. For an $\boldsymbol{\alpha} \in \mathbb{R}^q$, we denote by $g_h(\boldsymbol{\alpha})$ the sub-level function of function $h(\cdot)$, i.e.,

$$g_h(\boldsymbol{\alpha}) = \lambda(\{\mathbf{x} : h(\mathbf{x}) \leq \boldsymbol{\alpha}\}), \tag{7}$$

where $\lambda(\mathbb{T})$ denotes the fraction of sample points in the set $\mathbb{T}$. Consequently

$$g_h(\boldsymbol{\alpha}) = F_{\mathbf{y}}(\boldsymbol{\alpha}) = \mathbb{P}(\mathbf{y} \leq \boldsymbol{\alpha}) = \mathbb{P}(h(\mathbf{x}) \leq \boldsymbol{\alpha}), \tag{8}$$

where $F_{\mathbf{y}}(\cdot)$ denotes the cumulative distribution function (CDF) and $\mathbb{P}(\cdot)$ denotes the probability function. Using this notation and assuming that we uniformly partition all axes corresponding to $\mathbf{y}$ by ℓ_y rows, we can write the entropy of resulting partition $\mathbf{Q}$ as

$$\begin{aligned}
&H(\mathbf{Q}) \qquad (9)\\
&= -\sum_{i_1=0}^{\ell_y-1}\sum_{i_2=0}^{\ell_y-1}\cdots\sum_{i_q=0}^{\ell_y-1}\mathbb{P}(\mathbf{Q}=(i_1,i_2,\ldots,i_q))\\
&\qquad\qquad \times \log(\mathbb{P}(\mathbf{Q}=(i_1,i_2,\ldots,i_q)))\\
&= -\sum_{i_1=0}^{\ell_y-1}\sum_{i_2=0}^{\ell_y-1}\cdots\sum_{i_q=0}^{\ell_y-1}\\
&\mathbb{P}\left(\frac{i_1}{\ell_y}\leq y^{(1)}<\frac{i_1+1}{\ell_y},\ldots,\frac{i_q}{\ell_y}\leq y^{(q)}<\frac{i_q+1}{\ell_y}\right)\\
&\times\log\left(\mathbb{P}\left(\frac{i_1}{\ell_y}\leq y^{(1)}<\frac{i_1+1}{\ell_y},\ldots,\frac{i_q}{\ell_y}\leq y^{(q)}<\frac{i_q+1}{\ell_y}\right)\right)\\
&\overset{(a)}{=} -\sum_{i_1=0}^{\ell_y-1}\sum_{i_2=0}^{\ell_y-1}\cdots\sum_{i_q=0}^{\ell_y-1}\frac{1}{\ell_y^q}\frac{\partial^q g_h(\alpha_{i_1},\ldots,\alpha_{i_q})}{\partial\alpha_{i_1}\ldots\partial\alpha_{i_q}}\\
&\qquad\qquad \times\log\left(\frac{1}{\ell_y^q}\frac{\partial^q g_h(\alpha_{i_1},\ldots,\alpha_{i_q})}{\partial\alpha_{i_1}\ldots\partial\alpha_{i_q}}\right)
\end{aligned}$$

where (a) holds according to the mean value theorem and $\frac{i_j}{\ell_y} \leq \alpha_{i_j} < \frac{i_j+1}{\ell_y}$ for all $1 \leq j \leq q$, respectively. If without loss of generality we assume that $\min(\ell_x, \ell_y) = \ell_y$, then we have

$$\begin{aligned}
&\frac{H(\mathbf{Q})}{\log(\ell_y^q)} \qquad (10)\\
&= -\sum_{i_1=0}^{\ell_y-1}\sum_{i_2=0}^{\ell_y-1}\cdots\sum_{i_q=0}^{\ell_y-1}\frac{1}{\ell_y^q\log(\ell_y^q)}\frac{\partial^q g_h(\alpha_{i_1},\ldots,\alpha_{i_q})}{\partial\alpha_{i_1}\ldots\partial\alpha_{i_q}}\\
&\qquad\qquad \times\log\left(\frac{\partial^q g_h(\alpha_{i_1},\ldots,\alpha_{i_q})}{\partial\alpha_{i_1}\ldots\partial\alpha_{i_q}}\right)\\
&+\sum_{i_1=0}^{\ell_y-1}\sum_{i_2=0}^{\ell_y-1}\cdots\sum_{i_q=0}^{\ell_y-1}\frac{1}{\ell_y^q}\frac{\partial^q g_h(\alpha_{i_1},\ldots,\alpha_{i_q})}{\partial\alpha_{i_1}\ldots\partial\alpha_{i_q}}. \qquad (11)
\end{aligned}$$

Therefore, in the asymptotic setting where $n \to \infty$ (and accordingly $\ell_y \to \infty$), we can write

$$\begin{aligned}
&\lim_{\ell_y\to\infty}\frac{H(\mathbf{Q})}{\log(\ell_y^q)}\\
&= \lim_{\ell_y\to\infty}\sum_{i_1=0}^{\ell_y-1}\sum_{i_2=0}^{\ell_y-1}\cdots\sum_{i_q=0}^{\ell_y-1}\frac{1}{\ell_y^q}\frac{\partial^q g_h(\alpha_{i_1},\ldots,\alpha_{i_q})}{\partial\alpha_{i_1}\ldots\partial\alpha_{i_q}}\\
&= 1, \qquad (12)
\end{aligned}$$

since $\lim_{\ell_y\to\infty}\sum_{i_1=0}^{\ell_y-1}\sum_{i_2=0}^{\ell_y-1}\cdots\sum_{i_q=0}^{\ell_y-1}\frac{1}{\ell_y^q}\frac{\partial^q g_h(\alpha_{i_1},\ldots,\alpha_{i_q})}{\partial\alpha_{i_1}\ldots\partial\alpha_{i_q}}$ is the Riemann integral of function $g_h(\cdot)$. Since $h^{(j)} : \mathbb{R}^m \to \mathbb{R}$ and $|\nabla h^{(j)}(\mathbf{x})| < \infty$, for every $1 \leq u \leq m$ there exists a $c < \infty$ such that we have

$$\left|h^{(j)}\left(\mathbf{x}+\frac{\mathbf{e}_u}{\ell_y}\right) - h^{(j)}(\mathbf{x})\right| \leq \frac{c}{\ell_y}, \tag{13}$$

where $\mathbf{e}_u \in \mathbb{R}^m$ is the unit vector in the direction of u-axis. Equation (13) states that for any particular column of the $\mathbf{x}$-axes partition, the curve of the function passes through at most $c+1$ cells of that column. We use this fact in upper-bounding $H(\mathbf{Q}|\mathbf{P})$. Similar to (9), we can write

$$\begin{aligned}
&H(\mathbf{Q}|\mathbf{P}=\mathbf{k}) \qquad (14)\\
&= -\sum_{i_1=0}^{\ell_y-1}\sum_{i_2=0}^{\ell_y-1}\cdots\sum_{i_q=0}^{\ell_y-1}\mathbb{P}\left(\mathbf{Q}=(i_1,\ldots,i_q)|\mathbf{P}=\mathbf{k}\right)\\
&\qquad \times\log\left(\mathbb{P}\left(\mathbf{Q}=(i_1,\ldots,i_q)|\mathbf{P}=\mathbf{k}\right)\right)\\
&= -\sum_{i_1=0}^{\ell_y-1}\sum_{i_2=0}^{\ell_y-1}\cdots\sum_{i_q=0}^{\ell_y-1}\\
&\mathbb{P}\left(\frac{i_1}{\ell_y}\leq y^{(1)}<\frac{i_1+1}{\ell_y},\ldots,\frac{i_q}{\ell_y}\leq y^{(q)}<\frac{i_q+1}{\ell_y}|\mathbf{P}=\mathbf{k}\right)\\
&\log\left(\mathbb{P}\left(\frac{i_1}{\ell_y}\leq y^{(1)}<\frac{i_1+1}{\ell_y},\ldots,\frac{i_q}{\ell_y}\leq y^{(q)}<\frac{i_q+1}{\ell_y}|\mathbf{P}=\mathbf{k}\right)\right)\\
&\overset{(a)}{=} -\sum_{i_1=0}^{\ell_y-1}\sum_{i_2=0}^{\ell_y-1}\cdots\sum_{i_q=0}^{\ell_y-1}\frac{1}{\ell_y^q}f_{\mathbf{y}|\mathbf{x}}(\alpha_{i_1},\ldots,\alpha_{i_q}|\mathbf{P}=\mathbf{k})\\
&\qquad\qquad \log\left(\frac{1}{\ell_y^q}f_{\mathbf{y}|\mathbf{x}}(\alpha_{i_1},\ldots,\alpha_{i_q}|\mathbf{P}=\mathbf{k})\right),\\
&\qquad\qquad\qquad (15)
\end{aligned}$$

where in (a) $f_{\mathbf{y}|\mathbf{x}}(.|.)$ is the conditional probability distribution function (PDF) of $\mathbf{y}$ given $\mathbf{x}$. If we define $\mathbf{k}^* = \arg\max_k H(\mathbf{Q}|\mathbf{P} = \mathbf{k})$, then $H(\mathbf{Q}|\mathbf{P}) = \sum_{\mathbf{k}}\mathbb{P}(\mathbf{P} = \mathbf{k})H(\mathbf{Q}|\mathbf{P} = \mathbf{k}) \leq H(\mathbf{Q}|\mathbf{P} = \mathbf{k}^*)$. Therefore, given (13), we can simplify (14) and write

$$\begin{aligned}
&H(\mathbf{Q}|\mathbf{P}) \qquad (16)\\
&\leq -\sum_{i_1=c_1}^{c_1+c}\sum_{i_2=c_2}^{c_2+c}\cdots\sum_{i_q=c_q}^{c_q+c}\frac{1}{\ell_y^q}f_{\mathbf{y}|\mathbf{x}}(\alpha_{i_1},\ldots,\alpha_{i_q}|\mathbf{P}=\mathbf{k}^*)\\
&\qquad\qquad \log\left(f_{\mathbf{y}|\mathbf{x}}(\alpha_{i_1},\ldots,\alpha_{i_q}|\mathbf{P}=\mathbf{k}^*)\right)\\
&+\sum_{i_1=c_1}^{c_1+c}\sum_{i_2=c_2}^{c_2+c}\cdots\sum_{i_q=c_q}^{c_q+c}\frac{1}{\ell_y^q}f_{\mathbf{y}|\mathbf{x}}(\alpha_{i_1},\ldots,\alpha_{i_q}|\mathbf{P}=\mathbf{k}^*)\log(\ell_y^q).
\end{aligned}$$

Accordingly, in the asymptotic setting where $n \to \infty$ (and accordingly $\ell_y \to \infty$), we can write

$$\begin{aligned}
&\lim_{\ell_y\to\infty}\frac{H(\mathbf{Q}|\mathbf{P})}{\log(\ell_y^q)} \qquad (17)\\
&\leq \lim_{\ell_y\to\infty}\sum_{i_1=c_1}^{c_1+c}\sum_{i_2=c_2}^{c_2+c}\cdots\sum_{i_q=c_q}^{c_q+c}\frac{1}{\ell_y^q}f_{\mathbf{y}|\mathbf{x}}(\alpha_{i_1},\ldots,\alpha_{i_q}|\mathbf{P}=\mathbf{k}^*)\\
&= 0,
\end{aligned}$$

where the last equality holds since $\frac{1}{\ell_y} \to 0$ but $c < \infty$. As a result

$$\begin{aligned}\lim_{\ell_y \to \infty} \text{UIC}(\mathbf{D}) &= \frac{I(\mathbf{P};\mathbf{Q})}{\log(\min\{\ell_x^m, \ell_y^q\})} \qquad (18)\\ &= \frac{H(\mathbf{Q}) - H(\mathbf{Q}|\mathbf{P})}{\log(\min\{\ell_x^m, \ell_y^q\})} = 1.\end{aligned}$$

□

Appendix B
Proof of Proposition 3.2

Proof. Similar to the proof of Proposition 3.1, we uniformly partition all axes corresponding to $\mathbf{y}$ by ℓ_y rows, and all axes corresponding to $\mathbf{x}$ by ℓ_x rows. Let $n_{\mathbf{ij}}$ denote the number of datapoints that fall in the cell where $\mathbf{x} = \mathbf{i}$ and $\mathbf{y} = \mathbf{j}$. We can write $n_{\mathbf{ij}} = \sum_{k=1}^{n} d_k$, where $d_k = 1$ with probability $\mathbb{P}(\mathbf{x} = \mathbf{i}, \mathbf{y} = \mathbf{j})$ and $d_k = 0$ with probability $1 - \mathbb{P}(\mathbf{x} = \mathbf{i}, \mathbf{y} = \mathbf{j})$. Therefore, since $\mathbb{E}[n_{\mathbf{ij}}] = np_{\mathbf{ij}}$ where $p_{\mathbf{ij}} = \mathbb{P}(\mathbf{x} = \mathbf{i}, \mathbf{y} = \mathbf{j})$, according to the Chernoff bound we can write

$$\mathbb{P}\left(\left|\frac{n_{\mathbf{ij}}}{n} - p_{\mathbf{ij}}\right| \geq \delta p_{\mathbf{ij}}\right) \leq 2e^{\frac{-\delta^2 np_{\mathbf{ij}}}{3}}, \tag{19}$$

for all $0 < \delta < 1$. If we define $\epsilon_{\mathbf{ij}} = \frac{\frac{n_{\mathbf{ij}}}{n} - p_{\mathbf{ij}}}{p_{\mathbf{ij}}}$, we can rewrite (19) as

$$\mathbb{P}\left(|\epsilon_{\mathbf{ij}}| \geq \delta\right) \leq 2e^{\frac{-\delta^2 np_{\mathbf{ij}}}{3}}, \tag{20}$$

for all $0 < \delta < 1$. If we let $\delta = \frac{1}{\sqrt{p_{\mathbf{ij}} \log(n)}}$, (20) would turn into

$$\mathbb{P}\left(|\epsilon_{\mathbf{ij}}| \geq \frac{1}{\sqrt{p_{\mathbf{ij}} \log(n)}}\right) \leq 2e^{\frac{-n}{3\log(n)}}. \tag{21}$$

Therefore, if we apply the union bound over different $\mathbf{i}$s and different $\mathbf{j}$s, we have

$$\mathbb{P}\left(|\epsilon_{\mathbf{ij}}| \geq \frac{1}{\sqrt{p_{\mathbf{ij}} \log(n)}}\right) \leq 2\ell_x^m \ell_y^q e^{\frac{-n}{3\log(n)}} \quad \forall \mathbf{i}, \mathbf{j}. \tag{22}$$

Since we bound the size of partitions similar to [1], we have $\ell_x^m \ell_y^q = \mathcal{O}(n^{1-\epsilon})$ and hence $\epsilon_{\mathbf{ij}} \to 0$, in probability.

We can write the mutual information as the Kullback–Leibler (KL) divergence of the product of marginal distributions $\mathbb{P}(\mathbf{Q})\mathbb{P}(\mathbf{P})$ from the joint distribution $\mathbb{P}(\mathbf{Q}, \mathbf{P})$.

$$\begin{aligned}I(\mathbf{P};\mathbf{Q}) &= D_{\text{KL}}(\mathbb{P}(\mathbf{Q},\mathbf{P}), \mathbb{P}(\mathbf{Q})\mathbb{P}(\mathbf{P})) \qquad (23)\\ &= \sum_{\mathbf{i},\mathbf{j}} \frac{n_{\mathbf{ij}}}{n} \log\left(\frac{\frac{n_{\mathbf{ij}}}{n}}{\frac{n_{\mathbf{i}}}{n} \times \frac{n_{\mathbf{j}}}{n}}\right)\\ &= \sum_{\mathbf{i},\mathbf{j}} p_{\mathbf{ij}}(1+\epsilon_{\mathbf{ij}})\log(1+\epsilon_{\mathbf{ij}}) - p_{\mathbf{ij}}(1+\epsilon_{\mathbf{ij}})\log(1+\epsilon_{\mathbf{i}})\\ &\qquad - p_{\mathbf{ij}}(1+\epsilon_{\mathbf{ij}})\log(1+\epsilon_{\mathbf{j}})\\ &\overset{(a)}{=} \sum_{\mathbf{i},\mathbf{j}} p_{\mathbf{ij}}(1+\epsilon_{\mathbf{ij}})(\epsilon_{\mathbf{ij}} - \mathcal{O}(\epsilon_{\mathbf{ij}}^2)) - p_{\mathbf{ij}}(1+\epsilon_{\mathbf{ij}})(\epsilon_{\mathbf{i}} - \mathcal{O}(\epsilon_{\mathbf{i}}^2))\\ &\qquad - p_{\mathbf{ij}}(1+\epsilon_{\mathbf{ij}})(\epsilon_{\mathbf{j}} - \mathcal{O}(\epsilon_{\mathbf{j}}^2)),\end{aligned}$$

where (a) holds because of the Taylor series expansion of $\log(1+\epsilon_{\mathbf{ij}})$. Since $\epsilon_{\mathbf{ij}} \to 0$ in probability, it is easy to show that $\epsilon_{\mathbf{ij}}^2, \epsilon_{\mathbf{i}} = \sum_j \epsilon_{\mathbf{ij}}$, and $\epsilon_{\mathbf{j}} = \sum_i \epsilon_{\mathbf{ij}}$ also converge to 0 in probability and hence, $I(\mathbf{P};\mathbf{Q}) \to 0$. □

Appendix C
Proof of Corollary 3.3

Proof. The proof is similar to the univariate case we presented in [24] and hence, here we present a proof sketch. Since $I(\mathbf{P},\mathbf{Q}) = H(\mathbf{Q}) - H(\mathbf{Q}|\mathbf{P})$, we need to have an upper-bound on $H(\mathbf{Q}|\mathbf{P})$ in order to determine a lower-bound on $I(\mathbf{P},\mathbf{Q})$. According to the entropy definition, we can write

$$H(\mathbf{Q}) = \frac{|\mathbf{D}|\log(|\mathbf{D}|) - |\mathbf{D}_1|\log(|\mathbf{D}_1|) - |\mathbf{D}_2|\log(|\mathbf{D}_2|)}{|\mathbf{D}|}. \tag{24}$$

In addition, as we have shown in the univariate case [24], we can upper-bound the $H(\mathbf{Q}|\mathbf{P})$ as the following

$$H(\mathbf{Q}|\mathbf{P}) \leq \frac{r}{\ell_x^m} \tag{25}$$

The lower-bound is then derived by combining (24) and (25). □

Appendix D
Proof of Lemma 3.4

Proof. Let $\mathbb{I}(\cdot)$ denote the indicator function. We can characterize the fraction of sample points in $\mathcal{N}_j$ as the following

$$\begin{aligned}\epsilon_n^{(j)} &= \frac{|\mathcal{N}_j|}{n} \qquad (26)\\ &= \frac{1}{n}\sum_{i=1}^{N} \mathbb{I}(|\mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)}| \leq \delta_n, \ldots, |\mathbf{x}_i^{(m)} - \mathbf{x}_j^{(m)}| \leq \delta_n).\end{aligned}$$

Therefore we have

$$\mathbb{E}[\epsilon_n^{(j)}] = (2\delta_n)^m.$$

Using the Hoeffding inequality, we can write

$$\mathbb{P}(|\epsilon_n^{(j)} - \mathbb{E}[\epsilon_n^{(j)}]| \geq t) \leq 2e^{-2t^2 n}. \tag{27}$$

If we let $t = \frac{1}{\log n}$, then $\lim_{n\to\infty}(\epsilon_n^{(j)}) = (2\delta_n)^m$. □

Appendix E
Proof of Corollary 3.6

Proof. According to the Lemma 3.5, we can write $\mathbb{P}\{\max_{1\leq i\leq n} |\bar{\mathbf{z}}_i^{(j)}| > t\} \leq 2ne^{\frac{-t^2\phi_n}{2\sigma^2}}$. The result then follows from letting $t = \frac{1}{\log n}$ and $\phi_n = \phi_n^*$ which was derived in (6). □