Data Min Knowl Disc
DOI 10.1007/s10618-014-0350-5

Unsupervised interaction-preserving discretization of multivariate data

Hoang-Vu Nguyen · Emmanuel Müller · Jilles Vreeken · Klemens Böhm

Received: 1 November 2013 / Accepted: 21 March 2014

Abstract Discretization is the transformation of continuous data into discrete bins. It is an important and general pre-processing technique, and a critical element of many data mining and data management tasks. The general goal is to obtain data that retains as much information in the continuous original as possible. In general, but in particular for exploratory tasks, a key open question is how to discretize multivariate data such that significant associations and patterns are preserved. That is exactly the problem we study in this paper. We propose IPD, an information-theoretic method for unsupervised discretization that focuses on preserving multivariate interactions. To this end, when discretizing a dimension, we consider the distribution of the data over all other dimensions. In particular, our method examines consecutive multivariate regions and combines them if (a) their multivariate data distributions are statistically similar, and (b) this merge reduces the MDL encoding cost. To assess the similarities, we propose *ID*, a novel interaction distance that does not require assuming a distribution

Responsible editor: Toon Calders, Floriana Esposito, Eyke Hüllermeier, Rosa Meo.

H.-V. Nguyen (✉) · E. Müller · K. Böhm
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
e-mail: hoang.nguyen@kit.edu

E. Müller
e-mail: emmanuel.mueller@kit.edu

K. Böhm
e-mail: klemens.boehm@kit.edu

J. Vreeken
Max-Planck Institute for Informatics, Saarbrücken, Germany

J. Vreeken
Cluster of Excellence MMCI, Saarland University, Saarbrücken, Germany
e-mail: jilles@mpi-inf.mpg.de

Published online: 04 April 2014

and permits computation in closed form. We give an efficient algorithm for finding the optimal bin merge, as well as a fast well-performing heuristic. Empirical evaluation through pattern-based compression, outlier mining, and classification shows that by preserving interactions we consistently outperform the state of the art in both quality and speed.

Keywords Discretization · Interaction preservation · Pattern mining · Outlier mining · Classification

1 Introduction

Unsupervised discretization is a crucial part of many knowledge discovery tasks and widely used as a pre-processing step in modern data management. Examples where we find discretization under the hood include selectivity estimation (Gunopulos et al. 2000; Tzoumas et al. 2011), entropy-based schema extraction (Yang et al. 2009, 2011), density estimation for subspace mining (Müller et al. 2009), subgroup discovery (Grosskreutz and Rüping 2009; Lemmerich et al. 2013), and correlation analysis (Reshef et al. 2011). In addition, there exist many tasks that require discrete data as input, such as a wide range of pattern mining algorithms (Han et al. 2007), as well as correlation measures using Shannon entropy, such as mutual information and total correlation (Cover and Thomas 2006).

Not only useful for when discrete data is required, unsupervised discretization has increasingly found its way into many areas that traditionally consider continuous data. For example, clustering (Agrawal et al. 1998; Moise and Sander 2008) and outlier detection (Aggarwal and Yu 2001; Akoglu et al. 2012) use discretization techniques for unsupervised learning on multivariate data. In face, any method that works with decision boundaries on continuous domain needs or performs discretization; be it explicitly or implicitly (Grosskreutz and Rüping 2009).

In general, unsupervised discretization aims at transforming continuous data into discrete bins without prior knowledge about any patterns hidden in the data. Well-known examples include equal-width and equal-frequency binning, as well as methods that optimize the binning w.r.t. univariate the data distribution (Kontkanen and Myllymäki 2007). Considering only a single dimension, all of these methods fail to preserve *interactions* among multiple dimensions, i.e., they may unwittingly cut a multivariate distribution into many parts and so destroy essential characteristics of that data. In contrast, we propose to focus on *interaction-preserving discretization* (IPD) and exploit dependencies among dimensions for better multivariate discretization.

To show the importance and difficulty of IPD, let us consider an example where it is impossible to find correct cut points by univariate discretization. Consider the simple toy example in Fig. 1. It features a 3-dimensional data set with 4 clusters. There are interactions among dimensions X_1, X_2, and X_3. The clusters are only detectable when all dimensions are considered together. For illustration purposes, we assume that X_1 initially has 4 bins: b_1, b_2, b_3, and b_4. To discretize this data set while preserving all clusters, one should discretize X_1 into two bins ($X_1 < 0$ and $X_1 \geq 0$), and similarly

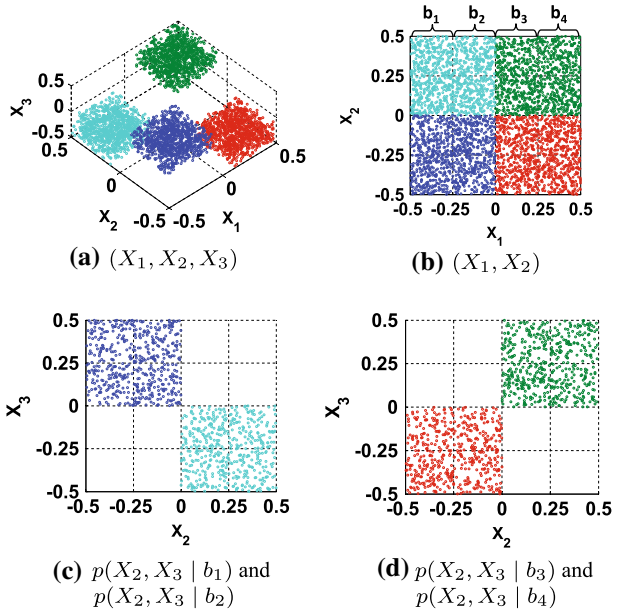

Fig. 1 Example of the parity problem. There are 4 clusters in (X_1, X_2, X_3) marked by *different colors*. The correct discretization: one cut point at 0.0 for each dimension (Color figure online)

for X_2 and X_3. Univariate methods, however, will place cut points randomly w.r.t. the multivariate distribution of the data, and will hence fail to preserve the interactions and clusters. Our method, on the other hand, does identify the correct cut points in all dimensions.

In general, although many real-world applications require multivariate discretization, and while discretization in general is a classic problem, IPD is very much an open research problem. In our work, we introduce a novel discretization approach that aims at preserving interactions. By interaction preservation we mean that two multivariate regions should only be in the same bin if and only if the objects in those regions have similar multivariate joint distributions in the other dimensions. That is, we enforce each bin to only contain data of similar distributions. For instance, in Fig. 1, bins b_1 and b_2 should be merged because the distributions of X_2 and X_3 in the two bins (see Fig. 1c) are similar.

Due to its generality, this problem statement gives way to several open research questions. First, it is unclear how to measure the difference between multivariate distributions in different bins without assuming an underlying distribution. Well-known measures such as Kullback-Leibler divergence (Cover and Thomas 2006) and Earth Mover's Distance (Peleg et al. 1989) require assumptions on the data distribution. Second, one needs a bin merge strategy that balances how well interactions are maintained with the level of detail of the discretization. Third, the search space of all possible discretizations is potentially exponential to the number of data points. As a result, one needs efficient methods to find good discretizations.

With our method, IPD, short for interaction-preserving discretization, we tackle each of these problems. It performs multivariate discretization in the sense that it discretizes one dimension while preserving its interactions with *all* other dimensions.

More in particular, we propose to assess the similarity between multivariate distributions in different bins through a new interaction distance, or *ID* for short. It works on *cumulative distributions* and hence does not require any prior assumption on the data distribution. In addition, its computation on empirical data is in closed form, and we prove that it is a metric distance. This ensures easy-to-interpret distance values and reliable assessment of multivariate distributions.

As the second main ingredient of IPD, we define the task of multivariate discretization in terms of the minimum description length (MDL) principle (Rissanen 1978). By optimizing the resulting objective function, IPD is able to balance between preserving dimension interactions and information of the underlying dimension.

Third, we give two efficient methods for finding good discretizations. The first strategy finds the optimal bin merge by dynamic programming, while the second is a fast greedy heuristic. Though both algorithms have the same theoretical complexity, the latter is by far the faster in practice, while providing very high quality results—in fact, we formally prove that it is a $(2 - \epsilon)$-approximation of our optimal strategy.

Empirical evaluation of IPD on a wide range of real and synthetic data, and unsupervised and supervised tasks including pattern-based compression, outlier mining, and classification clearly shows IPD to consistently outperform the state of the art in both quality and speed. In short, our main contributions include:

(a) General notions and a set of abstract desiderata for interaction-preserving multivariate discretization.
(b) The first interaction distance for quantifying the (dis-)similarity of multivariate distributions over bins, specifically designed for continuous data and computed in closed form on empirical data.
(c) An MDL-based framework for finding a balance between complexity and interaction preservation of a discretization.
(d) Efficient algorithms for discretization, including an optimal solution based on dynamic programming, and a greedy $(2 - \epsilon)$-approximation.

The road map of this paper is as follows. We start by general notions and a set of abstract criteria for interaction-preserving discretization. In Sect. 3 we review related work. Afterward, we introduce IPD, which consists of *ID*, our new interaction distance (Sect. 4), our MDL-based framework for multivariate discretization (Sect. 5), and efficient algorithms (Sect. 6). We empirically evaluate IPD in Sect. 7. We round up with discussion in Sect. 8 and finally conclude in Sect. 9. For readability we postpone the proofs of all theorems to Appendix.

We make our code available for research purposes.[1]

[1] http://www.ipd.kit.edu/~nguyenh/ipd/.

2 General notions

We consider a database $\mathbf{D}$ of n objects and m dimensions. Each dimension $X_i \in \mathbf{A}$, where $\mathbf{A} = \{X_1, \ldots, X_m\}$, is considered as a continuous-valued random variable. We denote the domain of X_i on $\mathbf{D}$ as $[min_i, max_i]$. We write $p(X_i)$ for the probability density function (pdf) of the database projected on X_i. Further, we write $p(x_i)$ for $p(X_i = x_i)$. All logarithms are to base 2, and by convention $0 \log 0 = 0$.

A discretization of X_i into k_i bins induces a set of cut points $K_i = \{c_i^1, \ldots, c_i^{k_i - 1}\}$, which partitions $[min_i, max_i]$ into k_i *bins*, $[min_i, c_i^1]$, $(c_i^1, c_i^2]$, $\ldots$, $(c_i^{k_i-1}, max_i]$.

To preserve interactions, two sets of objects should only be in the same bin if they have similar multivariate joint distributions. That is, one should only consider merging two consecutive bins B^g and B^f of dimension X_i if the distributions of all other dimensions, i.e., $\mathbf{A} \setminus \{X_i\}$, over the objects identified by B^g and B^f are similar. In such a case, we say the bins exhibit similar interactions with regard to the data. To tell whether sets of objects should be in the same bin we need to quantify the interactions of bins. We propose a general notion of interaction distance.

Definition 1 [*Interaction Distance*] Assume we want to measure the interaction distance between bins of dimension X_i over target variables $\{Y_1, \ldots, Y_z\}$. In our setting of unsupervised multivariate discretization, the target variables will typically be $\mathbf{A} \setminus \{X_i\}$, i.e., $\{Y_1, \ldots, Y_z\} = \mathbf{A} \setminus \{X_i\}$. Let $\mathcal{B}_i$ be the set of all possible bins on X_i. An interaction distance should be applicable to any two bins of $\mathcal{B}_i$. As such we have $G : \mathcal{B}_i \times \mathcal{B}_i \to \mathbb{R}_0^+$. In general, an interaction distance $G(B^g, B^f)$ with $B^g, B^f \in \mathcal{B}_i$ quantifies the difference of the distributions over variables $\{Y_1, \ldots, Y_z\}$ in two bins B^g and B^f. The more different they are, i.e., the less B^g and B^f interact, the higher their interaction distance $G(B^g, B^f)$ will be. Formally,

$$G\left(B^g, B^f\right) \sim diff\left(p\left(Y_1, \ldots, Y_z | B^g\right), p\left(Y_1, \ldots, Y_z | B^f\right)\right).$$

In order to facilitate assessment of existing techniques, we introduce four desired properties of meaningful interaction distances.

Property 1 (Unsupervised): G does not require labeled data.
Property 2 (Non-negativity): For any two bins B^g and B^f, $G(B^g, B^f) \geq 0$.
Property 3 (Zero interaction): $G(B^g, B^f) = 0$ if and only if the distributions of $\{Y_1, \ldots, Y_z\}$ in B^g and B^f are identical.
Property 4 (Non-parametric): G should not require prior assumptions on either distributions or correlations.

Properties 1 and 4 are mandatory to ensure the generality of the discretization scheme: to be applicable for exploratory data analysis, to be applicable on unlabeled data, as well as easily computable on empirical data. In particular, one should only use data distribution functions that can be computed directly from empirical data with no prior assumptions or given labels. Properties 2 and 3 guarantee that the distance properly quantifies the difference in data distributions of any two bins.

3 Related work

3.1 Univariate discretization

First, we consider univariate solutions, which include standard approaches such as equal-width and equal-frequency binning, as well as state of the art methods such as UD (Kontkanen and Myllymäki 2007) and its close cousin Bayesian Blocks (Scargle et al. 2013). All of these methods discretize dimensions individually, without considering other dimensions. By definition, they do not preserve interactions. As they neither instantiate G, Properties 1–4 are not applicable.

Supervised techniques (Fayyad and Irani 1993; Kerber 1992) aim to preserve interactions with a target class label. As such, they instantiate $G(B^g, B^f)$ by the difference between the class label distributions of two bins. To measure this difference, ChiM (Kerber 1992) employs χ^2 test. By requiring prior information on class labels, supervised methods do not match Property 1. More importantly, as only interactions with class labels are preserved, the discretized data preserves only structure related to the given class labels. In contrast, IPD aims to preserve all major interactions.

3.2 Multivariate discretization

Ferrandiz and Boullé (2005) proposed a supervised multivariate discretization technique. Though multivariate, its objective function is tightly coupled with the class label distribution.

Kang et al. (2006) introduced an unsupervised multivariate technique based on ICA. However, due to approximation (Seth et al. 2011), ICA transformation is not guaranteed to preserve all important interactions. As a result, it is not guaranteed to fulfill Property 3. Further, it does not meet Property 4 as it implicitly assumes the dimensions to be non-Gaussian in the transformed space.

MVD (Bay 2001) instantiates $G(B^g, B^f)$ by means of Stucco (Bay and Pazzani 1999), a contrast set mining algorithm. Two bins B^g and B^f are considered similar if no itemset can be found that separates the two. However, by considering continuous values as items, the support of most itemsets is very low, which leads to high false alarm rates. Hence, in general, MVD does not meet Property 3. CPD (Mehta et al. 2005) transforms the data using PCA, mines itemsets on the eigenspaces of the bins, and instantiates $G(B^g, B^f)$ as the Jaccard coefficient between the resulting collections. By using PCA it can only capture and preserve linear correlations (Lee and Verleysen 2007). As such, it may miss complex interactions, and is hence not guaranteed to satisfy Property 3. In addition, neither MVD nor CPD are designed to work directly with data distribution functions, and hence neither meet Property 4.

Both MVD and CPD employ a heuristic bottom-up approach as their binning strategy. That is, two consecutive bins are merged if their interaction distance is low. As such, the binning strategy of both MVD and CPD can be viewed as a hierarchical clustering where no objective function is directly optimized. In contrast, univariate methods UD (Kontkanen and Myllymäki 2007) and SD (Fayyad and Irani 1993) search for bins based on the MDL principle (Grünwald 2007). That is, they seek the

bins that yield the best balance between goodness of fit and model complexity. Their encodings are designed for univariate discretization and hence only reward precision, not the preservation of the multivariate interactions of the data.

3.3 Assessing dimension interactions

An interaction distance quantifies differences between two multivariate data distributions. In principle, one could use Kullback–Leibler divergence or a variant, such as Jensen–Shannon divergence (Cover and Thomas 2006). To apply these, one needs to assume a distribution, or estimate the multivariate pdf—which involves hard parametrization (Lee and Verleysen 2007). Advanced density estimation techniques such as kernel methods (Silverman 1986) require selecting a kernel function and bandwidth. Other popular measures include Earth Mover's Distance (Peleg et al. 1989), which requires a probability mass function.

In contrast, we employ cumulative distribution functions (cdfs), which do not require assumptions on the data distribution and do not have free parameters. Further, they can be computed in closed form. In theory, smoother estimates can be obtained through cdf kernel estimation (Liu and Yang 2008). However, similar to pdf kernel estimation, performance depends on the chosen kernel. In addition, though theoretical optimal bandwidth selection exists, its realization needs estimation.

Interactions among dimensions can encompass different types of relationships among dimensions; one of which is correlation. Various correlation measures have been proposed to find correlations hidden in the data (Breiman and Friedman 1985; Reshef et al. 2011), determine independence of dimensions (Rao et al. 2011; Seth et al. 2011), and search for relevant subspaces in high dimensional data (Cheng et al. 1999; Nguyen et al. 2013). Our work here deals with unsupervised discretization and does not directly address such correlation analysis. However, it benefits correlation analysis in the sense that we also aim to preserve interactions among dimensions during the discretization process. Further, many correlation measures are based on Shannon entropy, and hence rely on discrete data. With multivariate discretization, we aim at a general contribution to enhance a variety of techniques, e.g., mutual information and total correlation (Reshef et al. 2011; Cheng et al. 1999), rather than proposing a single solution (Nguyen et al. 2013) improving one specific notion of correlation directly on continuous data.

3.4 Other related work

Lakshmanan et al. (2002) and Bu et al. (2005) studied grouping cells of data cubes satisfying a given property, e.g., frequencies higher than a pre-specified threshold. The methods are designed for cell properties for which the validation does not involve other cells. In other words, they do not check if cells interact, and hence do not address the issue of preserving interactions.

Aue et al. (2009) discussed detecting changes in multivariate time series. Their problem setting is different from ours in two main aspects: (a) they focus on covariance matrices, i.e., second-order pairwise interactions, and (b) break points signify changes

in all dimensions, i.e., each set of cut points correspond to one data point. Philip Preuß (2013) focus on a similar problem, but instead of covariance, targets autocovariance. Like Aue et al. (2009), it is also constrained to pairwise interactions.

The work by Allen (1983); Allen and Ferguson (1994) discusses a representation of time that uses temporal intervals as primitives. It can be used to derive intervals, and for event change detection. While the exact relationship between events can be unknown, some prior knowledge on how they could be temporally related is required.

4 Interaction distance

To construct IPD, we start by introducing our interaction distance, *ID*. Quickly going over its properties, *ID* does not require any prior knowledge such as class labels and hence satisfies Property 1. In the following we will prove that *ID* meets Properties 2 and 3, as well as show that *ID* is computed in closed form on empirical data, i.e., it does not need to make any prior assumption on the data distribution. Therefore, it also meets Property 4. The details of *ID* are as follows.

In principle, when discretizing $X_i \in \mathbf{A}$, to preserve its interactions with all other dimensions, we only consider merging two consecutive bins B^g and B^f of X_i if the distributions of $\mathbf{A} \setminus \{X_i\}$ over the objects identified by B^g and B^f are similar. That is, we instantiate $G(B^g, B^f)$ by $diff\,(p(\mathbf{A} \setminus \{X_i\}|B^g), p(\mathbf{A} \setminus \{X_i\}|B^f))$.

Typically, the *diff* function measures the difference between two multivariate pdfs corresponding to two consecutive bins of any dimension. Formally, we consider two pdfs p and q defined on the set of variables $\mathbf{A} = \{X_1, \ldots, X_m\}$. In practice, we want to measure the differences over $\mathbf{A} \setminus \{X_i\}$. For notational convenience, we however write $diff\,(p(\mathbf{A}), q(\mathbf{A}))$ instead of $diff\,(p(\mathbf{A} \setminus \{X_i\}), q(\mathbf{A} \setminus \{X_i\}))$. The domain of $\mathbf{A}$ is $\Omega = [min_1, max_1] \times \ldots \times [min_m, max_m]$. Though the analysis below considers all dimensions, we note that our discussion holds for any $\mathbf{A} \setminus \{X_i\}$.

To solve the problem of measuring $diff\,(p(\mathbf{A}), q(\mathbf{A}))$ we propose *ID*, a function for quantifying the difference between distributions without requiring any prior assumption. *ID* works with cumulative distributions that can be determined directly from empirical data. That is, *ID* addresses Property 4. In particular, let $P(\mathbf{A})$ and $Q(\mathbf{A})$ be the cumulative distribution function of $p(\mathbf{A})$ and $q(\mathbf{A})$, respectively. That is, for any vector $\mathbf{a} = \{a_1, \ldots, a_m\} \in \Omega$, we have

$$P(\mathbf{a}) = \int_{min_1}^{a_1} \ldots \int_{min_m}^{a_m} p(x_1, \ldots, x_m) dx_1 \cdots dx_m$$

and similarly for $Q(\mathbf{a})$. We define *ID* as follows:

Definition 2 [*Interaction Distance ID*] The interaction distance *ID* between $p(\mathbf{A})$ and $q(\mathbf{A})$, denoted as $ID(p(\mathbf{A}) \,||\, q(\mathbf{A}))$, is defined as $\sqrt{\int_{\Omega} (P(\mathbf{a}) - Q(\mathbf{a}))^2 d\mathbf{a}}$.

In other words, *ID* quantifies the difference between $p(\mathbf{A})$ and $q(\mathbf{A})$ by (a) integrating the squared difference of their respective cumulative distributions, and (b) taking

the square root of this integral. From Definition 2, we can immediately derive the following theorem:

Theorem 1 *ID*$(p(\mathbf{A}) \mid\mid q(\mathbf{A})) \geq 0$ *with equality iff* $p(\mathbf{A}) = q(\mathbf{A})$.

Based on Theorem 1, *ID* meets Properties 2 and 3. Further, we prove that *ID* satisfies the triangle inequality. Let $r(\mathbf{A})$ be a pdf defined on $\mathbf{A}$ and $R(\mathbf{A})$ is its cdf.

Theorem 2 $ID(p(\mathbf{A}) \mid\mid r(\mathbf{A})) + ID(r(\mathbf{A}) \mid\mid q(\mathbf{A})) \geq ID(p(\mathbf{A}) \mid\mid q(\mathbf{A}))$.

Following Theorems 1 and 2, and the fact that *ID* is symmetric, we conclude it is a distance metric. This characteristic ensures easy-to-interpret distance values, as well as reliable assessment of multivariate distributions. Another advantage w.r.t. multivariate discretization is that its measurements on empirical data can be described in closed form. Assume that the empirical data forming $p(\mathbf{A})$ contains data points $\{R_1, \ldots, R_k\}$ of $\mathbf{D}$. Analogously, we denote $\{S_1, \ldots, S_l\}$ as the data points forming $q(\mathbf{A})$. For each $R \in \{R_1, \ldots, R_k\}$, we write R^i for R projected onto the dimension X_i. We define S^i similarly for any $S \in \{S_1, \ldots, S_l\}$. We have:

Theorem 3 *ID*$(p(\mathbf{A}) \mid\mid q(\mathbf{A}))$ *equals to*

$$\left(\frac{1}{k^2} \sum_{j_1=1}^{k} \sum_{j_2=1}^{k} \prod_{i=1}^{m} \left(max_i - \max(R^i_{j_1}, R^i_{j_2}) \right) - \frac{2}{kl} \sum_{j_1=1}^{k} \sum_{j_2=1}^{l} \prod_{i=1}^{m} \left(max_i - \max(R^i_{j_1}, S^i_{j_2}) \right) + \frac{1}{l^2} \sum_{j_1=1}^{l} \sum_{j_2=1}^{l} \prod_{i=1}^{m} \left(max_i - \max(S^i_{j_1}, S^i_{j_2}) \right) \right)^{1/2}.$$

By Theorem 3 we can compute *ID* directly on empirical data in closed form, without assumptions on the data distribution. That is, *ID* meets Property 4.

In the remainder of this paper we will use *ID* to implement *diff* in Definition 1. In particular, we set *diff* $(p(\mathbf{A}), q(\mathbf{A}))$ to $ID(p(\mathbf{A}) \mid\mid q(\mathbf{A}))$, and will employ it in our practical score (Sect. 5.2) for identifying good IPDs.

5 Identifying the optimal discretization

Our overall goal is to find a discretization that balances preserving interactions and detail of the data. We introduce a global objective function for identifying the optimal multivariate discretization—which can be used to compare fairly between *any* discretization—and a practical variant that allows for easier optimization.

We regard the problem of discretization as a model selection problem. Hence, in order to select the best model we need an appropriate model selection criterion. As we explicitly do not want to assume prior distributions, the MDL principle (Rissanen 1978) makes for a natural and well-founded choice.

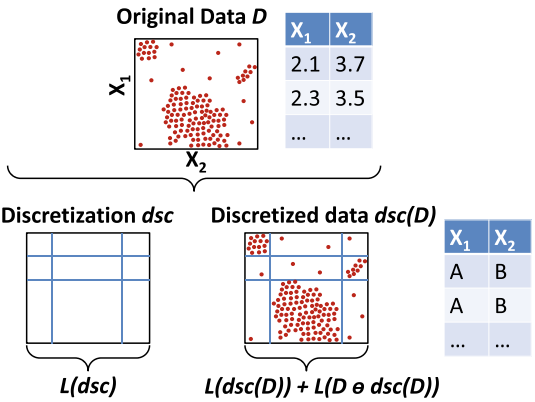

Fig. 2 Example of discretization component costs. $L(dsc)$ is the encoding cost of a discretization dsc of $\mathbf{D}$. In this example, dsc is a discretization of *both* X_1 and X_2, i.e., here dsc is a 2-dimensional grid. $L(dsc(\mathbf{D}))$ is the cost of encoding the discretized data $dsc(\mathbf{D})$. Finally, $L(\mathbf{D} \ominus dsc(\mathbf{D}))$ is the cost for encoding the exact data points within each bin

Loosely speaking, MDL identifies the best model as the model that obtains the best lossless compression of the data. More formally, given a set of models $\mathcal{M}$, the best model $M \in \mathcal{M}$ is identified as the one that minimizes

$$L(\mathbf{D}, M) = L(M) + L(\mathbf{D} \mid M) \tag{1}$$

where $L(M)$ is the length, in bits, of the description of the model M, and $L(\mathbf{D} \mid M)$ is the length, again in bits, of the description of the data $\mathbf{D}$ as encoded by M. That is, MDL helps select a model that yields the best balance between goodness of fit and model complexity. To ensure fair comparison, MDL requires lossless encodings.

In the following, we will define interaction preserving discretization in terms of MDL. First, we will discuss our *ideal objective function* in Sect. 5.1, which allows for fair comparison between any discretization; yet, however, does not lend itself for fast optimization. To facilitate efficient search, in Sect. 5.2 we extend it into a *practical score* that uses *ID* and does allow for efficient search.

5.1 MDL for interaction-preserving discretization

In the context of discretization, let dsc be a discretization of $\mathbf{D}$, and $dsc(\mathbf{D})$ be the discretized data that dsc creates on $\mathbf{D}$ (see Fig. 2). For IPD, we need to encode the following.

5.1.1 Encoding the discretization

Encoding the discretization grid dsc means encoding the number of bins, and their cut points, per dimension X_i. Technically, we first encode the number of bins. Assume that X_i has k_i bins, i.e., $(k_i - 1)$ cut points. To encode the number of bins k_i, we use $L_{\mathbb{N}}$, the MDL-optimal encoding of integers (Rissanen 1983). It is defined for $z \geq 1$, with $L_{\mathbb{N}}(z) = \log^*(z) + \log c_0$, where $\log^*(z) = \log(z) + \log\log(z) + \ldots$, where we only include the positive terms, and c_0 is set to 2.8654 to make it a valid encoding

by ensuring that all probabilities sum to 1. The intuition is that larger integers require more bits.

Next, to encode the locations of the cut points, we note that digitally stored data is recorded at a finite resolution (Kontkanen and Myllymäki 2007), and hence the resolution res_i of a dimension X_i is data-dependent. For example, let us assume that dimension X_i has domain $[0, 1]$. From the data we can observe that it is encoded with, e.g., 2 significant digits, or, at resolution of $res_i = 0.01$. Given the resolution res_i for a dimension X_i, we have that every possible value of X_i belongs to the set $\mathcal{V}_i = \{min_i + z \cdot res_i : z = 0, 1, \ldots, n_i\}$ where $n_i = \frac{max_i - min_i}{res_i}$.

As we have no prior expectation on their location, any set of $(k_i - 1)$ distinct cut points is equally likely—and hence, data-to-model codes are optimal (Vereshchagin and Vitányi 2004). A data-to-model code is an index into a canonically ordered enumeration of all possible data (i.e., values) given the model (the provided information). Here, we know $(k_i - 1)$ cut points have to be selected out of n_i candidates; a choice for which there are $\binom{n_i}{k_i - 1}$ possibilities. Assuming a canonical order, $\log \binom{n_i}{k_i - 1}$ gives the number of bits to identify the actual set of cut points. As such, we have

$$L(dsc) = \sum_{i=1}^{m} L_{\mathbb{N}}(k_i) + \log \binom{n_i}{k_i - 1}$$

for the number of bits required to encode a discretization dsc. Next, we discuss how to encode the discretized data, $dsc(\mathbf{D})$.

5.1.2 Encoding the discretized data

The length of the discretized data $dsc(\mathbf{D})$ in bits, $L(dsc(\mathbf{D}))$, is defined as

$$L(dsc(\mathbf{D})) = \log |\mathcal{C}| + \min_{C \in \mathcal{C}} L(dsc(\mathbf{D}), C).$$

Here, $\mathcal{C}$ is the set of all lossless compressors applicable to discrete data of the form $dsc(\mathbf{D})$. By MDL, we identify the best compressor $C \in \mathcal{C}$ as the one that encodes $dsc(\mathbf{D})$ most succinctly. Encoding the index of C requires $\log |\mathcal{C}|$ bits, and the lossless encoding of $dsc(\mathbf{D})$ by compressor C takes $L(dsc(\mathbf{D}), C)$ bits.

This cost is ideal as it minimizes over all possible compressors. That is, it can detect and reward *any* interaction present in the discretized data. However, this general form is not very practical: we do not have access to all applicable compressors, nor the time to evaluate them all. To use the score, we will hence have to instantiate $\mathcal{C}$.

To this end, any compressor suited for categorical data can be used. Naively, we can even encode dimensions independently using prefix codes (Cover and Thomas 2006). However, as our aim is to find IPDs, we should rather use a compressor that is interaction-aware. That is, one that can detect and reward correlations over dimensions. For instance, we could first serialize the data row per row, and then use GZIP or one of its many variants. This, however, would be very sensitive to the serialization order of the data. Better choices hence include modern itemset-based compressors, such as KRIMP (Vreeken et al. 2011), COMPREX (Akoglu et al. 2012), MTV (Mampaey

et al. 2012), or PACK (Tatti and Vreeken 2008), as these can detect interactions among dimensions independently to the order of the data. Each can be used as plug in for $L(dsc(\mathbf{D}), C)$—in our experiments we will evaluate using a number of applicable compressors.

For the following, let us assume we have chosen a suited compressor, i.e., that we can calculate $L(dsc(\mathbf{D}))$. We will now finalize the score by discussing how to reconstruct the continuous input data $\mathbf{D}$ given the discretized data $dsc(\mathbf{D})$.

5.1.3 Encoding the errors

In order to make our score lossless, a necessary requirement in MDL, we have to formalize how to compute the encoding cost of reaching the continuous-valued entries of $\mathbf{D}$ from the discrete values in $dsc(\mathbf{D})$.

Let us write $L(\mathbf{D} \ominus dsc(\mathbf{D}))$ for the number of bits required to identify the exact data points within their respective multivariate cell. This cost can be factorized per dimension, that is, per univariate bin. Hence, over all data points, we have

$$L(\mathbf{D} \ominus dsc(\mathbf{D})) = \sum_{i=1}^{m} L(X_i \ominus dsc(X_i)).$$

It is more convenient to aggregate this cost per bin. Let $\{B_i^1, \ldots, B_i^{k_i}\}$ be the set of bins induced by dsc on dimension X_i. We write $|B_i^j|$ as the number of values of X_i that B_i^j contains. We then have

$$L(X_i \ominus dsc(X_i)) = \sum_{j=1}^{k_i} |B_i^j| \log\left(\left\lfloor \frac{ub(B_i^j) - lb(B_i^j)}{res_i} \right\rfloor + 1\right)$$

as the cost of reaching the actual values for a dimension X_i given the discretized representation $dsc(X_i)$, where $ub(B_i^j)$ is the upper bound of bin B_i^j, $lb(B_i^j)$ its lower bound. With the resolution res_i of X_i, we have $\left(\left\lfloor \frac{ub(B_i^j) - lb(B_i^j)}{res_i} \right\rfloor + 1\right)$ as the number of possible values in B_i^j.

5.1.4 The ideal score

With the above three elements we can construct our ideal score. It identifies the best interaction-preserving multivariate discretization dsc^* as the discretization that minimizes the following cost function (i.e., our *ideal* score):

$$L(\mathbf{D}, dsc) = \underbrace{L(dsc) + L(dsc(\mathbf{D}))}_{L(M)} + \underbrace{L(\mathbf{D} \ominus dsc(\mathbf{D}))}_{L(\mathbf{D}|M)}. \tag{2}$$

We give the intuition on how it identifies the best discretization. First, assume a discretization dsc that is too detailed, i.e., it splits dimensions into overly many bins. That is, it is difficult (up to impossible) to detect interactions in $dsc(\mathbf{D})$—i.e., $L(dsc(\mathbf{D}))$ will be very high. At the same time, as the bins are small identifying exact values within them is easy—i.e., $L(\mathbf{D} \ominus dsc(\mathbf{D}))$ will be low.

Alternatively, assuming dsc is very coarse, i.e., data points are grouped into too few, or even only 1 bin. $dsc(\mathbf{D})$ will now show interactions even when there are none in $\mathbf{D}$. It will be easy to compress this data, and $L(dsc(\mathbf{D}))$ will be low. However, there are now many more possible values per bin, and hence encoding the exact values of the data costs many more bits—$L(\mathbf{D} \ominus dsc(\mathbf{D}))$ will be very high. The optimal discretization dsc^* is the discretization that is neither too detailed or too coarse: the one that maintains the true interactions of $\mathbf{D}$.

A key benefit of this score is that it allows for fair and unbiased comparison between *any* discretization discovered by *any* discretization method—simply by instantiating it using different compressors, and comparing the total number of bits. We will use it as such in our experiments in Sect. 7.3.

Though ideal for identifying the optimal discretization, the score does not lend itself for fast optimization towards that goal. For example, it does not factor over dimensions, and so we would have to discretize all dimensions concurrently. However, for multivariate data the search space is exponential to both n and m, and hence restrictively large in practice. Moreover, we do not have access to the optimal compressor C^*, and can hence not compute $L(dsc(\mathbf{D}))$ directly. In theory we could approximate C^* by instantiation $\mathcal{C}$ with a collection of compressors, but this could lead to erratic behavior. Most importantly, though, is that we aim for a fast general approach for high quality IPD and hence want to avoid including computationally expensive heuristics in our objective that only reward specific types of interactions, such as Vreeken et al. (2011) and Akoglu et al. (2012).

To this end, in the next section, we will take the ideal score and adapt it into a practical score that is independent of specific compressors, can detect and reward interactions in general, and does allow for efficient optimization.

5.2 A fast optimizable score for IPD

In this section we will discuss our practical score, which maintains key properties of our ideal score, yet does allow for efficient optimization. In short, it discretizes one dimension at a time while considering its interactions with the other dimensions. We achieve this in three ways. First, we formalize a score that is factorized per dimension. This allows us to identify the optimal discretization per dimension. Second, we base on *ID* to evaluate whether interactions are maintained. Third, to avoid problems of insufficient data for meaningful statistical assessment (Lee and Verleysen 2007), we consider data at the level of micro bins instead of the individual objects.

We form micro bins for X_i by partitioning its interval into T_i fine-grained bins (e.g., by clustering). For each X_i, let $\mathbf{M}_i$ be the corresponding set of micro bins. To avoid confusion, we refer to bins B_i^j of dsc_i as *macro* bins.

With T_i micro bins, we have $(T_i - 1)$ cut points. Merging these micro bins into k_i macro bins $B_i^1, \ldots, B_i^{k_i}$ (each B_i^j contains $|B_i^j|$ micro bins) means choosing $(k_i - 1)$ out of $(T_i - 1)$ cut points. As such, a discretization for dimension X_i corresponds to a subset of all cut points, where the empty subset corresponds to merging all micro bins of X_i into just one macro bin, and the full set corresponds to the input micro bins. Given a discretization dsc_i of X_i, we denote $dsc_i(\mathbf{M}_i)$ as the resulting discretized data of X_i, i.e., the resulting set of macro bins.

The building blocks of the score are analogue to the ideal score (Eq. 2), yet now defined per dimension and defined over micro bins. We will discuss its terms in detail below.

5.2.1 Encoding the discretization

The intuition for encoding dsc_i is identical to the ideal score. We have

$$L(dsc_i) = L_{\mathbb{N}}(k_i) + \log \binom{T_i - 1}{k_i - 1} \quad ,$$

where we encode the number of bins, and identify the cut points from $(T_i - 1)$ options.

5.2.2 Encoding the discretized data

Instead of optimizing towards a specific compressor, we will use *ID* to determine how well a discretization dsc_i maintains the interactions of the data. We define the cost of encoding the discretized data $dsc_i(\mathbf{M}_i)$ as

$$L(dsc_i(\mathbf{M}_i)) = L_{bid}(dsc_i(\mathbf{M}_i)) + L_{mh}(dsc_i(\mathbf{M}_i)) \quad ,$$

where $L_{bid}(.)$ is the cost of the discretized data under the independence model, and $L_{mh}(.)$ is a penalty on the multivariate heterogeneity of the discretization.

Encoding the macro bin ids Encoding the discretized data means encoding the macro bin id per micro bin. We do this by assigning optimal prefix codes to the macro bins, the lengths of which we calculate by Shannon entropy. The code length of the id of macro bin B_i^j then is $-\log \frac{|B_i^j|}{T_i}$. Over all macro bins, we have

$$L_{bid}\left(dsc_i\left(\mathbf{M}_i\right)\right) = \sum_{j=1}^{k_i} \left(L_{\mathbb{N}}\left(|B_i^j|\right) - \log \frac{|B_i^j|}{T_i} - |B_i^j| \log \frac{|B_i^j|}{T_i} \right).$$

where the first term encodes the number of micro bins in B_i^j, the second is the cost of the macro bin code in the dictionary, and the third term is the cost of using this code to identify the macro bin per associated micro bin.

Penalizing multivariate heterogeneity The above encoding is lossless, but unaware of interactions. To make it interaction-aware, we include a penalty term based on *ID*. The intuition is to reward regions with similar distributions to be in the same

hypercubes (cubes in the space formed by all dimensions), and vice-versa. That is, two micro bins with different data distributions, as identified by *ID*, should stay separate. If a discretization combines them, we penalize accordingly.

We penalize on the number of *break points* in a macro bin—the indexes of consecutive micro bins within the macro bin for which the interaction distance is large. More formally, let us consider an arbitrary macro bin B_i^j with micro bins $b_i^{j,1}, \ldots, b_i^{j,|B_i^j|}$. For each pair of consecutive micro bins $b_i^{j,w}$ and $b_i^{j,w+1}$ in B_i^j, if their interaction distance is large, we will encode index w to represent a break point between the distributions. We write $I(B_i^j)$ for the set of indices of these break points, with

$$I\left(B_i^j\right) = \Big\{ w \in [1, |B_i^j| - 1] \mid \\ ID\left(p\left(\mathbf{A}\backslash\{X_i\}|b_i^{j,w}\right) \;||\; p\left(\mathbf{A}\backslash\{X_i\}|b_i^{j,w+1}\right)\right) \text{ is large} \Big\}.$$

This allows us to include the cost of encoding $I(B_i^j)$ in $L_{mh}(dsc_i(\mathbf{M}_i))$. (We will discuss how to decide if an interaction distance is large in Sect. 6.)

To ensure we only penalize when interactions are broken, L_{mh} includes only those macro bins for which $I(B_i^j)$ is non-empty. Formally, we define

$$L_{mh}\left(dsc_i\left(\mathbf{M}_i\right)\right) = \sum_{\substack{j=1: \\ |I\left(B_i^j\right)|>0}}^{k_i} L_{\mathbb{N}}\left(|I\left(B_i^j\right)|\right) + |I\left(B_i^j\right)| \log\left(|B_i^j| - 1\right)$$

where we encode the number of break points by $L_{\mathbb{N}}$, and encode $I(B_i^j)$ using optimal prefix codes. Here, this entails identifying the index of each break point out of $(|B_i^j|-1)$ possible pairs, which hence costs $\log(|B_i^j| - 1)$ bits per index. This penalty captures our intuition: the more micro bins with different multivariate distributions in a macro bin, the higher its cost. Combined, L_{bid} and L_{mh} tell us how well a discretization maintains the interactions and detail of the data.

5.2.3 Encoding the errors

With the above we know the discrete data. The final step is to reconstruct the data up to the micro bin ids. As we know the number of micro bins per macro bin from $dsc_i(\mathbf{M}_i)$, here we only have to identify the ids of the micro bins. Using optimal prefix codes, we have

$$L(\mathbf{M}_i \ominus dsc_i(\mathbf{M}_i)) = \sum_{j=1}^{k_i} |B_i^j| \log |B_i^j|.$$

Note that we do not have to reconstruct the original data up till the exact values of X_i for fair model selection. This has two reasons. First, recall that our practical score only considers data up to the resolution of the micro bins: it does not 'see' the data in

higher detail. Second, the cost of encoding the exact values of X_i for a given set of micro bins $\mathbf{M}_i$ is constant over all models. Hence, we can safely ignore it here.

5.2.4 The practical score

We have now completed the definition of our practical score, which aims to identify the best IPD *per* dimension. Formally, we aim at finding the discretization dsc_i^* per dimension X_i that minimizes

$$L(\mathbf{M}_i, dsc_i) = L(dsc_i) + L(dsc_i(\mathbf{M}_i)) + L(\mathbf{M}_i \ominus dsc_i(\mathbf{M}_i)). \tag{3}$$

It is easy to see that its terms are analogue to the ideal score (Eq. 2), and though there exists no formal connection between our two scores, the general intuition is identical: both reward and punish similarly. Intuitively, a good solution under the practical score is also a good solution under the ideal score.

We do note that our practical score is independent of res_i. In addition, it addresses the lack of the optimal compressor, and can be optimized independently per dimension. Furthermore, it can be instantiated by any interaction distance. In this paper, we use *ID* as an instantiation since it yields a good combination of theoretical correctness and foundations, simplicity, and ease of computation.

6 The IPD algorithm

Having introduced the theoretical model of IPD, which consists of our interaction distance *ID* and our MDL-based score, we now detail our algorithmic approach. In order, we will first give two efficient bin merge strategies, then discuss parameter settings, and finally we will analyze the time complexity of our algorithms.

6.1 Algorithms

We will now discuss the IPD algorithm, for which we give the pseudo code as Algorithm 1. First, we pre-process the data to obtain micro bins (Line 3), after which for every pair of consecutive micro bins we use *ID* to calculate their interaction. The most important step in IPD is the bin merge strategy on Line 6. Given T_i micro bins, there are 2^{T_i-1} merge possibilities, which is too many to evaluate exhaustively. For exam-

Algorithm 1: IPD

```
1 for each dimension X_i ∈ A do
2     A_i ← A \ {X_i}
3     Form micro bins {b_i^1, ..., b_i^{T_i}} for X_i
4     for w = 1 to T_i − 1 do
5         id_i^w = ID(p(A_i|b_i^w) || p(A_i|b_i^{w+1}))
6     Macro bins {B_i^1, ..., B_i^{k_i}} ← Merge of {b_i^1, ..., b_i^{T_i}} using {id_i^1, ..., id_i^{T_i−1}}
```

ple, for $T_i = 50$ we already have more than 1 trillion options. To tackle this problem, we prove that the optimal merge of micro bins can be found in polynomial time by dynamic programming. The details are as follows.

6.1.1 Optimal bin merge strategy IPD_{opt}

We show that the search space of bin merges for a dimension X_i is structured, i.e., intermediate results can be re-used to avoid redundant computation. In particular, let $F(c, k)$ be the minimum total encoding cost over all merges of the first c micro bins of X_i ($1 < c \leq T_i$) producing k macro bins ($1 < k \leq c$). For each l with $k - 1 \leq l < c$, consider a merge of the first c micro bins that combines the first l of them into $(k-1)$ macro bins, and combines the remaining $(c-l)$ micro bins into its kth macro bin $B_i^{k,l}$. We arrive at the following theorem.

Theorem 4 *$F(c, k)$ is equal to*

$$
\begin{aligned}
&\min_{k-1 \leq l < c} \Bigg\{ F(l, k-1) \\
&\quad + L_{\mathbb{N}}(k) + \log \binom{c-1}{k-1} - L_{\mathbb{N}}(k-1) - \log \binom{l-1}{k-2} && (4)\\
&\quad + L_{\mathbb{N}}\left(|B_i^{k,l}|\right) - \log \frac{|B_i^{k,l}|}{c} - (k-1) \log \frac{c - |B_i^{k,l}|}{c} && (5)\\
&\quad - |B_i^{k,l}| \log \frac{|B_i^{k,l}|}{c} - \left(c - |B_i^{k,l}|\right) \log \frac{c - |B_i^{k,l}|}{c} && (6)\\
&\quad + L_{\mathbb{N}}\left(|I\left(B_i^{k,l}\right)|\right) + |I\left(B_i^{k,l}\right)| \log\left(|B_i^{k,l}| - 1\right) && (7)\\
&\quad + |B_i^{k,l}| \log |B_i^{k,l}| \Bigg\} && (8)
\end{aligned}
$$

Informally speaking, with regard to our practical score we can consider term (4) to represent $L(dsc_i)$, terms (5) and (6) to correspond with $L_{bid}(dsc_i(\mathbf{M}_i))$, term (7) with $L_{mh}(dsc_i(\mathbf{M}_i))$, and term (8) with $L(\mathbf{M}_i \ominus dsc_i(\mathbf{M}_i))$.

Theorem 4 permits an algorithm based on dynamic programming, since the solution of the first (left-most) c micro bins can be derived from that of the first $l < c$ micro bins. Using dynamic programming, the search for the optimal bin merge is feasible in a polynomial time: For each k such that $1 < k \leq T_i$, we find the optimal bin merge w.r.t. our practical score producing k macro bins on X_i using dynamic programming. When $k = 1$, there is no need to apply the algorithm. Finally, the one yielding the minimum cost across all $k \geq 1$ is selected as the final output. Note that X_i could end up with only one bin. One possible interpretation of this is that X_i contains no significant interaction with other dimensions since, e.g., X_i is a noisy dimension where data values are randomly scattered.

Though dynamic programming is an efficient strategy for traversing an exponential search space, it may require prohibitively long runtime for large data. In addition to this optimal solution, we therefore propose a fast greedy heuristic.

6.1.2 Greedy bin merge strategy IPD_{gr}

Our greedy bin merge is as follows. Starting with the T_i micro bins of X_i, in each step, it searches for two bins whose merge minimizes the practical MDL-based score. If this score is less than the current score, i.e., their merge is beneficial, the greedy algorithm merges these two bins and continues. Otherwise, it terminates. We have two performance bounds of IPD_{gr} on X_i as follows.

Theorem 5 *Asymptotically* IPD_{gr} *is a* 2*-approximation algorithm of* IPD_{opt}.

Theorem 6 *Let* dsc_i^{gr} *be the discretization yielded by* IPD_{gr} *on* X_i. *Further, let* dsc_i^1 *be the discretization that merges all micro bins of* X_i *into one single macro bin. Assuming* $L(\mathbf{M}_i, dsc_i^{gr}) \leq L(\mathbf{M}_i, dsc_i^1)$, *for* $\epsilon \in [0, 1]$ *such that* $(T_i - 1) \cdot \epsilon$ *pairs of consecutive micro bins of* X_i *have low interaction distance, we asymptotically have* IPD_{gr} *as a* $(2 - \epsilon)$*-approximation algorithm of* IPD_{opt}.

As in general ϵ will be larger than zero, the bound in Theorem 6 improves over Theorem 5. For instance, when $\epsilon = 1/3$, IPD_{gr} is a 1.67-approximation algorithm of IPD_{opt}. We observe that the assumption made in Theorem 6 holds for all data sets tested in the experiments. In fact, the results show that IPD_{gr} achieves an approximation factor of about 1.1 of IPD_{opt}, while being up to an order of magnitude faster. Overall, we find that in practice IPD_{gr} strikes a very good balance between time and quality. Still, we note that both variants are both more efficient and produce higher quality discretizations than existing techniques.

6.2 Parameter settings

6.2.1 Setting the number of micro bins

To set T_i, we rely on a recent result by Reshef et al. (2011). They show that to avoid inflated pairwise correlation scores when discretizing data, the number of bins in each dimension must be $\leq n^{1-\delta}$, with n being the number of data points of $\mathbf{D}$ and $\delta \in (0, 1)$. Based on this result, and our own preliminary empirical analysis, we use $T_i = n^{0.5}$ in the remainder of this paper.

6.2.2 Setting a threshold for interaction distances

To use *ID* in our practical score, i.e., to compute $L_{mh}(dsc_i(\mathbf{M}_i))$, we need to be able to decide which interactions distances are 'large'. This is a difficult problem in general, also for other distance functions. The naive way is to use a fixed cutoff threshold. However, preliminary analysis showed this does not work well. That is, in practice there is no global threshold suitable for all dimensions and all data sets.

Instead, we propose a data-driven approach: sort the distances between consecutive micro bins in ascending order and pick a threshold equal to a quantile t_q of the distance values. We thus make the threshold dependent on the distance distribution of each dimension. Preliminary experiments showed that the first tertile is a good choice. Of

course, one can adjust t_q, e.g., by analyzing the distance values, to reflect the level of detail one is willing to keep. One can just set t_q as we do, and let our discretization methods handle the task of merging bins appropriately. Throughout all experiments in this paper we will use the first tertile.

6.3 Complexity analysis

The computational complexity of IPD consists of three parts (1) pre-sorting the data per dimension, (2) computing interaction distances, and (3) bin merging.

Sorting the data costs $O(n \log n)$ per dimension. For each dimension X_i we compute the distances between all pairs of consecutive micro bins, with an individual cost of $O(\frac{mn^2}{T_i^2})$ (cf., Theorem 3). As there are $T_i - 1$ pairs of micro bins, the cost per dimension is $O(\frac{mn^2}{T_i}) = O(mn^{1.5})$ (cf., Sect. 6.2). Bin merging takes $O(T_i^3) = O(n^{1.5})$ for the dynamic programming method, and $O(T_i^2) = O(n)$ for the greedy method. In sum, we see that for both bin merge strategies the total theoretical complexity of IPD is $O(m^2 n^{1.5})$.

It is interesting to compare this result to existing techniques. When we do so, we find that with regard to size MVD (Bay 2001), CPD (Mehta et al. 2005), SD (Fayyad and Irani 1993), and ChiM (Kerber 1992) all have a complexity of $O(n^2)$, while UD potentially scales cubicly to n. With regard to dimensionality, CPD requires $O(m^3)$ time for performing PCA, and potentially exponential time with m for mining itemsets inside the bins. Similarly, in the worst case MVD scales exponentially to m due to its use of contrast set mining. UD, SD, and ChiM scale linearly to m, as they do not analyze interactions with the other $(m - 1)$ dimensions. Overall, in terms of worst case complexity, we find that IPD is at least as efficient as its multivariate competitors. However, the empirical results show that in practice IPD is much faster than both its univariate and multivariate competitors.

7 Experiments

In our experiments, we study the ability of IPD to maintain multivariate interactions. We test its two variants: optimal IPD (IPD_{opt}) using dynamic programming and greedy IPD (IPD_{gr}), which employs our greedy bin merge strategy.

All experiments were conducted on an Intel i5-2520M machine with 8GB RAM. For research purposes we provide the code of IPD on our website.[2]

7.1 Setup

We perform four sets of experiments. We first evaluate, using synthetic data, if our methods preserve known interactions (Sect. 7.2). Next, we evaluate on real-world

[2] http://www.ipd.kit.edu/~nguyenh/ipd/.

Table 1 Characteristics of methods

	Unsupervised	Multivariate	Interaction preserving
UD (Kontkanen and Myllymäki 2007)	✓	–	–
CPD (Mehta et al. 2005)	✓	✓	*
SD (Fayyad and Irani 1993)	–	–	–
IPD	✓	✓	✓

* means partially

Table 2 Characteristics of the real data sets

Data	n	m	Classes
Climate	35601	251	–
Crime	1993	101	–
Gas	13910	128	7
KDD	311029	41	38
Energy	48501	540	2
Mini	130064	50	2
PAMAP	1686000	42	15
PAMAP2	1662216	51	18
Parkinson	5875	18	–
SatImage	6435	36	6

data using three representative use cases for multivariate discretization: pattern-based compression (Sect. 7.3), outlier detection (Sect. 7.4), and classification (Sect. 7.5).

We compare IPD$_{opt}$ and IPD$_{gr}$ against state of the art methods in both supervised and unsupervised discretization. Table 1 summarizes their characteristics. UD (Kontkanen and Myllymäki 2007) performs unsupervised univariate discretization, CPD (Mehta et al. 2005) is for unsupervised multivariate discretization, and SD (Fayyad and Irani 1993) is for supervised discretization. For each method, we optimize parameter settings according to their respective papers. We create the initial micro bins on the basis of equal-frequency, similar to Mehta et al. (2005), and hence, allow for fair comparison. For IPD, we always fix t_q to the first tertile.

We experiment on 10 real data sets. We draw six of them from the UCI Machine Learning Repository, the publicly available PAMAP database [3], and two further data sets on energy and climate data. The Energy data set contains hourly energy consumption indicators (e.g., water, heating, electricity) of different buildings in KIT university campus, recorded from 2006 to 2011. The climate data set contains climate data of an office building in Frankfurt, Germany, collected from 2004 to 2012 (Wagner et al. 2014). Note that SD requires labels and is hence, inapplicable on Climate, Crime, and Parkinson. We summarize the characteristics of these data sets in Table 2.

[3] http://www.pamap.org/demo.html.

Table 3 Preserving interactions on Synthetic Case 1

Ideal	X_1 0.0	X_2 0.0	X_3 0.0	X_4 0.0	X_5 0.0	X_6 0.0	X_7 0.0	$X_8 - X_{100}$ –
IPD_{opt}	−0.04	0.03	−0.04	−0.05	0.02	0.04	−0.04	–
IPD_{gr}	−0.05	0.05	−0.04	−0.05	0.04	0.05	−0.05	–
UD	–	–	–	–	–	–	–	–
SD	–	–	–	–	–	–	−0.04	–
CPD	0.28	0.24	0.16	−0.21	−0.11	−0.24	0.22	+
	0.41	0.37	0.34	0.39	0.33	0.10	0.36	

The ideal outcome: one cut point at 0.0 for dimensions X_1–X_7, no cut points for X_8–X_{100}. (−) means no cut point. (+) means has at least one cut point

7.2 Preserving interactions

To show that our methods are able to preserve known dimension interactions, we first experiment on synthetic data.

7.2.1 Synthetic Case 1

First, we generate data according to the $R + I + S$ parity problem, which is the continuous version of the parity problem. That is, each data set consists of (a) R dimensions uniformly distributed in the range $[-0.5, 0.5]$, (b) another dimension whose value is a random number drawn from $(0, 0.5]$ if an even number of values of the first R dimensions are positive, and drawn from $[-0.5, 0]$ otherwise, and (c) S irrelevant dimensions. For each R, we create a data set of 10,000 points, including 10 % noise. We present representative results with $R = 6$ and $S = 93$, i.e., 100 dimensions in total. For SD, we create the class label as follows: If the $(R + 1)$th dimension is positive, the class is 1, and 0 otherwise.

Please note that, in the ideal solution, each of the dimensions X_1 to X_7 only has one cut point at 0.0 (i.e., two bins) while no irrelevant dimension has a cut point. Table 3 shows that IPD_{opt} and IPD_{gr} produce the results closest to the ideal. UD is univariate and oblivious to dimension interactions, and hence here creates only one bin for X_1 to X_7. SD also yields only one bin for X_1 to X_6. This is because it only considers the interaction of each dimension with the class label, while in this case, interactions among multiple dimensions are required to find proper cut points. CPD in turn introduces spurious cut points for all dimensions, including the irrelevant ones.

To assess robustness of *ID* with regard to high dimensional interactions, we evaluated on data with $R = 60$ and $S = 300$. The result are consistent with those above.

7.2.2 Synthetic Case 2

Next, we generate data according to multivariate histograms. Each data set created has $(R + 2R + 3R + 20R)$ dimensions in the range $[-5.0, 5.0]$. Each of the first R dimensions (R-dimensional interaction) has one cut point at 0.0. Each of the next $2R$

Table 4 Preserving interactions on Synthetic Case 2

	IPD_{opt}	IPD_{gr}	UD	CPD	SD
5-dimensional interaction	✓	✓	–	–	n/a
10-dimensional interaction	✓	✓	–	–	n/a
15-dimensional interaction	✓	✓	–	–	n/a
100 irrelevant dimensions	✓	✓	✓	–	n/a

"✓" means the respective method discovers the correct discretization over all dimensions of the given group, and "–" if otherwise. The ideal outcome: "✓" in all groups. SD is not applicable as the task is unsupervised

dimensions ($2R$-dimensional interaction) has three cut points at -3.0, 0.0, 3.0. Each of the next $3R$ dimensions ($3R$-dimensional interaction) has five cut points at -3.0, -1.0, 0.0, 1.0, 3.0. The remaining $20R$ dimensions are irrelevant. For each group of relevant dimensions with n bins per dimension, we pick n multivariate cells that do not overlap in any univariate bin. For instance, assuming that $R = 3$, cells $(0, 1, 0)$ and $(1, 0, 1)$ of the first group do not have a common univariate bin. For each cell picked, we assign a multivariate Gaussian distribution with a dimensionally matching mean vector and covariance matrix. To create a new data point o, from each relevant group we pick a cell (with equal probability) and sample the values of o in the respective dimensions accordingly. In the irrelevant dimensions, the values of o are sampled uniformly randomly from $[-5.0, 5.0]$. To increase complexity we add 10 % random data points to the unselected cells. Using our procedure, we ensure each data set to follow a known histogram, i.e., known ground-truth cut points.

For a given data set, a discretization method is considered to produce a correct result for a group iff (a) it produces the correct number of cut points in all member dimensions, and (b) if the group contains relevant dimensions, each cut point is within a distance $\delta = 0.5$ to the correct cut point. Table 4 shows the results on a data set with $R = 5$, and containing 10,000 points. For brevity, we only show if methods correctly identify cut points for different groups. Only IPD_{opt} and IPD_{gr} produce the correct discretizations for all groups. This implies that IPD and hence *ID* are robust w.r.t. high dimensional interactions.

Overall, the experiments on synthetic data show that our methods successfully identify and preserve synthetic interactions among dimensions.

7.3 Compression

Next, we examine whether IPD preserves interactions in real-world data. To this end, we use our ideal score (Sect. 5.1) to fairly compare between discretizations. The resolutions per dimension needed in $L(dsc)$ and $L(\mathbf{D} \ominus dsc(\mathbf{D}))$ are determined from the data following Kontkanen and Myllymäki (2007).

We instantiate the score with three different compressors. First, we use GZIP, a general purpose compressor. To apply it, we serialize the data per row, per column in the original order of the data. Further, we use KRIMP (Vreeken et al. 2011) and COMPREX (Akoglu et al. 2012), two pattern-based methods that compress data using itemsets—which allows these methods to detect and reward multivariate interactions.

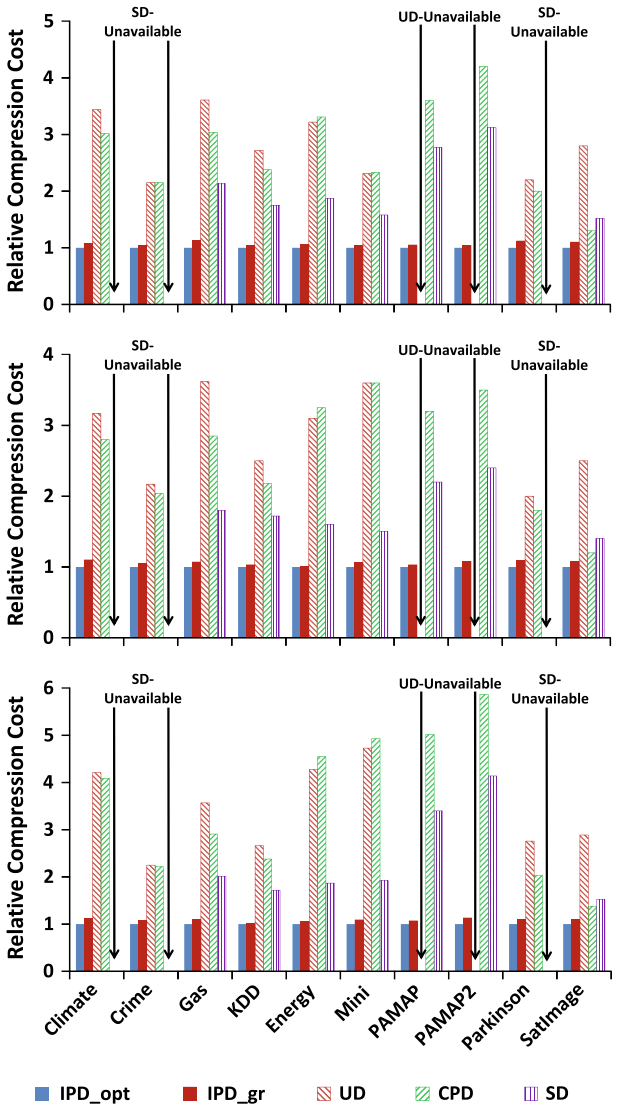

Fig. 3 [Lower is better] Relative total compression costs for all data sets, using GZIP (*top*), KRIMP (*middle*), and COMPREX (*bottom*) as compressor. The compression costs of IPD_{opt} are the bases. SD is not applicable on unlabeled data. UD did not finish within 6 days on the PAMAP data (Color figure online)

In addition, COMPREX can exploit correlations by grouping and encoding highly interacting attributes together. For GZIP and COMPREX we use default parameters, for KRIMP we use a minimal support of 10 %. Overall, for each compressor, the total encoding cost $L(\mathbf{D}, dsc)$ will be low only if the interactions are preserved well.

We present the results in Fig. 3. The plots show the relative compression rates, with IPD_{opt} as base, per dataset, for each of the considered discretization methods,

Table 5 [Lower is better] The total compression costs in bits of IPD_{opt} and IPD_{gr}, $L(\mathbf{D}, dsc)$, using different compressors

Data	GZIP		KRIMP		COMPREX	
	IPD_{opt}	IPD_{gr}	IPD_{opt}	IPD_{gr}	IPD_{opt}	IPD_{gr}
Climate	19413*k*	21018*k*	14092*k*	15501*k*	11247*k*	12597*k*
Crime	754*k*	786*k*	486*k*	511*k*	282*k*	305*k*
Gas	3867*k*	4358*k*	2929*k*	3134*k*	2753*k*	3028*k*
KDD	3178*k*	3178*k*	2838*k*	2923*k*	2724*k*	2751*k*
Energy	28367*k*	30197*k*	22743*k*	22971*k*	13246*k*	13908*k*
Mini	56033*k*	58360*k*	19348*k*	20509*k*	9757*k*	10635*k*
PAMAP	113062*k*	118373*k*	92215*k*	94982*k*	80187*k*	85800*k*
PAMAP2	132673*k*	137874*k*	105967*k*	114444*k*	92146*k*	104125*k*
Parkinson	294*k*	328*k*	210*k*	228*k*	169*k*	186*k*
SatImage	688*k*	754*k*	491*k*	530*k*	420*k*	462*k*

using respectively GZIP, KRIMP, and COMPREX as compressor. The absolute total compression costs for IPD_{opt} and IPD_{gr} are reported in Table 5.

From the figures, we see that IPD_{opt} yields the best results across all data sets and all three compressors. The performance of IPD_{gr} is very close to that of IPD_{opt}. Further, our methods provide about 100 % saving in bits compared to SD, and even over 200 % compared to UD and CPD. This implies that our methods preserve dimension interactions well, aiding interaction-aware methods like KRIMP and COMPREX to detect patterns over multiple attributes, leading to lower compression costs.

Concerning the competition, UD did not finish within 6 days for the PAMAP data sets. Although multivariate in nature, CPD did not obtain very good scores, which indicates it either does not maintain all strong interactions, or that spurious interactions are introduced. Overall, the results show that IPD_{opt} and IPD_{gr} best preserve complex patterns hidden in different real-world data sets.

7.4 Outlier detection

The previous experiments showed our methods preserve interactions. Next, we investigate how well outliers can still be identified in the discretized data. If a discretization is too detailed, all values will be unique, and hence all records are 'outliers'. If the discretization is too coarse, no outliers will be detectable.

To detect outliers we use COMPREX with the different discretization methods as preprocessing step. As argued by Akoglu et al. (2012), patterns that compress the majority of the data well define its norm, and hence, data points that cannot be compressed well can be safely regarded as an outlier. To evaluate performance we use labeled data: one class as the 'normal' objects and another as the 'outliers'. The evaluation metric is the average precision, computed as the area under the precision-recall curve. It is the average of the precision values obtained across recall levels. As standard baseline, we run LOF (Breunig et al. 2000) on the original data. Climate, Crime, and Parkinson are unlabeled and hence not applicable in this experiment.

Table 6 [Higher is better] Average precision (area under precision-recall curve) for outlier mining.

LOF ran on the original continuous data. Highest values are in bold. (*) means the result is unavailable due to excessive runtime

Data	IPD_{opt}	IPD_{gr}	UD	CPD	SD	LOF
Gas	**0.74**	0.72	0.36	0.42	0.48	0.64
KDD	**0.54**	0.53	0.14	0.19	0.03	0.44
Energy	**0.70**	0.69	0.33	0.45	0.45	0.36
Mini	**0.79**	**0.79**	0.30	0.53	0.51	0.60
PAMAP	**0.82**	0.79	*	0.38	0.24	0.54
PAMAP2	**0.84**	0.82	*	0.41	0.27	0.53
SatImage	**0.41**	**0.41**	0.21	0.28	0.15	0.33
Average	**0.69**	0.68	0.27	0.38	0.30	0.49

We present the results in Table 6. Both IPD_{opt} and IPD_{gr} obtain very high average precision, outperforming the competition with a broad margin. In fact, a Friedman test (Demsar 2006) at $\alpha = 0.05$ shows that the observed differences are significant. A Nemenyi test in the post-hoc analysis learns us that: (a) IPD_{opt} significantly outperforms UD, CPD, and SD, and (b) IPD_{gr} significantly outperforms UD and SD. Using a Wilcoxon signed rank test with $\alpha = 0.05$ to compare IPD_{gr} and CPD, we find IPD_{gr} to be significantly better.

Interestingly, IPD_{opt} and IPD_{gr} beat LOF with a wide margin—significant under a Wilcoxon signed rank test—despite that discretized data contains inherently less information. By weeding out irrelevant associations, IPD provides COMPREX the chance to outperform LOF.

We are aware that outlier detection methods exist that may outperform LOF. However, our goal here is not to push the envelope in outlier detection, but to compare the quality of the discovered discretizations. Moreover, LOF is often used as baseline.

7.5 Classification

Next, we evaluate the methods in the context of a supervised task: classification. To evaluate performance, we train Random Forests (Breiman 2001) on the discretized data, and consider accuracy as the performance metric. In addition, as a baseline, we also report its performance on the continuous data (RF). We use the implementation in the WEKA toolkit with default parameters. All results are obtained by performing tenfold cross validation. For the unsupervised methods IPD_{opt}, IPD_{gr}, UD, and CPD, we do not show the class labels during discretization. As above, the unlabeled data sets are not applicable. For PAMAP, RF did not finish due to memory overflows.

We present the results in Table 7. We report both mean and standard deviation over the cross-validation folds. Considering the results, we see that the supervised methods SD and plain RF obtain much higher accuracies than UD and CPD. Interestingly, and somewhat surprisingly, both IPD_{opt} and IPD_{gr} consistently outperform SD, and perform as least as good as plain RF—even though they were unaware of the class labels. A possible explanation is that RF and SD only maintain interactions between individual dimensions and the class label, and by making decisions locally may misalign bins of interacting dimensions. Our methods, however, are able to detect and maintain the

Table 7 [Higher is better] Classification accuracy

Data	IPD_{opt}	IPD_{gr}	UD	CPD	SD	RF
Gas	**0.99 ± 0.00**	**0.99 ± 0.00**	0.56 ± 0.01	0.71 ± .01	0.98 ± 0.00	**0.99 ± 0.00**
KDD	**0.98 ± 0.00**	**0.98 ± 0.00**	0.58 ± 0.00	0.70 ± 0.00	**0.98 ± 0.00**	**0.98 ± 0.00**
Energy	**0.97 ± 0.01**	0.96 ± 0.01	0.48 ± 0.02	0.68 ± 0.02	0.94 ± 0.01	0.93 ± 0.01
Mini	**0.92 ± 0.00**	0.91 ± 0.00	0.75 ± 0.00	0.72 ± 0.00	0.89 ± 0.00	0.91 ± 0.00
PAMAP	**0.92 ± 0.01**	0.90 ± 0.01	*	0.71 ± 0.01	0.87 ± 0.01	–
PAMAP2	**0.98 ± 0.01**	**0.98 ± 0.01**	*	0.66 ± 0.00	0.86 ± 0.01	–
SatImage	**0.89 ± 0.01**	0.87 ± 0.01	0.82 ± 0.01	0.81 ± 0.01	0.86 ± 0.01	**0.89 ± 0.01**
Average	**0.95**	0.94	0.64	0.71	0.91	0.94

RF ran on the original continuous data. Highest values are in bold. (*) means the result is unavailable due to excessive runtime. (–) means the result is unavailable due to memory overflow

multivariate structure of the data, which if it correlates with the class label, will aid classification.

By applying a Friedman test at $\alpha = 0.05$, we find the observed differences between the discretization methods to be significant. A Nemenyi test in the post-hoc analysis shows IPD_{opt} and IPD_{gr} perform significantly better than UD and CPD. A Wilcoxon signed rank test between IPD_{opt} and SD, shows IPD_{opt} to be significantly better. Repeating this test between IPD_{gr} and SD, we find IPD_{gr} to be significantly better. The difference between IPD_{opt}, resp. IPD_{gr}, and RF is not significant.

Note that we are aware of other modern classifiers (e.g., SVMs), which may outperform RF. However, for us, state of the art classification is not the goal, but simply a means for evaluating how well discretization techniques maintain interactions.

7.6 Runtime

Last, we evaluate runtime. In Fig. 4, we show the relative runtimes of all methods on all data sets considered. We pick the runtimes of IPD_{opt} as the bases. The wall-clock runtimes of IPD_{opt} and IPD_{gr} are in Table 8. The results show that in practice, both our methods are faster than the competition, with IPD_{gr} by far the fastest method overall. UD did not finish within 6 days on the PAMAP data.

8 Discussion

The experiments show that IPD provides very high quality discrete data, maintaining the key structures and interactions of the original continuous data. We found that it consistently outperforms its competitors in maintaining interactions and patterns, allowing for the identification of outliers, as well as in classification accuracy. Moreover, the runtime experiment shows that it is faster than both state of the art univariate and multivariate competitors. The improved performance of IPD compared to existing techniques can be traced back to its three main components, (a) our interaction distance *ID*, (b) our MDL-based balancing of preserving dimension

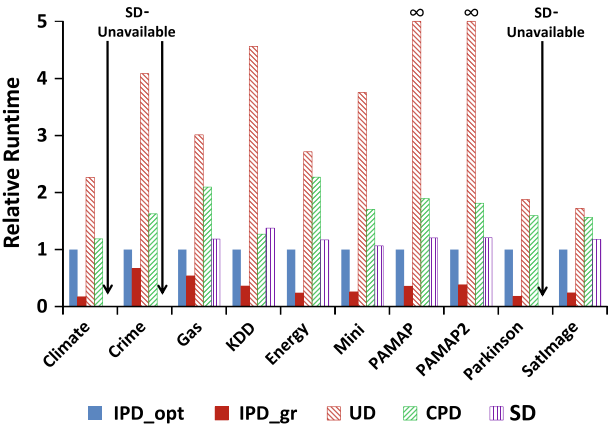

Fig. 4 [Lower is better] Relative runtimes of all methods compared to IPD_{opt}. SD is only applicable on labeled data (Color figure online)

Table 8 [Lower is better] The single-threaded, wall-clock runtimes in seconds of IPD_{opt} and IPD_{gr}

Data	n	m	IPD_{opt}	IPD_{gr}
Climate	35601	251	61242	10708
Crime	1993	101	47	32
Gas	13910	128	1267	682
KDD	311029	41	10200	3674
Energy	48501	540	107233	25532
Mini	130064	50	15208	3923
PAMAP	1686000	42	217835	77798
PAMAP2	1662216	51	284778	109530
Parkinson	5875	18	32	6
SatImage	6435	36	77	18

interactions and the information within dimensions, and (c) our efficient bin merge strategies. In sum, IPD provides a powerful approach for unsupervised multivariate discretization.

This is not to say that the problem of interaction-preserving multivariate discretization is now solved. IPD involves some design choices and there is room for alternative solutions and improvements, which are beyond the scope of this article. For instance, we form micro bins on the basis of equal-frequency. It is interesting to see whether more advanced techniques for forming micro bins, such as UD, may lead to better overall interaction preservation.

MDL here guides us to very good discretizations. However, constructing an encoding involves choices that determine which structure is rewarded. We formalized a general MDL framework for IPD; if one is willing to make explicit assumptions on the distribution of the data or structure of the ideal discretization, other encodings or model selection techniques can be considered. Also, for a more specialized solution, one could optimize towards one specific compressor.

We do note that our MDL framework can be instantiated by any interaction distance. In this paper, we use *ID* as an instantiation. Depending on the task at hand, other interaction distances may be preferred, such as the Kullback-Leibler divergence or the Jensen-Shannon divergence (Cover and Thomas 2006). As long as one is able to find a reliable way to compute such distances, these can be used within our framework. In our context, we find that *ID* yields a good combination of theoretical correctness and foundations, simplicity, and ease of computation.

In this paper we focused on discretization quality. As future work, we consider looking into highly scalable approximations to IPD. Key ideas include that *ID* can be sped up by considering not all dimensions, but only those within the subspace most strongly interacting with the current dimension. That is, we can factor the full-space into smaller subspaces that can then be considered independently. This reduces the complexity for *ID*, and allows for efficient implementation by parallelization. Additionally, as IPD discretizes dimensions independently of the others this can be trivially parallelized, as can the computation of *ID* between consecutive micro bins.

9 Conclusion

In this paper, we proposed IPD, an information-theoretic method for unsupervised discretization of multivariate data, that specifically focuses on the preservation of interactions. That is, for each dimension we consider the distribution of the data over all others. In particular, we only merge consecutive regions if their multivariate distributions are statistically similar and the merge reduces the MDL encoding cost. Empirical evaluation on both synthetic and real-world data sets shows that IPD obtains high quality discrete data that maintains the key interactions of the original, while outperforming existing methods in a range of knowledge discovery tasks.

IPD and *ID* have high potential impact in any setting where discretization is required, be it explicitly such as for methods that only consider discrete data, or implicitly wherever cut points need to be decided. As examples, we plan to investigate the application and embedding of our methods for pattern mining, subgroup discovery, and selectivity estimation. Moreover, we are investigating highly scalable approximations that would allow considering very large databases.

Acknowledgments We thank the anomymous reviewers for their insightful comments. Hoang-Vu Nguyen is supported by the German Research Foundation (DFG) within GRK 1194. Emmanuel Müller is supported by the YIG program of KIT as part of the German Excellence Initiative. Jilles Vreeken is supported by the Cluster of Excellence "Multimodal Computing and Interaction" within the Excellence Initiative of the German Federal Government. Emmanuel Müller and Jilles Vreeken are supported by Post-Doctoral Fellowships of the Research Foundation—Flanders (FWO).

10 Appendix

10.1 Proof of Theorem 2

Proof (Theorem 2) Let $H(\mathbf{A}) = P(\mathbf{A}) - R(\mathbf{A})$ and $G(\mathbf{A}) = R(\mathbf{A}) - Q(\mathbf{A})$. The inequality becomes

$$\sqrt{\int_{\Omega} H^2(\mathbf{a})d\mathbf{a}} + \sqrt{\int_{\Omega} G^2(\mathbf{a})d\mathbf{a}} \geq \sqrt{\int_{\Omega} (H(\mathbf{a}) + G(\mathbf{a}))^2 \, d\mathbf{a}}, \tag{9}$$

which in turn is equivalent to

$$\sqrt{\int_{\Omega} H^2(\mathbf{a})d\mathbf{a} \cdot \int_{\Omega} G^2(\mathbf{a})d\mathbf{a}} \geq \int_{\Omega} H(\mathbf{a})G(\mathbf{a})d\mathbf{a}, \tag{10}$$

which is also known as Hölder's inequality. □

10.2 Proof of Theorem 3

Proof (Theorem 3) Let $ind(\alpha)$ be an indicator function with value 1 if α is true and 0 otherwise. It holds

$$P(\mathbf{a}) = \int_{min_1}^{max_1} \dots \int_{min_m}^{max_m} ind(x_1 \leq a_1) \cdots ind(x_m \leq a_m) p(x_1, \dots, x_m) dx_1 \cdots dx_m \tag{11}$$

Using empirical data, we hence have

$$P(\mathbf{a}) = \frac{1}{k} \sum_{j=1}^{k} \prod_{i=1}^{m} ind(R_j^i \leq a_i), \quad \text{and} \quad Q(\mathbf{a}) = \frac{1}{l} \sum_{j=1}^{l} \prod_{i=1}^{m} ind(S_j^i \leq a_i),$$

and therefore $[ID(p(\mathbf{A}) \mid\mid q(\mathbf{A}))]^2$ equals to

$$\int_{min_1}^{max_1} \dots \int_{min_m}^{max_m} \left(\frac{1}{k} \sum_{j=1}^{k} \prod_{i=1}^{m} ind(R_j^i \leq a_i) - \frac{1}{l} \sum_{j=1}^{l} \prod_{i=1}^{m} ind(S_j^i \leq a_i) \right)^2 da_1 \cdots da_m \tag{12}$$

Expanding the above term and bringing the integrals inside the sums, we have

$$\begin{aligned} &\frac{1}{k^2} \sum_{j_1=1}^{k} \sum_{j_2=1}^{k} \prod_{i=1}^{m} \int_{min_i}^{max_i} ind\left(\max\left(R_{j_1}^i, R_{j_2}^i\right) \leq a_i\right) da_i \\ &\quad - \frac{2}{kl} \sum_{j_1=1}^{k} \sum_{j_2=1}^{l} \prod_{i=1}^{m} \int_{min_i}^{max_i} ind\left(\max\left(R_{j_1}^i, S_{j_2}^i\right) \leq a_i\right) da_i \\ &\quad + \frac{1}{l^2} \sum_{j_1=1}^{l} \sum_{j_2=1}^{l} \prod_{i=1}^{m} \int_{min_i}^{max_i} ind\left(\max\left(S_{j_1}^i, S_{j_2}^i\right) \leq a_i\right) da_i, \end{aligned} \tag{13}$$

by which we arrive at the final result. □

10.3 Proof of Theorem 5

Proof (Theorem 5) Consider a discretization dsc_i on dimension X_i with k_i macro bins $\{B_i^1, \ldots, B_i^{k_i}\}$. We have

$$L(\mathbf{M}_i, dsc_i) \geq L_{bid}(dsc_i(\mathbf{M}_i)) + L(\mathbf{M}_i \ominus dsc_i(\mathbf{M}_i)) \tag{14}$$

$$\geq \left(\sum_{j=1}^{k_i} L_{\mathbb{N}}\left(|B_i^j|\right) + \left(|B_i^j| + 1\right) \log \frac{T_i}{|B_i^j|} \right) + \sum_{j=1}^{k_i} |B_i^j| \log |B_i^j| \tag{15}$$

$$\geq (T_i + k_i) \log T_i - \sum_{j=1}^{k_i} \log |B_i^j| \tag{16}$$

$$\geq T_i \log T_i. \tag{17}$$

Let $dsc_i^{T_i}$ be the discretization that puts each micro bin into a separate macro bin. We have

$$L\left(\mathbf{M}_i, dsc_i^{T_i}\right) = L_{\mathbb{N}}(T_i) + T_i \log c_0 + 2T_i \log T_i. \tag{18}$$

Let dsc_i^{opt} and dsc_i^{gr} be the discretization yielded by IPD_{opt} and IPD_{gr}, respectively.

Let dsc_i be a discretization that merges two micro bins with a low interaction distance into the same macro bin and places each of the other micro bins into a separate macro bin. It holds that

$$L(\mathbf{M}_i, dsc_i) = L_{\mathbb{N}}(T_i - 1) + \log(T_i - 1) + (T_i - 1) \log c_0 + 2T_i \log T_i - \log T_i. \tag{19}$$

Thus, $L(\mathbf{M}_i, dsc_i) < L(\mathbf{M}_i, dsc_i^{T_i})$, i.e., merging two consecutive micro bins with a low interaction distance in the first place will yield an encoding cost lower than that of $dsc_i^{T_i}$. Thus, IPD_{gr} will proceed after this step. Hence, $L(\mathbf{M}_i, dsc_i^{gr}) \leq L(\mathbf{M}_i, dsc_i)$. We have $\dfrac{L(\mathbf{M}_i, dsc_i^{gr})}{L(\mathbf{M}_i, dsc_i^{opt})} \leq \dfrac{L(\mathbf{M}_i, dsc_i)}{T_i \log T_i}$. This leads to

$$\frac{L\left(\mathbf{M}_i, dsc_i^{gr}\right)}{L\left(\mathbf{M}_i, dsc_i^{opt}\right)} \leq \frac{L_{\mathbb{N}}(T_i - 1) + \log(T_i - 1) + (T_i - 1) \log c_0 + 2T_i \log T_i - \log T_i}{T_i \log T_i}. \tag{20}$$

Let RHS be the right hand side of (20). It holds that $\lim_{T_i \to \infty} RHS = 2$ as $\lim_{T_i \to \infty} \frac{L_{\mathbb{N}}(T_i - 1)}{T_i \log T_i} = 0$ (Grünwald 2007). In other words, as $T_i \to \infty$, $\frac{L(\mathbf{M}_i, dsc_i^{gr})}{L(\mathbf{M}_i, dsc_i^{opt})} \leq 2$. Therefore, asymptotically IPD_{gr} is a 2-approximation algorithm of IPD_{opt}. □

10.4 Proof of Theorem 6

Proof (Theorem 6) We assume that there are $(T_i - 1)\epsilon$ pairs of consecutive micro bins of X_i that have low interaction distance ($0 \leq \epsilon \leq 1$), i.e., $(T_i - 1)(1 - \epsilon)$ pairs have a large interaction distance. We have

$$L(\mathbf{M}_i, dsc_i^1) = \log c_0 + L_{\mathbb{N}}(T_i) + L_{\mathbb{N}}\left((T_i - 1)(1 - \epsilon)\right) + (T_i - 1)(1 - \epsilon)\log(T_i - 1) + T_i \log T_i. \tag{21}$$

This means $\frac{L(\mathbf{M}_i, dsc_i^{gr})}{L(\mathbf{M}_i, dsc_i^{opt})} \leq$

$$\frac{\log c_0 + L_{\mathbb{N}}(T_i) + L_{\mathbb{N}}\left((T_i - 1)(1 - \epsilon)\right) + (T_i - 1)(1 - \epsilon)\log(T_i - 1) + T_i \log T_i}{T_i \log T_i}. \tag{22}$$

Let *RHS* be the right hand side of (22). Note that $\lim_{T_i \to \infty} RHS = 2 - \epsilon$. In other words, as $T_i \to \infty$, $\frac{L(\mathbf{M}_i, dsc_i^{gr})}{L(\mathbf{M}_i, dsc_i^{opt})} \leq 2 - \epsilon$. □

References

Aggarwal CC, Yu PS (2001) Outlier detection for high dimensional data. In: SIGMOD Conference, p 37–46.

Agrawal R, Gehrke J, Gunopulos D, Raghavan P (1998) Automatic subspace clustering of high dimensional data for data mining applications. In: SIGMOD Conference, p 94–105.

Akoglu L, Tong H, Vreeken J, Faloutsos C (2012) Fast and reliable anomaly detection in categorical data. In: CIKM, p 415–424.

Allen JF (1983) Maintaining knowledge about temporal intervals. Commun ACM 26(11):832–843

Allen JF, Ferguson G (1994) Actions and events in interval temporal logic. J Log Comput 4(5):531–579

Aue A, Hörmann S, Horváth L, Reimherr M (2009) Break detection in the covariance structure of multivariate time series models. Ann Stat 37(6B):4046–4087

Bay SD (2001) Multivariate discretization for set mining. Knowl Inf Syst 3(4):491–512

Bay SD, Pazzani MJ (1999) Detecting change in categorical data: Mining contrast sets. In: KDD, p 302–306.

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

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

Breunig MM, Kriegel HP, Raymond T Ng JS (2000) LOF: identifying density-based local outliers. In: SIGMOD Conference, p 93–104.

Bu S, Lakshmanan LVS, Ng RT (2005) Mdl summarization with holes. In: VLDB, p 433–444.

Cheng CH, Fu AWC, Zhang Y (1999) Entropy-based subspace clustering for mining numerical data. In: KDD, p 84–93.

Cover TM, Thomas JA (2006) Elements of information theory. Wiley, New York

Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Fayyad UM, Irani KB (1993) Multi-interval discretization of continuous-valued attributes for classification learning. In: IJCAI, p 1022–1029.

Ferrandiz S, Boullé M (2005) Multivariate discretization by recursive supervised bipartition of graph. In: MLDM, p 253–264.

Grosskreutz H, Rüping S (2009) On subgroup discovery in numerical domains. Data Min Knowl Discov 19(2):210–226

Grünwald PD (2007) The minimum description length principle. MIT Press, Cambridge

Gunopulos D, Kollios G, Tsotras VJ, Domeniconi C (2000) Approximating multi-dimensional aggregate range queries over real attributes. In: SIGMOD Conference, p 463–474.
Han J, Cheng H, Xin D, Yan X (2007) Frequent pattern mining: current status and future directions. Data Min Knowl Disc 15:55–86
Kang Y, Wang S, Liu X, Lai H, Wang H, Miao B (2006) An ICA-based multivariate discretization algorithm. In: KSEM, p 556–562.
Kerber R (1992) ChiMerge: discretization of numeric attributes. In: AAAI, p 123–128.
Kontkanen P, Myllymäki P (2007) MDL histogram density estimation. J Mach Learn Res 2:219–226
Lakshmanan LVS, Ng RT, Wang CX, Zhou X, Johnson T (2002) The generalized MDL approach for summarization. In: VLDB, p 766–777.
Lee J, Verleysen M (2007) Nonlinear dimensionality reduction. Springer, New York
Lemmerich F, Becker M, Puppe F (2013) Difference-based estimates for generalization-aware subgroup discovery. In: ECML/PKDD (3), p 288–303.
Liu R, Yang L (2008) Kernel estimation of multivariate cumulative distribution function. J Nonparametr Stat 20(8):661–677
Mampaey M, Vreeken J, Tatti N (2012) Summarizing data succinctly with the most informative itemsets. ACM TKDD 6:1–44
Mehta S, Parthasarathy S, Yang H (2005) Toward unsupervised correlation preserving discretization. IEEE Trans Knowl Data Eng 17(9):1174–1185
Moise G, Sander J (2008) Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering. In: KDD, p 533–541.
Müller E, Assent I, Krieger R, Günnemann S, Seidl T (2009) DensEst: density estimation for data mining in high dimensional spaces. In: SDM, p 173–184.
Nguyen HV, Müller E, Vreeken J, Keller F, Böhm K (2013) CMI: an information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In: SDM, p 198–206.
Peleg S, Werman M, Rom H (1989) A unified approach to the change of resolution: space and gray-level. IEEE Trans Pattern Anal Mach Intell 11(7):739–742
Philip Preuß HD Ruprecht Puchstein (2013) Detection of multiple structural breaks in multivariate time series. arXiv:1309.1309v1.
Rao M, Seth S, Xu JW, Chen Y, Tagare H, Príncipe JC (2011) A test of independence based on a generalized correlation function. Signal Process 91(1):15–27
Reshef DN, Reshef YA, Finucane HK, Grossman SR, McVean G, Turnbaugh PJ, Lander ES, Mitzenmacher M, Sabeti PC (2011) Detecting novel associations in large data sets. Science 334(6062):1518–1524
Rissanen J (1978) Modeling by shortest data description. Automatica 14(1):465–471
Rissanen J (1983) Modeling by shortest data description. Ann Stat 11(2):416–431
Scargle JD, Norris JP, Jackson B, Chiang J (2013) Studies in astronomical time series analysis. vi. Bayesian block representations. Astrophys J 764(2)
Seth S, Rao M, Park I, Príncipe JC (2011) A unified framework for quadratic measures of independence. IEEE Trans Signal Process 59(8):3624–3635
Silverman BW (1986) Density estimation for statistics and data analysis. Chapman & Hall/CRC, London
Tatti N, Vreeken J (2008) Finding good itemsets by packing data. In: ICDM, p 588–597.
Tzoumas K, Deshpande A, Jensen CS (2011) Lightweight graphical models for selectivity estimation without independence assumptions. PVLDB 4(11):852–863
Vereshchagin NK, Vitányi PMB (2004) Kolmogorov's structure functions and model selection. IEEE Trans Inf Theory 50(12):3265–3290
Vreeken J, van Leeuwen M, Siebes A (2011) Krimp: mining itemsets that compress. Data Min Knowl Disc 23(1):169–214
Wagner A, Lützkendorf T, Voss K, Spars G, Maas A, Herkel S (2014) Performance analysis of commercial buildings: results and experiences from the german demonstration program 'energy optimized building (EnOB)'. Energy Build 68:634–638
Yang X, Procopiuc CM, Srivastava D (2009) Summarizing relational databases. PVLDB 2(1):634–645
Yang X, Procopiuc CM, Srivastava D (2011) Summary graphs for relational database schemas. PVLDB 4(11):899–910