Correlation Preserving Discretization

Sameep Mehta, Srinivasan Parthasarathy and Hui Yang
Department of Computer and Information Science,The Ohio State University
Contact: (mehtas, srini, yanghu)@cis.ohio-state.edu

Abstract

Discretization is a crucial preprocessing primitive for a variety of data warehousing and mining tasks. In this article we present a novel PCA-based unsupervised algorithm for the discretization of continuous attributes in multivariate datasets. The algorithm leverages the underlying correlation structure in the dataset to obtain the discrete intervals, and ensures that the inherent correlations are preserved. The approach also extends easily to datasets containing missing values. We demonstrate the efficacy of the approach on real datasets and as a preprocessing step for both classification and frequent itemset mining tasks. We also show that the intervals are meaningful and can uncover hidden patterns in data.

Keywords: Data preprocessing, Principal Components Analysis, Data Mining/Summarization

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 seen use as a preprocessing step for frequent itemset discovery applications[14], 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, 9] have pointed out that such methods are limited in a multivariate context resulting in non-optimal solutions. While approaches to address this limitation have been proposed, they are usually very specific to a given task and thus are not inter-operable, and are quite expensive in nature.

In this article 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 datasets. Our algorithm uses the distribution of *both* categorical and continuous attributes and the underlying correlation structure in the dataset to obtain the discrete intervals. 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 datasets containing missing data (a common problem for many data analysis algorithms).

To summarize, the key contributions of this article are:

- Novel unsupervised PCA-based correlation preserving methods for efficiently discretizing continuous attributes in high dimensional datasets.
- 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 datasets.
- Extensive experimental results on real and synthetic datasets demonstrating the discovery of meaningful intervals for the continuous attributes and accuracy in prediction of missing values.

The rest of this article is organized as follows. In Section 2 we describe related work. In Section 3 we discuss the key intuitions underlying the proposed methods and the basic algorithms. Section 4 reports on our empirical results. Finally we conclude with directions for future work in Section 5.

2 Related Work

Most work on discretization focuses on discretizing a single continuous attribute. These methods can compute optimal discretizations along one dimension, but they cannot generate optimal discretizations in the two dimensional

case. Dougherty *et al* [4], present an excellent classification of current methods in discretization along three separate axes, viz., global vs. local, supervised vs. unsupervised and static vs. dynamic. Among the discretization methods reviewed in [4] and elsewhere, the following are the most germane to our work.

The simplest discretization method is an unsupervised static method called equal sized discretization. It calculates the maximum and minimum for the attribute being discretized and simply partitions the range observed into (some k) equal sized intervals. Another unsupervised static method is equal frequency discretization. It counts the number of values we have from the attribute that we are trying to discretize and partitions it into intervals containing the same number of examples. ChiMerge is a supervised, incremental, bottom up method described by Kerber [8]. It suggests that intra-interval similarity should be maximized and inter-interval similarity should be minimized. ChiMerge uses the Chi-Squared statistic to determine the independence of the class from the two adjacent intervals. Entropy discretization is an supervised dynamic method described in Fayyad et al [5]. Entropy discretization recursively selects the cut-points that minimize entropy and uses the minimum-description-length principle to determine the appropriate number of intervals (stopping criteria). An improvement on this approach was presented recently by Subramonian et al [15]. Maass [10] provides an efficient algorithm that minimizes classification error. Catlett [3] describes a supervised dynamic discretization method that recursively selects cut-points maximizing Quinlan's gain [11] until a stopping criteria based on a set of heuristic rules ends the recursion.

Closely related to the work presented in this paper, in the area of multi-variate discretization, is the recent work by Bay[2] and also work by Ludl and Widmer[9]. Like our algorithm, Bay proposes an approach to discretization that considers the interactions among all attributes. His approach first finely partitions all continuous attributes into intervals by using simple discretization techniques such as equal-width. Then, a merge phase was carried out iteratively on two adjacent intervals based on the similarity between the multivariate distributions corresponding to the two intervals. Such a merging process continued until no more eligible intervals are available. Since the multivariate distribution involves all attributes, the resulting intervals were able to reflect the correlation among different attributes. The main limitation of his approach is that it can be computationally expensive, and perhaps impractically so for high dimensional and large datasets. Compared to Bay's method, our approach relies on Principle Component Analysis (PCA). By using PCA, our method intrinsically takes the interactions among all attributes into account. What is more, we are able to take advantage of the statistics provided by PCA to effectively reduce the data in the case of high-dimensionality. This further reduction enables us to deal with very large high-dimensional datasets in an efficient way.

Ludl and Widmer [9] suggest deriving the cut points for a continuous attribute by first projecting all the other attributes to the target attribute, and then by clustering the projected intervals and finally merging adjacent intervals if their difference is under a user-specified threshold. In order to project one continuous attribute to the other, their method requires a preliminary step that split a continuous attribute into equal-width intervals. A major difference between this work and our work is that we take the inter-dependences among all attributes into account, while the interaction considered in their work is only pair-wise and piecemeal. Several other groups have studied discretization [14, 6, 12, 13] in the context of mining association rules. However, the discretization approaches discussed in these studies are typically not generic and can be used only for mining associations. For instance, Fukuda *et al* [6] proposed a discretization approach that served only 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 first 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 dataset particularly if there is a strong correlation structure to the data. This is most often the case with real datasets. If we ignore the influence of other attributes, the resulting discretization can lead to a loss of information and our ability to discover important relationships among attributes is reduced.

To account for and preserve the correlation structure when performing discretization, we rely on two well known techniques in data mining, frequent association mining and principle component analysis. Principle component analysis helps identify the correlation structure among the continuous attributes and in conjunction with association patterns can help effectively capture correlations in datasets containing both categorical and continuous attributes as we shall see later. Next, we briefly describe these two techniques.

3.2 Principle 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 inter-attribute correlation in most real datasets it is possible to discretize a continuous attribute based on the other attributes. To analyze the inter-dependence among multiple attributes, we use the well-known Principle Component Analysis (PCA) [7]. PCA which generates a set of n orthogonal vectors in the input dataset with dimension N, where $n < N$ and the n orthogonal directions preserve most of the variance in the input dataset.

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 attribute in data set. The correlation matrix is a $d \times d$ matrix in which the (i, j)th entry is equal to the correlation between the dimensions i and j. In the second step we generate the eigenvectors $\{\overline{e_1} \ldots \overline{e_d}\}$ 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 $\overline{e_i}$ 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 $\overline{e_i}$. 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 $k < d$ eigenvectors that correspond to the largest eigenvalues which add up to 80%.

3.3 Association Pattern Mining

Discovery of association rules is an important problem in database 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 $A \Rightarrow B$, 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 $S\%$ 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 datasets (or two samples of the same dataset in our case). This metric will be adapted to determine the similarity between contiguous intervals for selecting the discretization cut-points.

Let A and B respectively be the two sets of frequent itemsets for a database sample d_1 and that for a database sample d_2. 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 has a default value of 1 and can be modified to reflect the significance the user attaches to variations in supports. For $\alpha = 0$ the similarity measure is identical to $\frac{\|A \cap B\|}{\|A \cup B\|}$, i.e., support variance carries no significance. Sim values 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

Our algorithm is composed of the following steps (pseudo-code in Figure 1):

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. Mean centralization is a common preprocessing element conducted prior to PCA computation. Normalization in our case is required to reduce the impact of attributes that have high variance. This aspect is discussed in greater detail later in this section.

2. **Eigenvector Computation**: We next compute the correlation matrix M from the data. The covariance matrix for a data set is positive semi-definite and can be expressed in the form $M = PNP^T$, where N is a diagonal matrix containing the eigenvalues $\lambda_1 \ldots \lambda_d$. The columns of P are the eigenvectors $\overrightarrow{e_1} \ldots \overrightarrow{e_d}$, which form an orthogonal axis-system. We assume without loss of generality that the eigenvectors are sorted so that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$. To find these eigenvectors, we rely on the popular Householder reduction to tri-diagonal form and then apply the QL transform [7], which is the fastest known method to compute eigenvectors for symmetric matrices. Once these eigenvectors have been determined, we decide to retain only those which preserve the greatest amount of

Input:

D : dataset that consists of contiuous 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 retrived when *MAP_TYPE* is PROJECTION

Output:

A set of intervals for each continuous attribute

Algorithm:

(1) **if** (NORMALIZE) //user-specified option
(2) Normalize each attribute in *O_C* //normalize attributes to 0-1
(3) Mean-centralize each attribute *o_i* ∈ *O_C*
(4) *P_C* ← do PCA on all attributes in *O_C*
(5) **if** ($O\_D \neq \Phi$)
(6) *AP_D* ← Compute association patterns on all attributes in *O_D*
(7) *P_C_s* ← set of most contributing *s* dimensions using correlation criteria // ($P\_C\_s \subset P\_C$)
(8) **Foreach** dimension *d* ∈ *P_C_s*
(9) determine the number of cut points on *d* based on proportion of variance
(10) //(eigenvalue of i/sum of eigenvalues)
(11) **If** ($O\_D \neq \Phi$)
(12) **Foreach** dimension *d* ∈ *P_C_s*
(13) compute the cut points by naturally partitioning each eigen component
(14) **else**
(15) **Foreach** dimension *d* ∈ *P_C_s*
(16) determine the cut points on *d* based on *AP_D*
(17) **Foreach** attribute *o_i* ∈ *O_C*
(18) Identify the principal component *p_i* ∈ *P_C* having the maximum impact on *o_i*
(19) **if** (*MAP_TYPE* = KNN)
(20) **begin**
(21) **Foreach** attribute *o_i* ∈ *O_C*
(22) **Foreach** cut point *c* on *p_i*
(23) **begin**
(24) Retrive the *k* points in *D* that have intercepts on *p_i* being closest to *c*
(25) *k_mean* ← mean point of the *k* points
(26) a cut point on *o_i* ← Project *k_mean* back to *o_i*
(27) **end**
(28) **end**
(29) **else** // *MAP_TYPE*= PROJECTION, normalization is required for this type
(30) **begin**
(31) *v_o* ← the unit vector representing *o_i*
(32) *v_p* ← the unit vector represneting *p_i*
(33) **Foreach** attribute *o_i* ∈ *O_C*
(34) **begin**
(35) **Foreach** cut point *cp* on it p_i
(36) **begin**
(37) scale ← the intercept of *cp* on *p_i*
(38) a cut point on *o_i* ← $(v\_o \cdot v\_p) \times scale$
(39) **end**
(40) **end**
(41) **end**

Figure 1. Algorithm

variance from the data. Well known heuristics for deciding the number of eigenvectors to be retained may be found in [7]. Let us assume that a total of $m \leq d$ eigenvectors $\overline{e_1} \ldots \overline{e_m}$ are retained (in our case preserving 80% of correlation).

3. **Data Projection onto Eigen-space**: In the next step we project our data elements D, onto the eigen-space determined by the vectors we retain from the previous step. Each data point d in the original space is projected on to the eigen-space where d_{e_i} is the projection of d on the i^{th} eigenvector.

4. **Discretization along Eigen-space**: Once all the data elements are projected onto the eigen-space we discretize each of the dimensions in the eigen-space. *It is important to note that when the data is projected along these directions, the second order correlations are zero, so we do not have to worry about the interactions of other dimensions in this space.*

   Our approach to discretization here depends on whether we have categorical attributes in the dataset or not. If there are no categorical attributes, we choose to identify natural intervals (distance-based clustering along each dimension) along each dimension based on the projections of the data elements composed in D. The resulting set of cutpoints are denoted as $c^1_{e_i} \ldots c^n_{e_i}$ for each eigenvector or eigen-dimension e_i.

   If the dataset contains categorical attributes then the discretization approach is as follows: First, we compute the frequent itemsets generated from all categorical attributes in the original dataset D (for a user-determined value of support). Let us refer to this as set A. We then split the eigen-dimension 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 we are left with a certain number of cutpoints along each eigen-dimension.

   An important question here is how many discrete intervals are needed? The upper bound limit for each dimension is determined by the respective eigenvalue proportions. Intuitively, this makes sense as dimensions capturing less of the variance have fewer intervals. Essentially the vector with lowest eigenvalue is limited to a user-defined number of intervals (unless otherwise noted, we use the value 2 in all our experiments) and all others are correspondingly scaled up. This gives us the upper bound on the number of discrete intervals along each eigen-dimension.

5. **Determining Impact of Eigenvectors on Original Dimensions**: The next step is to determine which eigenvectors are most influenced by which original dimensions or vice-versa. To accomplish this task each original dimension *j* is associated with exactly one of the eigenvectors, say e_i. This association is established by finding the contribution of *j* on each of the eigenvectors ($e_1 \ldots e_n$) and choosing the maximum. The contributions are obtained directly from computing the angle formed by the unit eigenvector and the unit vector along the original dimension. Once this mapping has been established we can re-project the appropriate cut points. Note that multiple original dimensions can be associated to one eigenvector but only one eigenvector can be associated with one original dimension.

6. **Re-projecting Eigen-cutpoints to Original Dimensions**:

   We consider two strategies in our work. To explain our approaches for re-projection, let us assume without loss of generality that the j^{th} original dimension is associated with eigenvector e_i.

   (a) **K-NN method**

   To project the cut-point $c^1_{e_i}$ onto the original dimension *j* using this method, we first find the **K** nearest neighbor intercepts on the eigenvector $\overline{e_i}$ closest to $c^1_{e_i}$. The original points $p_1 \ldots p_k$, representing each of the K-nearest neighbors, as well as $p_{c_{e_i}}$, representing the cut point $c^1_{e_i}$, are computed (as shown in Figure 2a). We then compute the mean (or alternately median) value of the jth dimension for each of these points: $p_1 \ldots p_k$ and $p_{c_{e_i}}$. This mean value represents the corresponding cutpoint along the original dimension j (as shown in Figure 2a).

   (b) **Direct projection**

   The other approach we consider is *direct projection*. To project the cut points $c^1_{e_i} \ldots c^n_{e_i}$ on *j* original dimensions using this method, we need to find the angle between eigenvector e_i and *j* original dimension. The process is shown in Fig 2b. 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 j^{th} dimension.

   Now the cut points $c^1_{e_i} \ldots c^n_{e_i}$ can be projected to original dimension (j) by multiplying it with

$cos(\theta_{ij})$. The same process is applied for all cut points.

Regardless of which method is adopted, if eigenvector e_i is associated with more than one original dimension (especially common in high dimensional datasets), the cut points along that eigenvector e_i are projected back on all associated original dimensions *enabling the discretization method to preserve the inherent correlation in the data*.

7. **Post processing**: The re-projection will give us the intervals on original dimensions. However we might get some interval such that an insignificant number of real data points fall in that interval. We remove (i.e. we merge them with contiguous ones) such intervals according to a user-defined threshold $minpts$. Please note that intervals might be very close to each other but still may exhibit different properties than other intervals, so our criteria of merging intervals is not based on the width of the interval. Rather it is based on number of data points in that interval. This step is particularly useful when the method is used as a preprocessing step for association rule mining as the rules in the small intervals will be very hard to find because of very low support.

3.5 Extension: Handling Missing Data

Incomplete data sets have become almost ubiquitous in a wide variety of application domains, e.g., climate, image, sensor and medical data sets. The incompleteness in these data sets may arise from a number of factors. In some cases, it may be a reflection of certain measurements not being available at the time. In others, the information may be lost due to partial system failure. Or it may simply be a result of users being unwilling to specify attributes due to privacy concerns. Given the ubiquitous prevalence of this problem, it is important to identify whether our algorithm can adapt to missing data.

Incomplete datasets *seemingly* pose the following problems for our discretization method. First, if values for continuous attributes are missing, then it affects the first part 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 Aggarwal and Parthasarathy [**?**] show that estimating the projections of records with missing values along the principle components is more accurate than direct imputation, especially when large parts of the dataset are missing[**?**]. This fits in very nicely with the first three steps of our algorithm presented in the previous section enabling us to handle missing continuous attributes effectively.

Second, if categorical attributes are missing, then it can affect step 4 of our algorithm. While 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 interesting patterns. We demonstrate the general-purpose utility of the proposed work as a preprocessing step for data mining tasks like association rule mining and classification. We also demonstrate the compressibility achieved by our approach and the fact that it readily adapts to datasets with missing information.

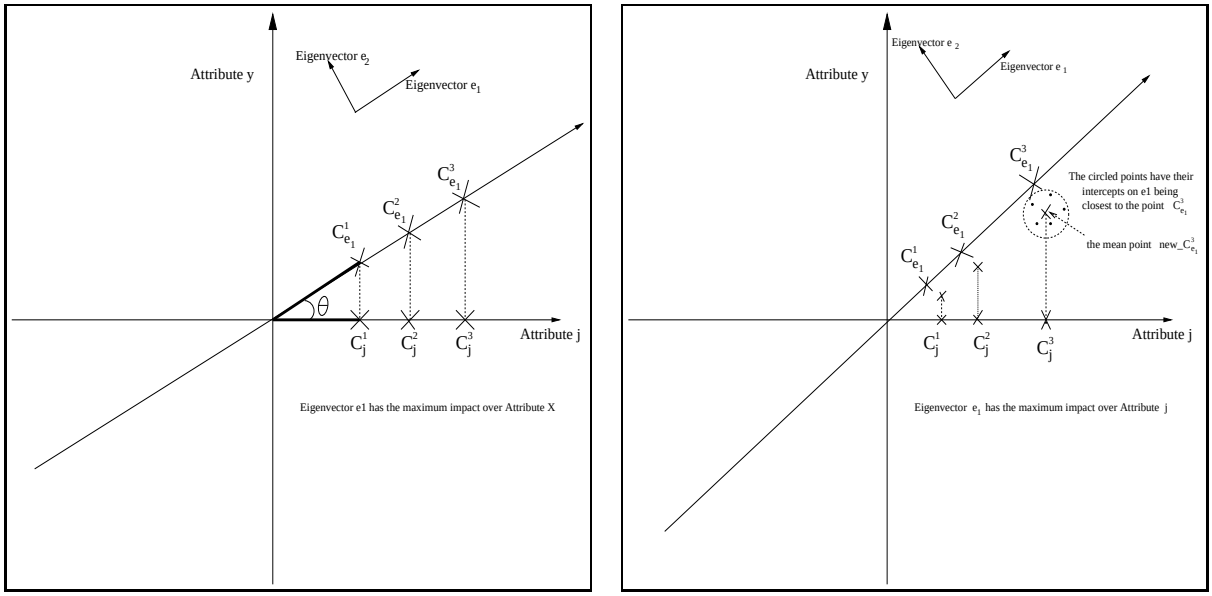
4.1 Experimental Setting

In Table 1 we describe the datasets on which we evaluate the proposed algorithms. The table summarizes information about number of records, number of continuous attributes and number of discrete attributes in the datasets. Two of the datasets have high dimensionality, several have both continuous and categorical attributes. In terms of algorithmic settings, for our K-NN approach the value of K we select for all experiments is 4 (i.e. 4-nearest neighbors and the point projecting onto the cut point itself are used to determine the cut point along the original dimension(s)). Our default similarity metric threshold (for merging intervals) is 0.8 ($\alpha = 0$). All experiments were run on a Pentium III 1GHZ processor with 512MB memory.

We first evaluate the impact of normalization on our proposed method.

4.2 Importance of Normalization in Direct Projection

Normalization of the data is a very important step in our direct projection algorithm, especially in situations where the attribute ranges are highly unbalanced. In such cases the re-projection step tends to be heavily influenced by attributes with large variance (range) resulting in near singular intervals for attributes with relatively small ranges. Table 2 highlights this pathological behavior on the cut points



(a) (b)

Figure 2. (a) Direct Projection (b) K-NN

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
Credit	690	14	6
Credit2	1000	20	7

Table 1. Datasets Used in Evaluation

we obtained with and without normalization on the Age attribute of the adult dataset. Specifically, observe the numerous cut points hovering around 35 in the case when normalization is not performed (number of cut points need not be the same in both cases). Normalization is not essential with the k-NN projection algorithm as there we are using the actual age values.

4.3 Qualitative Results based on Association Rules

In this section we focus on the discretization of the Adult dataset (containing both categorical and continuous attributes) as a preprocessing step for obtaining association rules and compare it with published work on multi-variate discretization (MVD)[2].

Due to the correlation preserving nature of our approach we strongly believe that the intervals our method produces are meaningful and should compare well with MVD. For an interval to be meaningful, we believe that the following two conditions should hold. First, the population in 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 dataset.

- Age: Figure 3 shows the intervals obtained from our approach (both KNN and projection) and the corresponding cut points from MVD on the Age attribute. First at a coarse-grained level we would like to note that the cut points obtained between the different methods are quite similar and quite intuitive. The cut point at 63 corresponds to the retirement age. The intervals 19-22 and 23-24 are quite narrow but represents two different group of people as illustrated below:

  – 3.4% of people aged 19-22 have a bachelors degree as opposed to 22.7% of people aged 23 to 24.

  – 6.1% of people aged 19-22 are married as opposed to 17.0% of people in other group.

Method	cut points
Projection(Normalized)	19,23,25,29,34,37,40,63,85
Projection(w/o Normalization)	35.39,35.39,35.39,35.39,35.40,35.40,35.40,35.41 35.41,35.41,35.41,35.41,35.42,35.42,,35.43,35.45 38.83,38.83

Table 2. Cut points on Age with and without Normalization (Repeated cut points had different decimal values after the second decimal point)

– 18.9% of people aged 19-22 work in service group as opposed to 12.2% people aged 23-24.

MVD also obtained similar cut points. However we had an extra cut point at age 37 giving us intervals 34-37 and 38-40. MVD combines them in one interval 33-41. At first glance these intervals do not seem meaningful since usually there is not much difference in education level and job profiles of people in these groups. However, upon closer inspection we find that 26% of people in the 34-37 interval are *Never Married*. This percentage drops to 13% in the interval 38-40.

MVD's last cut point was 62 which implies that after 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. The male/female demographics in this last interval is quite significant and can be traced to the well known maxim: *'The average life expectancy of females is more than males.'*

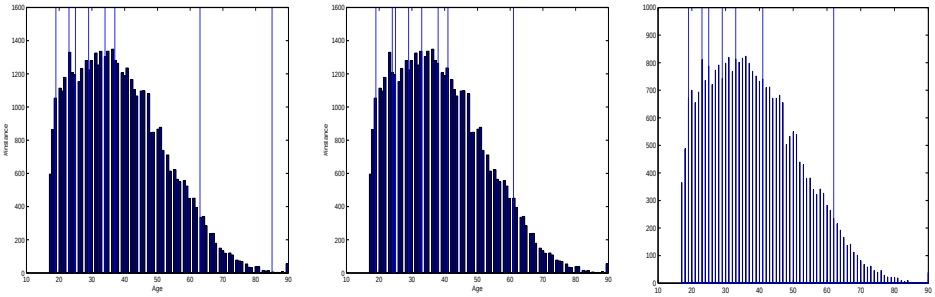
- Capital Gain - The cut points obtained by the three methods are comparable (shown in Fig 4). The cut point from the **projection method** is \$12745. MVD also had one cut point at \$5178. Both these methods separate out people with high gains from people who make little or no gains to moderate gains. Using KNN we were able to get even better cut points. It divided the entire range into 3 intervals, i.e., $< \$7298$ (low capital gain) which has 1981 people, *(\$7299,\$9998)* (moderate gain) having 920 people and $> \$9999$ (high gain) having 1134 people.

- Capital Loss - From Figure 5 we see that MVD and our approaches give the same intervals. Records are separated based on whether loss was declared. We were able to find the rule
CapitalLoss $\geq \$377 \Rightarrow salary > \$50K$ (3% support, 49.3% confidence), which was also found by MVD[2].

- Hours per week - Figure 6 shows our cut points for hours/week. 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 obtained the first cut point at 19 hours per week. We are thus able to extract the rule $Hoursperweek \leq 19 \Rightarrow age \leq 20$, 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 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.

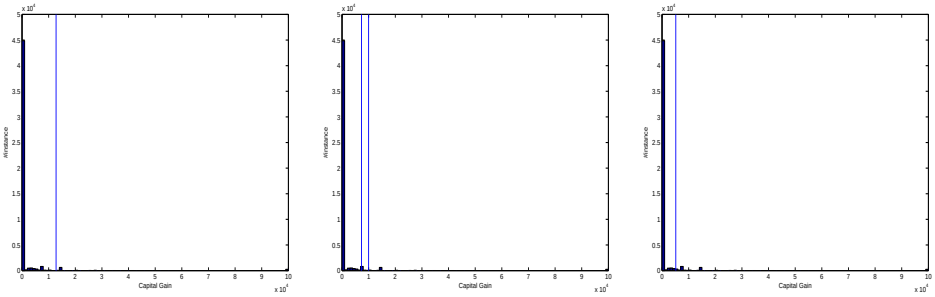
In terms of quantitative experiments we could not really compare with the MVD method as the source/executable code was not available to us. We will point out that for the larger of the datasets (both high dimensional and datasets with larger number of records) our approach took on the order of a few seconds. The order complexity of our method is bounded by the order complexity of each step. The steps that dominate the execution time are the ones to compute the correlation $O(d^2 \cdot N)$, the time to compute the eigenvectors $O(d^3)$ where d is the number of dimensions and N is the number of records in the dataset. The order complexity for the rest of the steps is dependent on the number of cut points, and if we use the K-NN strategy the value of K being used. The other steps are at most linear in the number of dimensions.

In comparing our method(s) with MVD we found that by and large we find intuitive cut points (like MVD), however we do so at the fraction of the cost. Our benefits over MVD in terms of execution time can be traced to the fact that we use PCA to reduce the dimensionality of the problem and we compute one set of cut points and project the resulting cut points onto the original dimension(s) simultaneously. We would also like to point out that our method of discretization can be used to reduce the storage costs of the data being stored and that it is not tied in to a particular method, i.e. as we show it is equally effective as a preprocessing step for classification as it is for a preprocessing step for association rule mining.



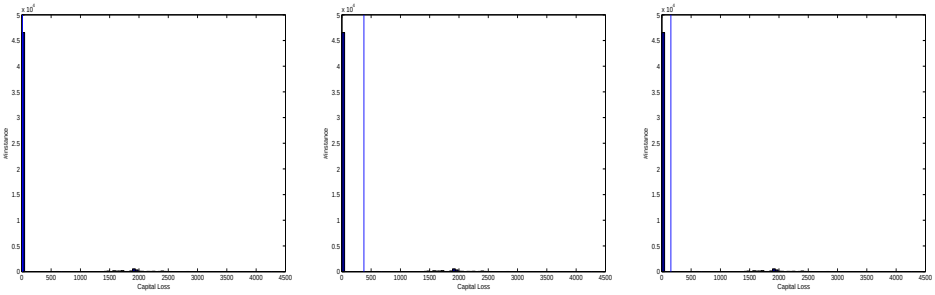
(a) (b) (c)

Figure 3. Cutpoints on Age: (a) Projection (b) KNN (c) MVD



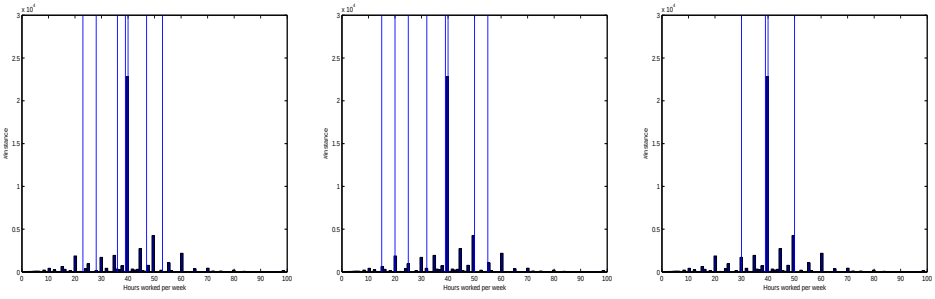
(a) (b) (c)

Figure 4. Cutpoints on Gain: (a) Projection (b) KNN (c) MVD



(a) (b) (c)

Figure 5. Cutpoints on Loss: (a) Projection (b) KNN (c) MVD



(a) (b) (c)

Figure 6. Cutpoints on Hours: (a) Projection (b) KNN (c) MVD

4.4 Qualitative Results using Classification

In this section we evaluate the viability of our approach as a preprocessing step for classification. For both the direct projection and KNN algorithm we bootstrap the results with the C4.5 decision tree classifier. We compare our approach against various classifiers supported by the Weka data mining toolkit[1]. Note that in many cases these classifiers use a supervised discretization algorithm (taking into account class label distributions) as a preprocessing step. Our algorithms are unsupervised in nature (i.e. they do not take into account class-label distributions). In our method once the discretization has been performed we append the class labels to the discretized datasets and run C4.5. All results use 10-fold cross-validation.

Table 3 shows the error rate of our approaches (last two columns) as compared to seven different classifiers (first seven columns). First, on viewing the results it is clear that our methods coupled with C4.5 often outperforms the other approaches (including C4.5 with supervised discretization which had the same exact settings for C4.5 as our approaches) and especially so on high dimensional datasets (Musk(1) and Musk(2)). The direct method performs marginally better than the KNN method but this is not statistically significant. The Bupa dataset is the only dataset on which our method performs slightly worse and this may be attributed to the fact that the correlation structure of this dataset is weak[**?**]. Second, we do better in-spite of the fact that our approach is unsupervised and many of the above classifiers use the class-label distributions to discretize continuous attributes, thus validating our claim that the inherent correlations that are preserved by our methods are useful for classification purposes. Finally, our approach also lends itself to faster classifier construction times. Decision trees built on top of the discretized datasets were constructed around 10-20% faster on the average. This did not represent a significant savings in execution times for our datasets (since they are quite small) but can be quite significant in larger datasets.

4.5 Experiments with Missing Data

The first experiment we ran compared the impact of missing data on the classification results on three of our datasets. We randomly eliminated a certain percentage of the dataset and then adopted the approach described in Section 3.5. Table 4 documents these results. Clearly as the percentage of data missing is increased the classification error increases. However this error differential is not too bad even when 30% of the data is missing. When 10% of the data is missing the differences are relatively insignificant indicating that the discretization approach can tolerate missing data quite well.

In the second experiment we randomly eliminated a percentage of the categorical components of our dataset and then attempted to predict the missing values. Our prediction strategy involved computing the discretization intervals in our projected space and then computing the frequent pattern rules that dominate each interval and using these rules to predict the missing values in a manner akin to CBA[**?**]. Prediction rules were ordered based on length, confidence and support and then applied [**?**]. We compared this strategy, referred to as PCA-based, against three strawman methods.

- **Dominant Value** - Under this scheme we find the most dominant value for each attribute in an interval. All the missing values are then replaced by corresponding dominant values. The dominant value is for an attribute is value which occurs most number of times.
- **Without Continuous Attributes** - In this scheme the predictions are based solely on the other categorical attributes in the dataset.
- **Random** - Missing values are predicted by randomly picking one of the possible value for a specific attribute. Results are averaged over 10 different runs.

Table 5 shows the accuracy of all four schemes on different datasets. The proposed PCA-based scheme has the highest accuracy and is remarkably accurate when 10% of the data is missing especially on the adult dataset. This further asserts the fact that our *correlation preserving discretization strategy indeed provides us with meaningful intervals that are extremely useful in predicting missing data values*. The dominant and random strategies are not very effective but the accuracy is worst in case where we do not leverage the continuous attributes. We plan to compare the PCA-based imputation scheme with an EM-based imputation method [**?**], as part of future work.

4.6 Compression of datasets

In this section we evaluate the compressibility achievable by discretization, a useful utility in the case of large datasets or warehousing environments. Note that here we do not consider classic compression utilities such as *gzip* etc., 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

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

Dataset	C4.5	IBK	PART	Bayes	ONER	Kernel-based	SMO	Projection	KNN
Adult	**15.7**	20.35		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
Credit	15	14.9	17	23.3.	15.5	17.4	15	**14.8**	**14.9**

Table 3. Classification Results (error comparison - best results in bold)

Dataset	Original	10% missing	20% missing	30% missing
Adult	15.7%	16%	17%	19%
Credit1	15%	17%	18.8%	18.9%
Credit2	25%	28%	30%	32%

Table 4. Classification Error on Missing Data

them we can easily reduce the storage requirements for each such attribute. The maximum number of bits required for an attribute A is a log function of the range.of values permissible. Table 6 shows the results for compression on various datasets. As we can see from the results in most datasets we achieve a compression factor of around 3 and in some cases the results are better.

5 Conclusions

In this article we propose correlation preserving discretization, an efficient method that can effectively discretize continuous attributes even in high dimensional datasets, by accounting for the inherent correlations in the data in a multi-variate context. The algorithm uses *both* the distribution of categorical and continuous attributes and the underlying correlation structure in the dataset to obtain the discrete intervals. The approach also ensures that *all attributes are used simultaneously* for deciding the cut points rather than one attribute at a time. We demonstrate the effectiveness of the approach on real datasets, including high dimensional datasets, both as a preprocessing step to classification as well as for frequent itemset mining. We also propose an extension to the algorithm so that it can deal with missing values effectively and validate this aspect as well. We show that the resulting discretized datasets can be used as a means to store data in a compressed fashion ready to use for different mining tasks. We also show that the intervals obtained are meaningful,intuitive and can uncover the hidden patterns in data.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *20th VLDB Conf.*, September 1994.

[2] Stephen D. Bay. Multivariate discretization for set mining. *Knowledge and Information Systems*, 3(4):491–512, 2001.

[3] J. Catlett. Changing continuous attributes into ordered discrete attributes. In *Proceedings of European Working Session on Learning*, 1991.

[4] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning*, 1995.

[5] Usama. M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *14th Joint Conference on AI*, 1993.

[6] Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Mining optimized association rules for numeric attributes. In *Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 182–191, 1996.

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

[8] Randy Kerber. Chimerge: Discretizaion of numeric attributes. In *National Conference on AI*, 1991.

	Adult			**Credit1**			**Credit2**		
	10%	20%	30%	10%	20%	30%	10%	20%	30%
PCA-based	75%	63%	62%	58%	53%	55%	65%	60%	60%
Dominant	47%	46%	48%	40%	30%	35%	37%	36%	33%
W/O Continuous	22%	15%	29%	15%	10%	10%	20%	18%	11%
Random	37%	40%	34%	40%	33%	35%	39%	36%	33%

Table 5. Missing Value Prediction Results

Datasets	Original	Byte Compressed and Discretized	Compression Factor
BUPA	3795	1035	3.67
ADULT	537350	195400	2.75
MUSK1	85680	29693	2.89
CANCER	6830	3415	2.00
MUSK2	1319800	422336	3.13
CREDIT1	28735	3450	8.33
CREDIT2	79793	16000	4.99

Table 6. Compression Results

[9] Marcus-Christopher Ludl and Gerhard Widmer. Relative unsupervised discretization for association rule mining. In *Proceedings of the 4th European Conference of Principles of Data Mining and Knowledge Discovery*, sep 2000.

[10] Wolfgang Maass. Efficient agnostic PAC-learning with simple hypotheses. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 1994.

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

[12] Rajeev Rastogi and Kyuseok Shim. Mining optimized support rules for numeric attributes. *Information Systems*, 26(6):425–444, 2001.

[13] Rajeev Rastogi and Kyuseok Shim. Mining optimized association rules with categorical and numeric attributes. *Knowledge and Data Engineering*, 14(1):29–50, 2002.

[14] Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 1–12, Montreal, Quebec, Canada, jun 1996.

[15] R. Subramonian, R. Venkata, and J. Chen. A visual interactive framework for attribute discretization. In *Proceedings of KDD'97*, 1997.