Toward Unsupervised Correlation Preserving Discretization

Sameep Mehta, Srinivasan Parthasarathy, *Member*, *IEEE*, and Hui Yang

Abstract—Discretization is a crucial preprocessing technique used for a variety of data warehousing and mining tasks. In this paper, we present a novel PCA-based unsupervised algorithm for the discretization of continuous attributes in multivariate data sets. The algorithm leverages the underlying correlation structure in the data set to obtain the discrete intervals and ensures that the inherent correlations are preserved. Previous efforts on this problem are largely supervised and consider only piecewise correlation among attributes. We consider the correlation among continuous attributes and, at the same time, also take into account the interactions between continuous and categorical attributes. Our approach also extends easily to data sets containing missing values. We demonstrate the efficacy of the approach on real data sets and as a preprocessing step for both classification and frequent itemset mining tasks. We show that the intervals are meaningful and can uncover hidden patterns in data. We also show that large compression factors can be obtained on the discretized data sets. The approach is task independent, i.e., the same discretized data set can be used for different data mining tasks. Thus, the data sets can be discretized, compressed, and stored once and can be used again and again.

Index Terms—Data preprocessing, principal component analysis, data mining/summarization, missing data, data compression.

1 INTRODUCTION

DISCRETIZATION, a widely used data preprocessing primitive, has typically been thought of as the partitioning of the range of a continuous (base) attribute into intervals in order to highlight the behavior of a related discrete (goal) attribute. It has been frequently used for classification in the decision tree context, as well as for summarization in situations where one needs to transform a continuous attribute into a discrete one with minimum "loss of information." It has recently been used as a preprocessing step for frequent itemset discovery applications [22], as well as a compression/summarization tool in data warehousing environments.

Typically, discretization methods have focused on discretizing a continuous attribute based on a *single* goal attribute. Recently, several researchers [2], [12] have pointed out that such methods are limited in a multivariate context, which can result in nonoptimal solutions. To gain an intuitive insight as to why this is so, consider the "XOR" example in a classification context. Let the distribution of a class (Class 1) be characterized by two normals with means at opposite corners of the unit square, representing the X-Y (the two base attributes being discretized) plane, say $((0,0),(1,1))$. Let the distribution of the other class (Class 2) be characterized by two normals with means at the other two corners of a unit square, i.e., $((1,0),(0,1))$. Viewing the joint distribution when projected onto a single dimension, the typical discretization approach blurs the obvious separation that exists. While approaches to address this limitation have been proposed, they are usually very specific to a given task; thus, they are not interoperable and quite expensive in nature. In this paper, we propose to obtain discrete intervals based on the *correlation structure* inherent in the database. We present a PCA-based algorithm for discretization of continuous attributes in multivariate data sets. Our algorithm uses the distribution of *both* categorical and continuous attributes and the underlying correlation structure in the data set to obtain the discrete intervals. We capture the interactions among the continuous attributes by using correlation matrix. We also take into account the correlations among continuous and categorical attributes by using association patterns. This approach also ensures that *all attributes are used simultaneously* for deciding the cut-points, rather than one attribute at a time. An additional advantage is that the approach is able to work well on data sets with missing data (a common problem for many data analysis algorithms).

To summarize, the key contributions of this article are:

- Developing novel unsupervised PCA-based correlation preserving methods to efficiently discretize continuous attributes in high-dimensional data sets.
- Demonstrating the efficacy of the above algorithms as a preprocessing step for classical data mining algorithms such as frequent itemset mining and classification.
- Extending the above idea to work in the presence of missing values in multivariate data sets.
- Extensive experimental results on real and synthetic data sets demonstrating the discovery of meaningful intervals for the continuous attributes.

The rest of this article is organized as follows: In Section 2, we describe related work. In Section 3, we discuss

- *The authors are with the Department of Computer Science and Engineering, Ohio State University, 395, Dreese Laboratories, 2015 Neil Ave., Columbus, OH 43210.*
E-mail: {mehtas, srini, yanghu}@cse.ohio-state.edu.

Manuscript received 24 Nov. 2004; revised 21 Mar. 2005; accepted 30 Mar. 2005; published online 19 July 2005.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org and reference IEEECS Log Number TKDESI-0494-1104.

the key intuitions underlying the proposed methods and the basic algorithms. Section 4 reports on our empirical results. In Section 5, we compare with existing techniques and discuss various performance issues. Finally, we conclude in Section 6.

2 Related Work

Most discretization methods proposed in the past are univariate. They discretize continuous attributes individually and are only able to compute optimal cut-points for single-dimensional data sets. As a result, they cannot generate optimal intervals for all the involved continuous attributes in multidimensional cases. Dougherty et al. [4] present an excellent classification of current methods in discretization along three separate axes, viz., global versus local, supervised versus unsupervised, and static versus dynamic. Local methods, such as entropy-based discretization, are applied to regions of the data sets. Global methods, such as binning, produce the cut-points over entire data sets. Supervised algorithms leverage the information about the class labels to produce intervals, whereas unsupervised algorithms like equi-frequency discretization ignore the class labels. Finally, static methods derive the parameters (e.g., number of intervals) in each dimension separately, whereas dynamic methods try to find such parameters for all the dimensions simultaneously and, thus, can preserve interdependence among variables. The algorithm proposed in this article is *global, unsupervised, and dynamic* in nature.

Among the discretization methods reviewed by Dougherty et al. [4] and elsewhere, the following are the most germane to our work. The simplest discretization method is an unsupervised static method, known as equal-sized discretization. It calculates the maximum and minimum for the target attribute to be discretized and simply partitions the range observed into (some k) equal-sized intervals. Equal-frequency discretization is another unsupervised and static discretization method. For each target attribute in a data set, it first identifies all of its associated values in the data set and sorts them in order. It then partitions these values into intervals in such a way that each interval contains the same number of values.

Many supervised discretization algorithms have also been proposed in the past. Compared to unsupervised methods, supervised discretization requires information external to the data set of interest. Typical external information includes a set of user-specified thresholds. For instance, ChiMerge is a supervised, incremental, and bottom-up method implemented by Kerber [9]. The main criterion of ChiMerge is that the intrainterval similarity should be maximized while the interinterval similarity should be minimized. ChiMerge uses the Chi-Squared statistic to determine the interdependence of two adjacent intervals of a target attribute. Two adjacent intervals are merged if they are closely related. Entropy-based discretization is another supervised method, which is proposed by Fayyad and Irani [5]. It recursively selects the cut-points on each target attribute to minimize the overall entropy and determines the appropriate number of intervals (stopping criteria) by using the minimum-description-length principal. An improvement on this approach was presented recently by Subramonian et al. [23]. The work not only proposes an unsupervised discretization approach, but also implements visualization tools that allow end-users to refine the discretization results. Catlett [3] also proposed a supervised dynamic discretization method that recursively selects cut-points to maximize Quinlan's information gain [19]. It ends until a stopping criterion based on a set of heuristic rules is satisfied.

We formulated and discussed the 2D discretization problem [18]. To solve this problem efficiently, we considered an approximate solution based on simulated-annealing search. More recently, we extended the work to provide an exact solution to this problem in a time-optimal manner. Additionally, we considered the problem of discretizing in the presence of dynamic updates as well as proposed a parallel algorithm to solve the problem [17].

The recent work by Bay [2] and Ludl and Widmer [12] in the area of multivariate discretization is closely related to the work presented in this paper. Bay proposed a discretization approach that considers the interactions among all attributes. It first finely partitions all continuous attributes into intervals by using simple discretization techniques such as equal-width. Then, a merge phase is carried out iteratively on two adjacent intervals, where two intervals are merged into one if they correspond to two similar multivariate distributions. The merging process continues until no more intervals can be merged. Since the multivariate distribution involves all attributes, the resulting intervals are able to reflect the correlation among different attributes. However, such an approach can be computationally expensive and perhaps impractically so for high-dimensional and large data sets. Compared to Bay's method, our approach relies on Principal Component Analysis (PCA). By using PCA, our method intrinsically takes the interactions among all attributes into account. Moreover, we are able to take advantage of the statistics provided by PCA to effectively reduce the data to manageable sizes in the case of high dimensionality and large data size. This further reduction enables us to efficiently process very large high-dimensional data sets.

Ludl and Widmer [12] propose a "mutual structure projection" discretization method, which combines aspects of both supervised and unsupervised discretization. It computes the cut-points of a target attribute by first projecting all the other attributes to the target attribute. It then clusters the projected intervals and merges adjacent intervals if their difference is under a user-specified threshold. In order to project one continuous attribute to a target attribute, the proposed method requires a preliminary step that splits a continuous attribute into equal-width intervals. A major difference between this work and ours is that we take the interdependences among all attributes into account, while the interaction considered in their work is only pair-wise and piecemeal. Furthermore, as pointed out by the authors, such a projection-based method can lead to unnecessary splits. Several other groups have studied discretization [6], [21], [22] in the context of mining association rules. However, the discretization approaches discussed in these studies are typically not generic and can only be used for mining associations. For instance, Fukuda et al. [6] proposed a discretization approach that only serves for a specific association rule of interest.

3 Algorithms

In this section, we describe our correlation preserving discretization methods. Before getting into the details of our approach, we present the key intuition behind our work.

3.1 Key Intuition

Our claim is that the discretization of a particular continuous attribute must be sensitive to the influence of the other attributes in the data set, especially if there is a strong correlation structure in the data. This is often the case with real data sets. If we ignore the influence of other attributes, the resulting discretization can lead to a loss of information and our ability to discover important/hidden relationships will be impaired. To achieve this goal, we rely on two well-known techniques in data mining, namely, Principal Component Analysis (PCA) and Association Mining. PCA helps identify the correlation structure among the continuous attributes and, in conjunction with association patterns, it can help to effectively capture correlations in data sets containing both categorical and continuous attributes as we shall see later. These two techniques enable us to leverage the interactions among attributes to find the intervals and discretize all attributes simultaneously rather than one at a time. The use of PCA also helps to deal with data set with very high dimensionality. Next, we briefly describe these two techniques.

3.2 Principal Component Analysis

As indicated earlier, the attributes in high-dimensional data are often correlated, which is an underlying assumption of this paper. So, discretizing each attribute separately (univariate discretization) will lead to loss of hidden patterns and result in intervals that will not be meaningful. Due to strong interattribute correlation in most real data sets, it is possible to discretize a continuous attribute based on the other attributes. To analyze the interdependence among multiple attributes, we use the well-known Principal Component Analysis (PCA) [8]. PCA generates a set of n orthogonal vectors from the input data set with dimensionality of N, where $n < N$ and the n orthogonal directions preserve most of the variance in the input data set.

Consider a data set with N records and dimensionality d. In the first step of the PCA technique, we generate the correlation matrix of the continuous attributes in the data set. The correlation matrix is a $d \times d$ matrix in which the $(i,j)\mathrm{th}$ entry is equal to the correlation between the dimensions i and j. In the second step, we generate the eigenvectors $\{\overrightarrow{e_1}, \ldots, \overrightarrow{e_d}\}$ of this correlation matrix. These are the directions in the data which are such that when the data is projected along these directions, the second order correlations are zero. Let us assume that the eigenvalue for the eigenvector $\overrightarrow{e_i}$ is equal to λ_i. When the data is transformed to this new axis-system, the value λ_i is also equal to the variance of the data along the axis $\overrightarrow{e_i}$. The property of this transformation is that most of the correlation is retained in a small number of eigenvectors corresponding to the largest values of λ_i. In our work, unless otherwise specified, we retain the k $(k < d)$ eigenvectors that correspond to the largest eigenvalues which add up to 90 percent [15].

3.3 Association Pattern Mining

Discovery of association rules is an important problem in data mining. The prototypical application is the analysis of sales or *basket* data [1], although, more recently, it has been adopted in the domains of scientific computing, bioinformatics, and performance modeling. The problem can be formally stated as: Let $\mathcal{I} = \{i_1, i_2, \cdots, i_m\}$ be a set of m distinct attributes, also called *items*. Each transaction T in the database $\mathcal{D}$ of transactions has a unique identifier and *contains* a set of items such that $T \subseteq \mathcal{I}$. An *association rule* is an expression of form $A \Rightarrow B$, where $A, B \subset \mathcal{I}$ are sets of items called *itemsets* and $A \cap B = \emptyset$. Each itemset is said to have a *support* S if S percent of the transactions in $\mathcal{D}$ contain the itemset.

In addition to basic association patterns, we also define a metric that determines the similarity of association patterns generated by two data sets (or two samples of the same data set in our case). This metric will be adapted to determine the similarity between contiguous intervals when selecting the discretization cut-points.

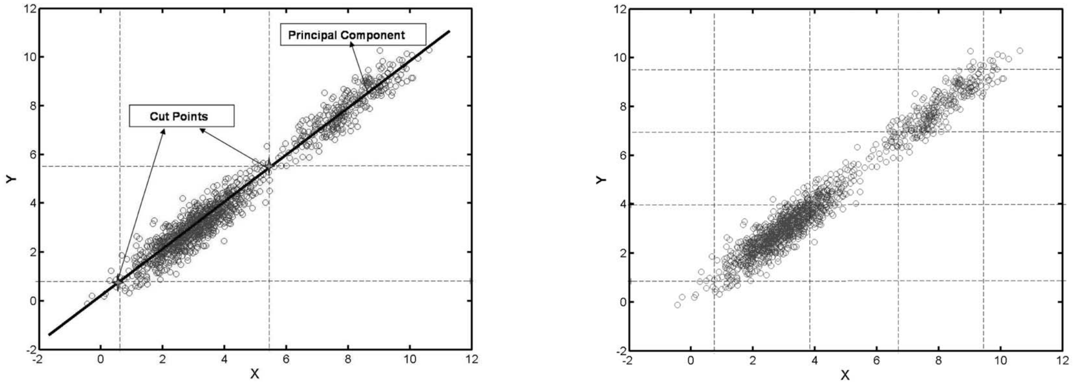
Let A and B be the two sets of frequent itemsets for database samples d_1 and d_2, respectively. For an element $x \in A$ (respectively, in B), let $\sup_{d_1}(x)$ (respectively, $\sup_{d_2}(x)$) be the frequency of x in d_1 (respectively, in d_2). Our metric is defined as:

$$Sim(d_1, d_2) = \frac{\sum_{x \in A \cap B} \max\{0, 1 - \alpha |\sup_{d_1}(x) - \sup_{d_2}(x)|\}}{\|A \cup B\|},$$

where α is a scaling parameter. The parameter α can be varied to reflect the significance that a user attaches to the variations of supports in d_1 and d_2. For $\alpha = 0$, the similarity measure is identical to $\frac{\|A \cap B\|}{\|A \cup B\|}$, i.e., support variance carries no significance. Values of Sim are bounded and lie in [0,1]. Sim also has the property of *relative ordinality*, i.e., if $Sim(X,Y) > Sim(X,Z)$, then X is more similar to Y than it is to Z. Note that while the above formulation does not explicitly consider correlations between itemsets (e.g., two itemsets (ABEK), (AEFK) that have many items in common are not treated differently), they are accounted for implicitly as all itemsets that can be formed by the common items (A,E,K) are part of the summation.

3.4 Correlation Preserving Discretization—An Overview

In this section, we provide an overview of our algorithm using a simple illustrative example. The data set used for this purpose is generated by a two-dimensional Gaussian distribution. No class labels are needed because our scheme is unsupervised. The attributes X and Y are highly correlated. Since the attributes are highly correlated, the first eigenvector alone is able to preserve most of the correlation. Fig. 1a shows the major principal component of the data set. Distance-based clustering is then applied to the data projected on this eigenvector to produce the cut-points. These cut-points are then projected back onto both original dimensions X and Y. Fig. 1a shows the cut-points (marked by the markers) produced by our approach. The



(a)

(b)

Fig. 1. (a) Data set discretized with our approach. (b) Equi-width discretization.

dotted lines show the reprojection of the cut-points onto the original dimensions. Our approach is able to correctly identify the high correlation in the data set and produces intervals such that the correlation is preserved. As shown in the figure, the highly correlated parts of the data set are not divided into smaller intervals. Fig. 1b shows the same data set discretized using the equi-width approach. As evident from the figure, equi-width discretization does not account for the correlation and can produce very unintuitive intervals.

Like equi-width and equi-frequency, our discretization scheme is unsupervised in nature. In other words, the discretization scheme does not take into account the class labels or any other goal attribute. Data discretized in a supervised manner often tends to be in favor of one mining task. For instance, the classification error tends to be minimized when the discretization takes class labels into account; however, an association rule mining algorithm may not be able to use the same discretized data to find meaningful patterns and associations. In contrast, unsupervised discretization is independent of the mining task. One may argue that unsupervised discretization can lead to high classification error since class labels are ignored. However, our extensive experiments show that the correlation-based discretization actually contradicts this belief. For example, C4.5 bootstrapped with our discretization scheme gives comparable or better accuracy when compared with classifiers with supervised discretization like C4.5 and Naive Bayes. We attribute this property of our discretization scheme to the use of correlation, which intrinsically captures the interaction among different attributes (continuous and categorical) in a global sense. Discretization based on global information also avoid the problems of data fragmentation and suboptimal intervals, which are common to most local information-based discretization approaches. Vilalta et al. [24] explain the data fragmentation problem in great detail. To reiterate, we propose an unsupervised and global information-based discretization algorithm. The data set discretized by our approach can be used for different data mining tasks.

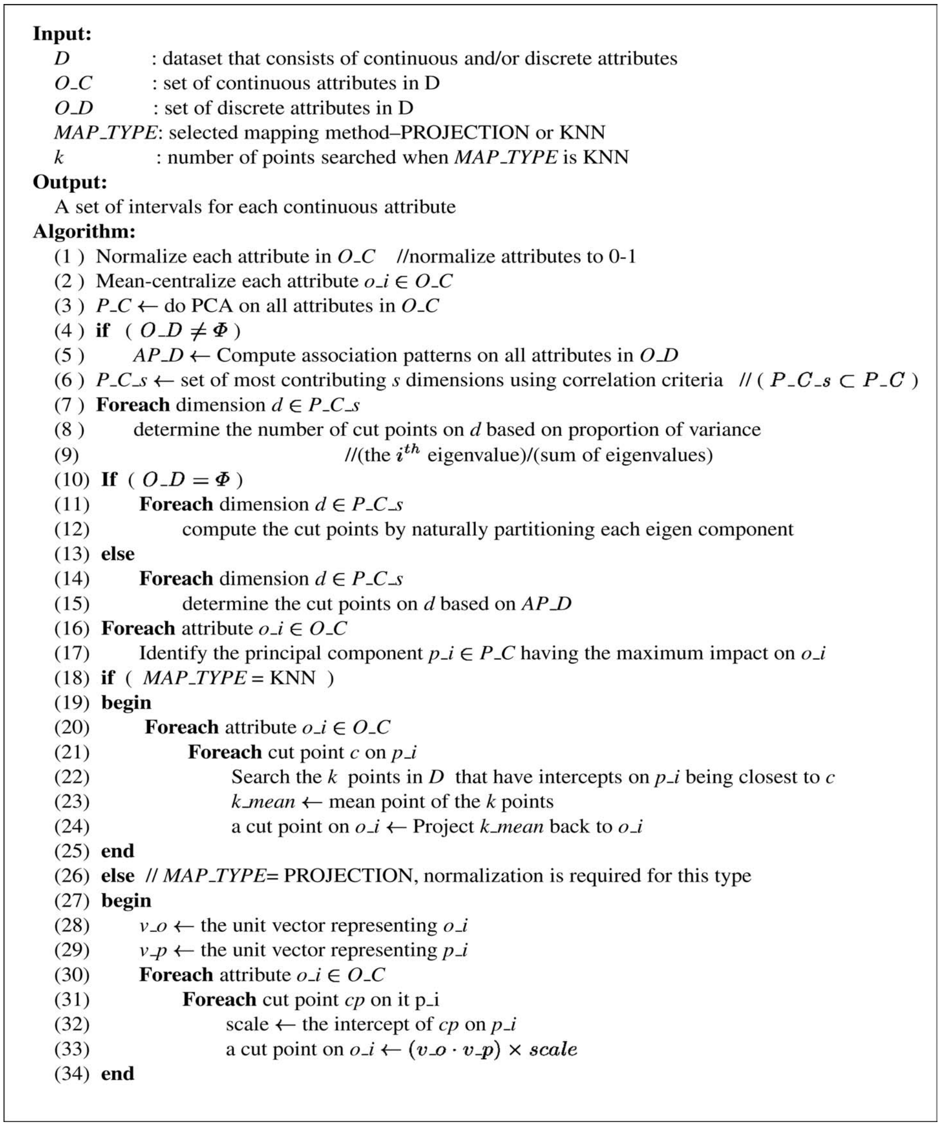
3.5 Correlation Preserving Discretization

Our algorithm is composed of the following steps (pseudocode in Fig. 2). While our algorithm is heuristic in nature, we strongly believe that each step in the proposed algorithm can be justified. We explain the rationale and/or the key intuition behind each step. We also remark on why the step is necessary and useful:

1. **Normalization and Mean Centralization**. The first step of the procedure involves normalizing all the continuous attributes (to lie between fixed intervals) and mean centralizing the data.

   Rationale. Mean centralization is a necessary and standard preprocessing step conducted prior to PCA computation.
2. **Eigenvector Computation**. We next compute the correlation matrix M from the data. The correlation matrix for a data set is positive semidefinite and can be expressed in the form $M = PNP^T$, where N is a diagonal matrix containing the eigenvalues $\lambda_1, \dots, \lambda_d$. The columns of P are the eigenvectors $\overrightarrow{e_1}, \dots, \overrightarrow{e_d}$, which form an orthogonal axis-system. We assume without loss of generality that the eigenvectors are sorted so that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$. To find these eigenvectors, we rely on the popular Householder reduction to tri-diagonal form and then apply the QL transform [8], which is the fastest method known to compute eigenvectors for symmetric matrices. Once these eigenvectors have been determined, we decide to retain only those which preserve the greatest amount of variance from the data. Well-known heuristics for deciding the number of eigenvectors to be retained may be found in [8].

   Rationale. This step identifies the directions of maximal variance. This step provides the correlation preserving nature to our approach since all the original dimensions that are highly correlated will be associated with the same eigendimension. PCA also facilitates the discretization approach to scale very well to high-dimensional data. For example, the UCI repository data set MUSK1 has 166 dimensions; however, by using PCA, we were able to reduce them to only nine dimensions. Therefore, the discretization



```
Input:
  $D$          : dataset that consists of continuous and/or discrete attributes
  $O\_C$       : set of continuous attributes in D
  $O\_D$       : set of discrete attributes in D
  $MAP\_TYPE$: selected mapping method–PROJECTION or KNN
  $k$          : number of points searched when $MAP\_TYPE$ is KNN
Output:
  A set of intervals for each continuous attribute
Algorithm:
  Normalize each attribute in $O\_C$   //normalize attributes to 0-1
  Mean-centralize each attribute $o\_i \in O\_C$
  $P\_C \leftarrow$ do PCA on all attributes in $O\_C$
  if ( $O\_D \neq \Phi$ )
      $AP\_D \leftarrow$ Compute association patterns on all attributes in $O\_D$
  $P\_C\_s \leftarrow$ set of most contributing $s$ dimensions using correlation criteria  // ( $P\_C\_s \subset P\_C$ )
  Foreach dimension $d \in P\_C\_s$
      determine the number of cut points on $d$ based on proportion of variance
                                  //(the $i^{th}$ eigenvalue)/(sum of eigenvalues)
  If ( $O\_D = \Phi$ )
      Foreach dimension $d \in P\_C\_s$
          compute the cut points by naturally partitioning each eigen component
  else
      Foreach dimension $d \in P\_C\_s$
          determine the cut points on $d$ based on $AP\_D$
  Foreach attribute $o\_i \in O\_C$
      Identify the principal component $p\_i \in P\_C$ having the maximum impact on $o\_i$
  if ( $MAP\_TYPE$ = KNN )
  begin
      Foreach attribute $o\_i \in O\_C$
          Foreach cut point $c$ on $p\_i$
              Search the $k$ points in $D$ that have intercepts on $p\_i$ being closest to $c$
              $k\_mean \leftarrow$ mean point of the $k$ points
              a cut point on $o\_i \leftarrow$ Project $k\_mean$ back to $o\_i$
  end
  else // $MAP\_TYPE$= PROJECTION, normalization is required for this type
  begin
      $v\_o \leftarrow$ the unit vector representing $o\_i$
      $v\_p \leftarrow$ the unit vector representing $p\_i$
      Foreach attribute $o\_i \in O\_C$
          Foreach cut point $cp$ on it p_i
              scale $\leftarrow$ the intercept of $cp$ on $p\_i$
              a cut point on $o\_i \leftarrow (v\_o \cdot v\_p) \times scale$
  end
```

Fig. 2. Algorithm.

only needs to process nine dimensions instead of 166 dimensions. Moreover, each dimension now can be processed independently because the second order correlations are zero (a property of PCA transformation). We also note that this step is effective if there is a strong correlation structure in a data set, which is true for most of the real data sets.

3. **Data Projection onto Eigenspace**. In this step, we project the data points in the original data set D onto the eigenspace, which is determined by the vectors we retain from the previous step. Each data point d in D will be projected onto the eigenspace.

   Rationale. To take advantage of dimensionality reduction, the points in the original space need to be projected onto the eigenspace. Furthermore, for noisy data sets, this step implicitly removes most of the noise by eliminating the dimensions of low eigenvalues [13].

4. **Discretization in Eigenspace**. Once all the data elements are projected onto the eigenspace, we discretize each of the dimensions separately in the eigenspace. Our approach to discretization here depends on whether a data set has categorical attributes or not. If there are no categorical attributes, we choose to apply simple distance-based clustering along each dimension in the eigenspace. The resulting cut-points are denoted as $c_{e_i}^1 \ldots c_{e_i}^n$ for each eigenvector or eigendimension $\overrightarrow{e_i}$.

   If the data set contains categorical attributes, then the discretization approach is as follows: First, we compute the frequent itemsets generated from all categorical attributes in the original data set D (for a user-determined support value). Let us refer to this as set A. We then split the eigendimension $\overrightarrow{e_i}$ into equal-frequency intervals (similar to the approach taken by Bay [2]) and compute the frequent itemsets in each interval that are constrained to being a subset of A. Next, we compute the similarity between contiguous intervals using the metric described in Section 3.3. If the similarity exceeds a user-defined threshold, the contiguous intervals are merged. Again, like the case without categorical attributes,

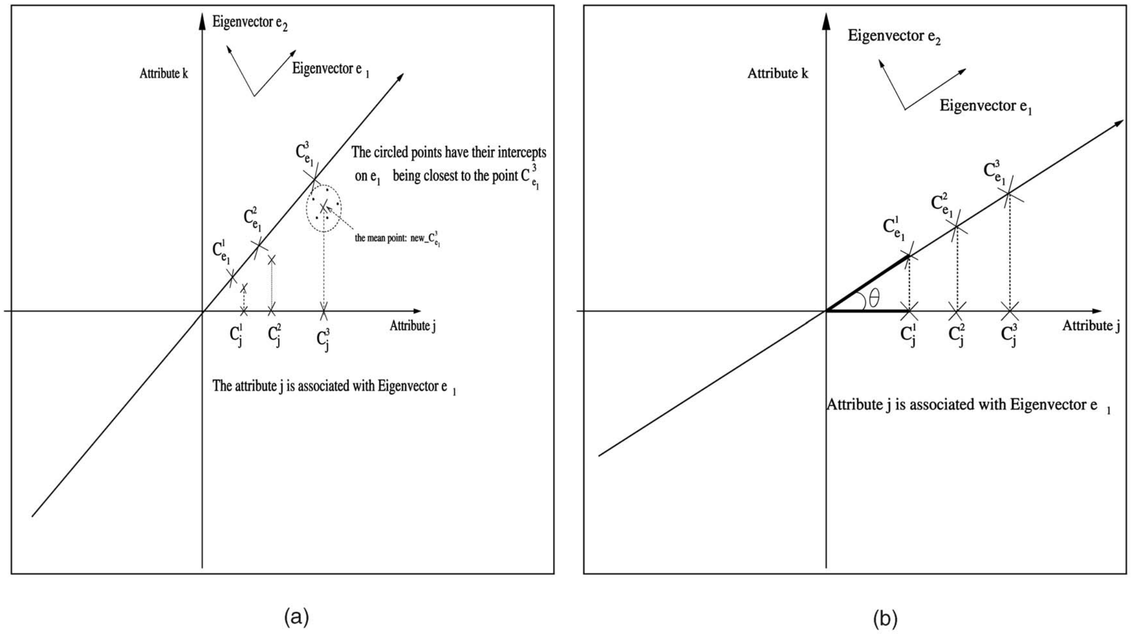

Eigenvector e$_2$
Eigenvector e$_1$
Attribute k
The circled points have their intercepts on e$_1$ being closest to the point $C^3_{e_1}$
the mean point: new_$C^3_{e_1}$
$C^1_{e_1}$
$C^2_{e_1}$
$C^3_{e_1}$
C^1_j
C^2_j
C^3_j
Attribute j
The attribute j is associated with Eigenvector e $_1$
(a)
Eigenvector e$_2$
Eigenvector e$_1$
Attribute k
θ
Attribute j
Attribute j is associated with Eigenvector e $_1$
(b)

Fig. 3. (a) K-NN. (b) Direct projection.

we are left with a certain number of cut-points along each eigendimension.

Rationale and Key Intuition. First, due to the property of PCA reduction, when we discretize attributes along a principal component, we do not need to consider the influence of other components since the second order correlations are zero. Thus, each principal component can be discretized independently. Second, the use of association rules and the use of A (as a constraint measure) ensures that correlations with respect to the categorical attributes are captured in the discretization process.

5. **Correlating Original Dimensions with Eigenvectors**. The step determines which original dimensions correlate most with which eigenvectors. This is a key step in factor analysis methods and can be computed by finding the contribution of dimension j on each of the eigenvectors ($\overrightarrow{e_1} \dots \overrightarrow{e_n}$), scaled by the corresponding eigenvalue and picking the maximum [10].

   Rationale and Key Intuition. This step is analogous to computing factor loadings in factor analysis. In essence, we find all the dimensions which are highly correlated in the original data space. The set of original dimensions associated with a single eigenvector will be discretized together, which really is what we want since these original dimensions are correlated with one another.

6. **Reprojecting Eigen Cut-Points to Original Dimensions**. We consider two strategies in our work. To explain our approaches for reprojection, let us assume without loss of generality that the jth original dimension is associated with eigenvector $\overrightarrow{e_i}$:

   a. **K-NN method**. To project the cut-point $c^3_{e_i}$ onto the original dimension j using this method, we first find the k nearest neighbor intercepts of $c^3_{e_i}$ on the eigenvector $\overrightarrow{e_i}$. The original points $p_1 \dots p_k$, representing each of the k nearest neighbors, as well as $p_{c_{e_i}}$, representing the cut-point $c^3_{e_i}$, are obtained (as shown in Fig. 3a). We then compute the mean (or, alternatively, median) value of the jth dimension for each of these points: $p_1 \dots p_k$ and $p_{c_{e_i}}$. This mean value represents the corresponding cut-point along the original dimension j (as shown in Fig. 3a).

   b. **Direct Projection**. The other approach we consider is *direct projection*. To project the cut-points $c^1_{e_i} \dots c^n_{e_i}$ onto the jth original dimension using this method, we need to find the angle between eigenvector $\overrightarrow{e_i}$ and the jth original dimension. The process is shown in Fig. 3b. The cosine of angle θ_{ij} can be calculated by the formula:

   $$cos(\theta_{ij}) = \overrightarrow{e_i}.\overrightarrow{o_j},$$

   where $\overrightarrow{o_j}$ is an N-dimensional unit vector along the jth dimension. Now, the cut-points $c^1_{e_i} \dots c^n_{e_i}$ can be projected to the original dimension j by multiplying it with $cos(\theta_{ij})$. The same process is applied for all cut-points.

Key Intuition. Regardless of which method is adopted, if eigenvector $\overrightarrow{e_i}$ is associated with more than one original dimension (especially common in high-dimensional data sets), the cut-points along that eigenvector $\overrightarrow{e_i}$ are projected back onto all the associated original dimensions, *which enables the discretization method to preserve the inherent correlation in the data.*

3.6 Extension: Handling Missing Data

Incomplete data sets *seemingly* pose the following problems for our discretization method: First, if values for continuous attributes are missing, then it affects the first step of our algorithm. Fortunately, if data is missing at random, then both the means and correlation matrix of the data can be suitably estimated using expectation-maximization-based approaches. Furthermore, in recent work, Parthasarathy and Aggarwal [15] show that estimating the projections of records with missing values along the principal components is more accurate than direct imputation, especially when large parts of the data set are missing. This fits in very nicely with the first three steps of our algorithm presented in the previous section, which enables us to handle missing continuous attributes effectively.

Second, if categorical attributes are missing, then it can affect Step 4 of our algorithm. However, the execution of the step will not be affected since frequent pattern algorithms naturally handle missing data. Missing entries can result in changes to the set of frequent itemsets found in each interval. This, in turn, can impact the similarity metric computation which can influence the discretization process. However, if these entries are also missing at random, our premise is that the structure of the rest of the data, within a given interval, will enable us to identify the relevant frequent patterns, thus ensuring that the similarity metric computation is unaffected. We will evaluate this premise in the next section.

4 Experimental Results and Analysis

In this section, we experimentally validate the proposed algorithms both in terms of the quality of the resulting discretization and its ability to uncover meaningful and interesting patterns. We demonstrate the general-purpose utility of the proposed work as a preprocessing step for data mining tasks such as association rule mining and classification. We also demonstrate that it readily adapts to data sets with missing information.

4.1 Data Sets and Experimental Setting

In Table 1,[1] we describe the data sets on which we evaluate the proposed algorithms. In terms of algorithmic settings, for the K-NN approach, we selected K to be 4 for all the experiments. (i.e., four nearest neighbors and the point projecting onto the cut-point itself are used to determine the cut-points along the original dimension(s)). Our default similarity metric threshold for merging intervals is 0.8 ($\alpha = 0$). All experiments were run on a 1GHz Pentium III PC with 512MB memory.

4.2 Qualitative Results Based on Association Rules

In this section, we focus on the discretization of the Adult data set (containing both categorical and continuous attributes) as a preprocessing step for obtaining association rules and compare these rules with published work on two multivariate discretization approaches: MVD and ME-MDL [11].

Due to the correlation preserving nature of our approach, we strongly believe that the intervals our methods produce are meaningful and should compare well with MVD. Before presenting results, we explain the notion of a meaningful interval. For an interval to be meaningful, the following two conditions should hold: First, the population within an interval should exhibit similar properties. Second, the population in different intervals should exhibit different properties. Thus, each cut-point should suggest a major change in population characteristics. Below, we discuss the cut-points obtained for several continuous attributes in the Adult data set and show that the intervals meet these two conditions:

TABLE 1
Data Sets Used in Evaluation

Dataset	#Records	#Attributes	#Continuous
Adult	48844	14	6
Shuttle	43500	9	9
Musk (1)	476	164	164
Musk (2)	6598	164	164
Cancer	683	8	8
Bupa	345	6	6
Credit1	690	14	6
Credit2	1000	20	7
Pima	768	8	8
Iris	150	4	4
Glass	214	10	10
Isolet	6238	616	616
Letter	20000	17	17

- Age—Table 2 shows the intervals obtained from our approaches (both KNN and projection) and the corresponding cut-points from MVD and ME-MDL on the Age attribute. The intervals found are very different from those that would be provided by equal-width or equal-frequency partitioning. Equal-width partitioning with the same number of intervals would result in cut-points at approximately every ten years. Equal-frequency partitioning would also suffer from a lack of resolution at young age ranges.

  First, at a coarse-grained level (as shown in Figs. 4a, 4b, and 4c which can be found on the Computer Society Digital Library at http://www.computer.org/tkde/archives.htm), the cut-points obtained by our methods and MVD are quite similar and intuitive. The cut-point at 63 corresponds to the retirement age. The intervals 19-22 and 23-24 are quite narrow but represent two different groups of people as illustrated below:

[1] All the data sets are obtained from UCI Data Repository (http://kdd.ics.uci.edu.)

TABLE 2
Cut-Points Obtained by Different Methods for Adult Data Set

Variable	Method	Cut-points
Age	**Projection**	19, 23, 25, 29, 34, 37, 40, 63, 85
	KNN	19, 23, 24, 29, 33, 41, 44, 62
	MVD	19, 23, 25, 29, 33, 41, 62
	ME-MDL	21.5, 23.5, 24.5, 27.5, 29.5, 30.5, 35.5, 61.5, 67.5, 71.5
Capital Gain	**Projection**	12745
	KNN	7298, 9998
	MVD	5178
	ME-MDL	5119, 5316.5, 6389, 6667.5, 7055.5, 7436.5, 8296, 10041, 10585.5, 21045.5, 26532, 70654.5
Capital Loss	**Projection**	165
	KNN	450
	MVD	155
	ME-MDL	1820.5, 1859, 1881.5, 1894.5, 1927.5, 1975.5, 1978.5, 2168.5, 2203, 2218.5, 2310.5, 2364.5,2384.5, 2450.5, 2581
Hours/Week	**Projection**	23, 28, 38, 40, 41, 48, 52
	KNN	19, 20, 25, 32, 40, 41, 50, 54
	MVD	30, 40, 41, 50
	ME-MDL	34.5, 41.5, 49.5, 61.5, 90.5

- 3.4 percent of people aged 19-22 have a bachelor's degree as compared to 22.7 percent of people aged 23-24.
- 6.0 percent of people aged 19-22 are married as compared to 17.0 percent of people in the other group.
- 19.0 percent of people aged 19-22 are in service as compared to 12.2 percent people aged 23-24.

MVD also obtains similar cut-points. However, we have an extra cut-point at age 37, which gives us intervals 34-37 and 38-40. MVD combines them into one interval 33-41. At first glance, these intervals do not seem meaningful, since there is usually not much difference in education level and job profiles of people in these two groups. However, upon a closer inspection on the resulting association patterns, we find that 26.0 percent of people in the 34-37 interval are *Never Married*, compared to 13.0 percent in the interval of 38-40. The interval 34-37 also captures an unusual behavior in the data set. Between the age of 34 and 37, there are 484 individuals who are still in school. *Moreover,* 181 *people in this interval are reported to be in between* 1st *and* 9th *grade!* We capture this anomalous behavior by deriving cut-points at age 34 and 37. This shows the usefulness of capturing the interactions between continuous attribute (age) and categorical attribute (education level). This distinction is obviously missed by MVD which reports cut-points at 33 and 41. Thus, our approach finds truly meaningful intervals where records in each interval exhibit radically different behavior from the adjoining intervals.

MVD's last cut-point is 62, which implies that, after the age of 62, there is not much change in demographic and employment variables. For the KNN method, we obtain an extra cut-point at age 85 because there are more females than males when $age \geq 85$.

- Capital Gain—The cut-points obtained by all four methods are shown in Table 2 and Figs. 5a and 5b (which can be found on the Computer Society Digital Library at http://www.computer.org/tkde/archives.htm). The cut-point from the projection method is $12,745$. MVD also gives one cut-point, which is at $5,178$. ME-MDL finds numerous cut-points which may not be very useful. Both of these methods separate people with high gains from people who make little or no gains to moderate gains. Using KNN, we are able to get even better cut-points. It divides the entire range into three intervals: $< \$7,298$ (low capital gain) which has 1,981 people, ($\$7,299, \$9,998$) (moderate gain) having 920 people, and $> \$9,999$ (high gain) having 1,134 people.
- Capital Loss—From Table 2 and Figs. 6a, 6b, and 6c (which can be found on the Computer Society Digital Library at http://www.computer.org/tkde/archives.htm), we see that MVD and our approaches give similar intervals on this attribute. One can observe that records are discretized based on whether a loss was declared. With such a cut-point, we were able to find the rule: $(CapitalLoss \geq 377) \Rightarrow (salary > 50K)$ (3 percent support, 49.3 percent confidence), which was also found by MVD [2]. Again, ME-MDL partitions the space very finely between range [2000,2600] as shown in Fig. 6d (which can be found on the Computer Society Digital Library at http://www.computer.org/tkde/archives.htm). It fails to find the cut-point around 500 which was found by the other three methods.
- Hours/week—Our cut-points for hours/week are also listed in Table 2 and Figs. 7a, 7b, and 7c (which can be found on the Computer Society Digital Library at http://www.computer.org/tkde/archives.htm). This is one attribute where we get significantly different cut-points from MVD. We believe that our cut-points are more intuitive. For example, MVD's first cut-point is at 30 hours/week, which implies anyone working less than 30 hours is

similar. This includes people in the age group (5 to 27) which is a group of very different people with respect to working habits, education level, etc. Yet, all of these are grouped together in MVD. Using KNN, we obtain the first cut-point at 19 hours/week. We are thus able to extract the rule $(Hours/week \leq 19) \Rightarrow (age \leq 20)$, which makes sense as children and young adults typically work less than 20 hours a week while others ($\geq 20\ years$) typically work longer hours. As another example, we obtain a rule that states that "people who work more than 54 hours a week typically earn less than 50K." Most likely, this rule refers to blue-collar workers. We note that there is a reduction in percentage of such people in the interval 50-54 hours, thus explaining the last couple of cut-points.

To summarize, the cut-points based on our approaches are more informative and do suggest major changes in one or more aspects of the population. This validates our claim that, by taking correlation into account, our discretization scheme is able to uncover more meaningful and hidden patterns. It also suggests that a discretization scheme using both association rules and principal components can well-capture the interactions between continuous attributes and categorical ones. Such an interaction has often been ignored in most of the earlier work. MVD also takes into account the interactions between attributes. We find similar cut-points as MVD. However, due to inherent correlation preserving nature, our scheme can uncover meaningful relationships which MVD might miss. Moreover, MVD can be computationally prohibitive especially in case of high-dimensional data sets such as Isolet (616 dimensions) and Musk (161 dimensions). ME-MDL looks at each dimension separately and, thus, misses most of the relationships among attributes. As a result, ME-MDL produces many cut-points (observed empirically) which are not useful and in some cases may miss important cut-points.

4.3 Qualitative Results Based on Classification

For both the direct projection and KNN algorithms, we use the discretization results with the C4.5 decision tree classifier. We compare our approach against various classifiers supported by the WEKA data mining toolkit.[2] We note that most of these classifiers use a supervised discretization algorithm (taking into account class label distributions) as a preprocessing step, whereas our approach is unsupervised. For example, C4.5 and PART perform entropy-based discretization while ONER attempts to form bins containing a majority of a particular class. The Naive Bayes implementation of WEKA assumes that all the attributes are conditionally independent for a given class. So, each attribute is modeled without considering other attributes (except the class labels) in data sets.

To evaluate our approaches, we first apply our discretization methods to a data set, then append the class labels back to the discretized data sets before applying the C4.5 classifier. All results reported here use 10-fold cross-validation. For large data sets, such as Letter and Isolet, WEKA fails to report any result due to high memory requirement. If this is the case, we cite the best results reported in published work that we are able to collect. Table 3 shows the classification accuracy of our approaches (last two columns) as compared to seven different classifiers (first seven columns) on the nondiscretized data sets. From the results, it is clear that our methods, coupled with C4.5, often outperform the other approaches (including C4.5 itself).

Our schemes perform very well on high-dimensional data sets: Musk(1), Musk(2), and Isolet. For Cancer, Credit1, and Credit2, our scheme again outperforms the existing schemes, however, the gain in accuracy is moderate. Our classification accuracy is comparable for Adult, Pima, Glass, and Letter data sets. For Bupa and Iris, the difference in accuracy is 1 percent. This can be attributed to the weak correlation structure present in both the data sets. This was also pointed out by Parthasarathy and Aggrawal [15].

To summarize, our schemes perform very well on high-dimensional data sets: Musk(1), Musk(2), and Isolet. We do better in spite of the fact that our approaches are unsupervised. This validates our claim that the inherent correlations preserved by our methods are useful. Our approach also lends itself to faster classifier construction time. Decision trees built on top of the discretized data sets were constructed around 10-20 percent faster on an average. This does not represent a significant saving in execution time for our data sets (since they are quite small), but it can become quite significant for large data sets.

4.4 Experiments with Missing Data

In this set of experiments, we compare the impact of missing data on the classification results on all the data sets. For each data set, we randomly eliminated a certain percentage of the data set and then adopted the approach described in Section 3.6. Figs. 8 and 9 document these results. Clearly, as the percentage of missing data increases, classification accuracy decreases. However, this decrease in accuracy is not too bad even when 30 percent of the data is missing. As shown in the figures, when 10 percent of the data is missing, the differences in classification accuracy are relatively insignificant, which indicates that our discretization approach can tolerate missing data quite well. The high classification accuracy, even in the presence of missing data, further solidifies our claim that discretization should take into account the interactions among attributes.

The conclusion here is that one can extract meaningful discrete intervals for continuous attributes if the data is missing at random.

4.5 Compression of Data Sets

In this section, we evaluate the compressibility that can be achieved by discretization, which is a useful utility in the case of large data sets or data warehousing environments. Note that, here, we do not consider classic compression utilities such as *gzip*, which are orthogonal to our approach and can be applied on top of our approach to achieve further compression. Discretization of continuous attributes enables fixed format compression, wherein a record can be reduced to a bit string and each attribute in a record is associated with a specific contiguous set of bits in the bit string. Continuous attributes are usually floating numbers and, thus, require the minimum four bytes to represent. However, by discretizing them, we can easily reduce the storage requirements for such attributes. Table 4 shows the results of compression on various data sets. As we can see from the results, on most data sets, we achieve a compression factor around 3 and, in some cases, the results are even better. A factor of 4 can be further achieved if gzip is used on discretized data sets.

[2] http://www.cs.waikato.ac.nz/~ml/.

TABLE 3
Classification Results (Error Comparison—Best Results in Bold)

Dataset	C4.5	IBK	PART	Bayes	ONER	Kernel-based	SMO	Projection	KNN
Adult	**15.7**	20.35	N/A	15.8	16.8	19.54	17	**15.7**	**15.7**
Shuttle	0	0	0	5.1	0	0	0	0	0
Musk (1)	17.3	17.2	18.9	25.7	39.4	17.3	15.6	**14.1**	14.6
Musk (2)	4.7	4.7	4.1	16.2	9.2	5.1	N/A	**4.1**	**4.1**
Cancer	5.4	4.3	4.8	**4.1**	8.2	5.1	4.3	**4.1**	**4.1**
Bupa	**32**	40	35	45	45	36	43	33	34
Credit1	15	14.9	17	23.3	15.5	17.4	15	**14.8**	**14.9**
Credit2	26.1	27.3	17	28.8.	24.1	35.7	25.9	**24**	24.4
Pima	29.9	30.08	26.17	23.96	28.52	29.04	**23.57**	25	24.1
Iris	4.7	4.67	**4.0**	4.67	7.33	4.67	5.1	5.0	4.9
Glass	32	29.91	**29**	50.47	41.12	30.7	42.12	**29**	29.3
Isolet	6.1	N/A	N/A	N/A	N/A	N/A	N/A	**4.7**	4.9
Letter	**12.9**	N/A	N/A	N/A	N/A	N/A	N/A	13.1	13

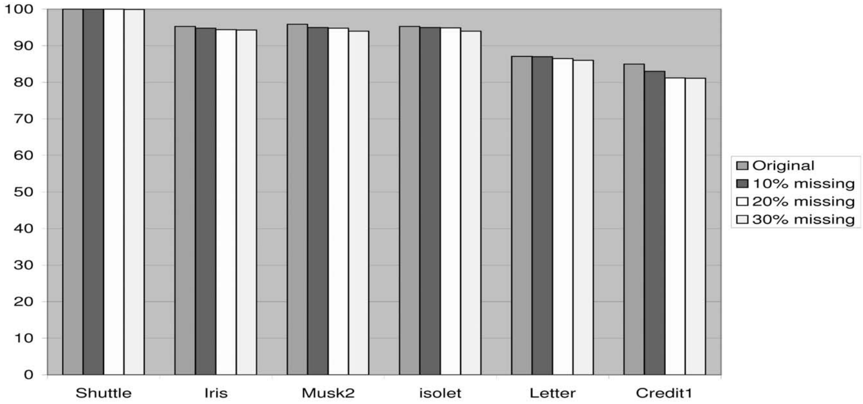


Fig. 8. Classification accuracy for incomplete data sets.

5 Discussion

5.1 Comparison with MVD and ME-MDL

In terms of quantitative experiments, we could not perform a direct comparison with the MVD method as the source/executable code was not available to us. We will point out that, for the large data sets (both in terms of dimensionality and number of records), our approaches take on the order of a few seconds in running time. The order complexity of our method is bounded by the order complexity of each step. The steps that dominate the execution time are: the one to compute the correlation with complexity of $O(d^2 \cdot N)$ and the one to compute the eigenvectors with complexity of $O(d^3)$. Here, d is the number of dimensions and N is the number of records in the data set. The order complexity for the rest of the steps depends on the number of cut-points. If we use the KNN strategy, the value of K will also need to be taken into account. The other steps are at most linear of the number of dimensions.

In comparing our methods with MVD, we find similar cut-points (to MVD). However, we do so at a fraction of the cost. Our benefits over MVD in terms of execution time can be ascribed to the fact that our discretization is carried out in a lower dimension space obtained as a result of PCA transformation. To illustrate this point, we use the Isolet data set, which has 616 dimensions, as one example.

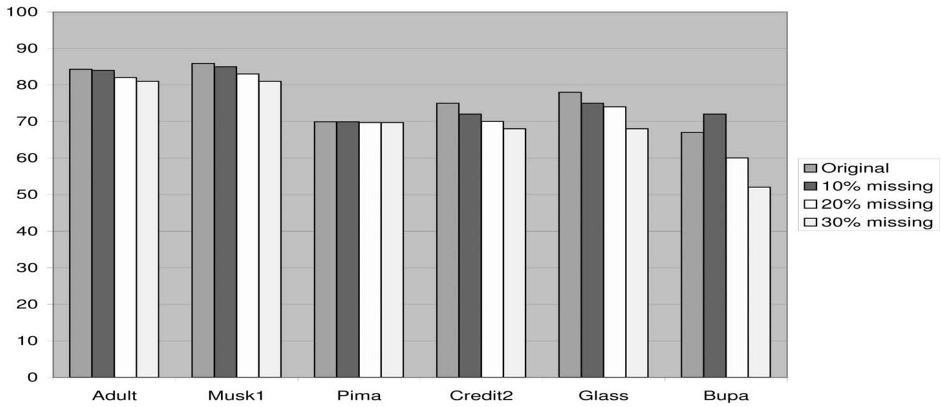


Fig. 9. Classification accuracy for incomplete data sets.

MVD would need to first partition all of these 616 dimensions and then merge adjacent intervals on each one based on their multivariate distribution. On the other hand, our approach only needs to consider 90 independent eigendimensions derived from PCA.

ME-MDL is significantly different from our method. It considers each attribute independently and tries to minimize the classification error. ME-MDL finds many cut-points which are very close of each other. Therefore, as pointed out by Bay [2], it is likely to find many meaningless intervals.

TABLE 4
Compression Results

Datasets	Original (in Bytes)	Byte Compressed and Discretized	Compression Factor
Adult	537350	195400	2.75
Shuttle	1153518	478500	2.4
Musk1	85680	29693	2.89
Musk2	1319800	422336	3.13
Cancer	6830	3415	2.00
Bupa	3795	1035	3.67
Credit1	28735	3450	8.33
Credit2	79793	16000	4.99
Pima	46000	45317	0
Iris	3196	4551	1.42
Glass	12888	3072	3
Isolet	33554432	4194304	8
Letter	712704	300000	2.37

5.2 Future Work

When discretizing a data set that has both continuous and categorical attributes, our approach makes use of association rules. The use of association rules poses two questions: 1) how to choose an "appropriate" minimum support threshold and 2) how to deal with the high computational complexity when mining association rules from a very large data set?

Our solution to the first issue is similar to the one adopted by most researchers: we choose the minimum support threshold empirically. We note that the threshold should not be too extreme. Too large a value will defeat the process (categorical correlations will be ignored) and too small a value may result in unacceptable computational time. We observe that, in our experiments, if the number of itemsets is at least a hundred, it works well, provided that there are several categorical attributes. Also, please note that if the discretization is a preprocessing step for frequent itemset mining, the same minimum support value (user-defined) can be directly used here.

To handle the second issue, a data set can be sampled and associations can then be computed just based on the sample [14]. Another alternative is to use only the first few levels of the itemset lattice [7]. One can also sample the lattice space [16] and use the sampled lattice to compute the associations. Moreover, these techniques can be combined to further speed up the process. We are currently working on some of these issues. We also plan to extend this work by applying out-of-core PCA, which will help us to handle very large data sets more effectively. We also plan to use the ideas proposed by Rabani and Toledo [20] toward this purpose. Additionally, we would like to try different data transformation schemes such as the logarithm transformation before normalization and systematically study the effect of different transformations. Finally, we plan to extend the proposed approaches so that they can also be used to discretize dynamic data, where the cut-points on an attribute may change over time [17].

6 Conclusions

In this paper, we proposed correlation preserving discretization, an efficient method that can effectively discretize continuous attributes even in high-dimensional data sets, by accounting for the inherent correlations in the data in a multivariate context. The algorithm considers the distribution of *both* categorical and continuous attributes and the underlying correlation structure in the data set to obtain the discrete intervals. The approach ensures that *all attributes are used simultaneously* for deciding the cut-points rather than one attribute at a time. We believe that the intervals produced by our scheme are more useful and intuitive than MVD. This fact is reflected in the detailed analysis of frequent itemsets on the Adult data set.

We demonstrated the effectiveness of the approach on real data sets, including high-dimensional data sets, as a preprocessing step for classification as well as for frequent association mining. We show that the resulting data sets can be easily used to store data in a compressed fashion ready to be used by other data mining tasks. We also propose an extension to the algorithm so that it can deal with missing values effectively and we vidate this idea. We also show that the intervals obtained are meaningful, intuitive, and can uncover hidden patterns in data.

Acknowledgments

This work is funded by US National Science Foundation grants ITR-NGS, ACI-0326386, CAREER IIS-0347662, and SOFTWARE ACI-0234273. All authors contributed equally to the work. The author list is arranged in alphabetical order by last name.

References

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," *Proc. 20th Very Large Database Conf.,* pp. 487-499, 1994.

[2] S.D. Bay, "Multivariate Discretization for Set Mining," *Knowledge and Information Systems,* vol. 3, no. 4, pp. 491-512, 2001.

[3] J. Catlett, "Changing Continuous Attributes into Ordered Discrete Attributes," *Proc. European Working Session on Learning,* pp. 164-178, 1991.

[4] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features," *Proc. Int'l Conf. Machine Learning,* pp. 194-202, 1995.

[5] U.M. Fayyad and K.B. Irani, "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning," *Proc. 13th Joint Conf. Artificial Intelligence,* pp. 1022-1029, 1993.

[6] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama, "Mining Optimized Association Rules for Numeric Attributes," *Proc. 15th ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems,* pp. 182-191, 1996.

[7] A. Ghoting, M. Otey, and S. Parthasarathy, "Loaded: Link-Based Outlier and Anomaly Detection in Evolving Data Sets," *Proc. Int'l Conf. Data Mining,* pp. 387-390, 2004.

[8] I.T. Jolliffe, *Principal Component Analysis.* Springer-Verlag, 1986.

[9] R. Kerber, "Chimerge: Discretization of Numeric Attributes," *Proc. Nat'l Conf. Artificial Intelligence,* pp. 123-128, 1991.

[10] J.-O. Kim and C.W. Mueller, *Factor Analysis: Statistical Methods and Practical Issues.* Sage Publications, 1978.

[11] R. Kohavi and M. Sahami, "Error-Based and Entropy-Based Discretization of Continuous Features," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining,* pp. 114-119, 1996.

[12] M.-C. Ludl and G. Widmer, "Relative Unsupervised Discretization for Association Rule Mining," *Proc. Fourth European Conf. Principles and Practice of Knowledge Discovery in Databases,* pp. 148-158, 2000.

[13] C. Papadiitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent Semantic Indexing: A Probabilistic Analysis," *Proc. ACM Symp. Principles of Database Systems,* pp. 159-168, 1998.

[14] S. Parthasarathy, "Efficient Progressive Sampling for Association Rules," *Proc. IEEE Int'l Conf. Data Mining,* pp. 354-361, 2002.

[15] S. Parthasarathy and C.C. Aggarwal, "On the Use of Conceptual Reconstruction for Mining Massively Incomplete Data Sets," *IEEE Trans. Knowledge and Data Eng.,* pp. 1512-1521, 2003.

[16] S. Parthasarathy and M. Ogihara, "Clustering Homogeneous Distributed Data Sets," *Proc. Int'l Conf. Practical Applications of Knowledge Discovery and Data Mining,* pp. 566-574, 2000.

[17] S. Parthasarathy and A. Ramakrishnan, "Parallel Incremental 2D-Discretization on Dynamic Data Sets," *Proc. Int'l Parallel and Distributed Processing Symp.,* pp. 247-254, 2002.

[18] S. Parthasarathy, R. Subramonian, and R. Venkata, "Generalized Discretization for Summarization and Classification," *Proc. Practical Applications of Discovery and Data Mining,* pp. 219-239, 1998.

[19] J.R. Quinlan, *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[20] E. Rabani and S. Toledo, "Out-of-Core SVD and QR Decompositions," *Proc. 10th SIAM Conf. Parallel Processing for Scientific Computing,* p. 10, 2001.

[21] R. Rastogi and K. Shim, "Mining Optimized Association Rules with Categorical and Numeric Attributes," *Knowledge and Data Eng.,* vol. 14, no. 1, pp. 29-50, 2002.

[22] R. Srikant and R. Agrawal, "Mining Quantitative Association Rules in Large Relational Tables," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* pp. 1-12, 1996.

[23] R. Subramonian, R. Venkata, and J. Chen, "A Visual Interactive Framework for Attribute Discretization," *Proc. Third Conf. Knowledge and Data Discovery,* pp. 218-225, 1997.

[24] R. Vilalta, G. Blix, and L. Rendell, "Global Data Analysis and the Fragmentation Problem in Decision Tree Induction," *Proc. Ninth European Conf. Machine Learning,* pp. 312-326, 1997.

Sameep Mehta received the BS degree in information science from the University of Delhi, New Delhi, India, in 2001. He has been a PhD student in the Department of Computer Science and Engineering at Ohio State University, Columbus, Ohio, since September 2001. His research interests include scientific data mining, visualization, spatial-temporal data mining, and time series analysis.

Srinivasan Parthasarathy received the BE degree in electrical engineering from the University of Roorkee (now IIT-Roorkee), India, in 1992 and the MS degree in electrical and computer engineering from the University of Cincinnati, Ohio, in 1994. Subsequently, he received the MS and PhD degrees in computer science from the University of Rochester in 1996 and 2000, respectively. While at Rochester, he spent a year consulting for Intel's Microcomputer Research Laboratory. He is currently on the computer science and engineering faculty at Ohio State University. He also holds a joint appointment with the newly formed department of biomedical informatics. He is a recipient of the US National Science Foundation CAREER award, the DOE Early Career Principal Investigator Award, and an Ameritech Faculty Fellowship. His research interests are in data mining and parallel and distributed computing systems. He has published more than 80 refereed technical papers related to these areas. His work has recently received a couple of best paper awards: one at the IEEE International Conference on Data Mining 2002 and one at the SIAM International Conference on Data Mining 2003. He is a member of the IEEE, the ACM, and SIAM.

Hui Yang received the MS degree in computer science from Ohio State University, Columbus, Ohio, in 2002. She is currently a PhD student in the Department of Computer Science and Engineering at Ohio State University, Columbus, Ohio. Her current research interests include spatial and spatio-temporal data mining and spatio-temporal reasoning.