Correlation Preserving Discretization *

Sameep Mehta, Srinivasan Parthasarathy and Hui Yang
Department of Computer Science and Engineering, The Ohio State University
Contact: (mehtas,srini,yanghu)@cse.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: Unsupervised Discretization, Missing Data

1 Introduction

Discretization is a widely used data preprocessing primitive. 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". Dougherty *et al* [3] present an excellent classification of current methods in discretization. A majority of the discretization methods in the literature [2, 4, 6, 10, 13, 14] are supervised in nature and are geared towards minimizing error in classification algorithms. Such methods are not general-purpose and cannot, for instance, be used as a preprocessing step for frequent itemset algorithms.

In this article we propose a general-purpose unsupervised algorithm for discretization based on the *correlation structure* inherent in the dataset. Closely related to our work is the recent work by Bay[1] and Ludl and Widmer[9]. Bay proposed an approach for discretization that also considers the interactions among all attributes. The main limitation of this approach is that it can be computationally expensive, and impractically so for high dimensional and large datasets. Ludl and Widmer [9] propose a similar approach however the interactions amongst attributes considered in their work is only pair-wise and piecemeal.

In this work 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 pairwise or one attribute at a time. An additional advantage is that the approach extends naturally to datasets with missing data.

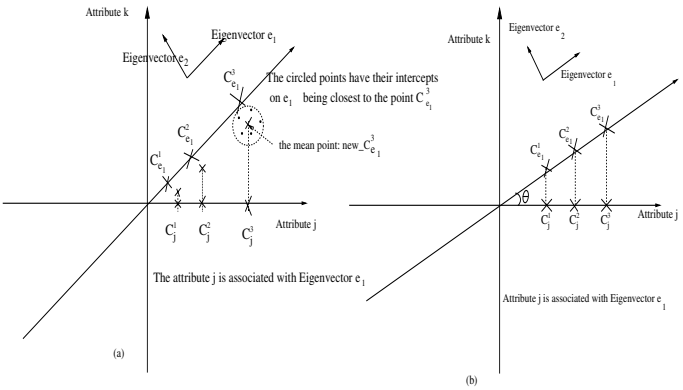

Figure 1. (a) K-NN (b)Direct Projection

We demonstrate the efficacy of the above algorithms as a preprocessing step for classical data mining algorithms such as frequent itemset mining and classification. Extensive experimental results on real and synthetic datasets demonstrate the discovery of meaningful intervals for continuous attributes and accuracy in prediction of missing values.

2 Algorithm

Our algorithm is composed of the following steps:

1. Normalization and Mean Centralization: The first step involves normalizing all the continuous attributes and mean centralizing the data.
Rationale: This is a standard preprocessing step [12].

2. Eigenvector Computation: We next compute the correlation matrix M from the data. We then compute the eigenvectors $\overrightarrow{e_1}, \ldots, \overrightarrow{e_d}$ from M. To find these eigenvectors, we rely on the popular Householder reduction to tridiagonal form and then apply the QL transform [5]. Once these eigenvectors have been determined, we retain only those that preserve 90% of the variance in the data.
Rationale: Retaining only those eigen vectors that are acounted for most of variance enables us to keep most of the correlations present among the continuous attributes in the dataset. Dimensionality reduction facilitates scalability.

*This work is funded by NSF grants ITR-NGS ACI-0326386, CAREER IIS-0347662 and SOFTWARE ACI-0234273.

3. Data Projection onto Eigen Space: Next, we project each data point in the original dataset D onto the eigen space formed by the vectors retained from the previous step. **Rationale:** To take advantage of dimensionality reduction.

4. Discretization in Eigen Space: We discretize each of the dimensions in the eigen space. Our approach to discretization depends on whether we have categorical attributes or not. If there are no categorical attributes, we apply a simple distance-based clustering along each dimension to derive a set of cut-points. If the dataset contains categorical attributes, our approach is as follows. First, we compute the frequent itemsets generated from all categorical attributes in the original dataset D (for a user-determined support value). Let us refer to this as set A. We then split the eigen dimension $\overrightarrow{e_i}$ into equal-frequency intervals 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 B and C as follows:

For an element $x \in B$ (respectively in C), 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:

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

where α is a scaling parameter. If the similarity exceeds a user defined threshold the contiguous intervals are merged. Again, we are left with a set of cut-points along each eigen dimension.

Rationale and Key Intuitions: First, there is no need to consider the influence of the other components in the eigen space since the second order correlations are zero after the PCA reduction, Second, the use of frequent itemsets (as a constraint measure) ensures that correlations w.r.t. the categorical attributes are captured effectively.

5. Correlating Original Dimensions with Eigenvectors: The next step is to determine which original dimensions correlate most with which eigenvectors. This can be computed by finding the contribution of dimension *j* on each of the eigenvectors ($\overrightarrow{e_1}, \ldots, \overrightarrow{e_n}$), scaled by the corresponding eigenvalue and picking the maximum [7].

Rationale: This step is analogous to computing factor loadings in factor analysis. This step ensures that the set of original dimensions associated with a single eigenvector have corresponding discrete intervals, which ensures that these original dimensions remain correlated with one another.

6. Reprojecting Eigen cutpoints to Original Dimensions: We consider two strategies which are explained below.

a. K-NN method: To project the cut-point $c^1_{e_i}$ onto the original dimension *j*, we first find the *k* nearest neighbor intercepts of $c^1_{e_i}$ on the eigenvector $\overrightarrow{e_i}$. The original points $p_1, \ldots, p_k$, representing each of the k nearest neighbors, as well as $c^1_{e_i}$, are obtained (as shown in Figure 1a). We then compute the mean (or median) value of of these points. This mean (or median) value represents the corresponding cut-point along the original dimension j.

b. Direct projection: In this approach, to project the cut points $c^1_{e_i}, \ldots, c^n_{e_i}$ onto the original dimension j, we need to find the angle θ_{ij} between eigenvector $\overrightarrow{e_i}$ and dimension j. The process is shown in Figure 1b. Now the cut-points $c^1_{e_i} \ldots c^n_{e_i}$ can be projected to the original dimension j by multiplying it with $cos(\theta_{ij})$.

The re-projection will give us the intervals on the original dimensions. However, it is possible that we get some intervals that involves an insignificant number of real data points. In this case, we merge them with its adjacent intervals according to a user-defined threshold.

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

Handling Missing Data: Incomplete datasets *seemingly* pose the following problems for our discretization method. First, if values for continuous attributes are missing, steps 1 through 3, of our algorithm will be affected. Fortunately, our PCA-based discretization approach enables us to handle missing values for continuous attributes effectively by adopting recent work by Parthasarathy and Aggarwal [12]. Second, if categorical attributes are missing they can affect step 4 of our algorithm. However, our premise is that, when entries are missing at random, 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 mostly unaffected. More details on the algorithms can be found in our technical report[11].

3 Experimental Results and Analysis

In Table 1 we describe the datasets[1] on which we evaluate the proposed algorithms. In terms of algorithmic settings, for our K-NN approach the value of K we select for all experiments is 4. Our default similarity metric threshold (for merging intervals) is 0.8 ($\alpha = 0$).

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

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 multivariate discretization.

[1] All the datasets are obtained from UCI Data Repository

Specifically, we compare with ME-MDL (supervised) using salary as the class attribute and MVD (unsupervised), limiting ourselves only to those attributes discussed by Bay[1].

First, upon glancing at the results in Table 2 it is clear that KNN, MVD and Projection all outperform ME-MDL in terms of identifying meaningful intervals (also reported by Bay[1]). Moreover, ME-MDL seems to favor more number of cut-points many of which have little or no latent information. For the rest of the discussion we limit ourselves to comparing the first three methods.

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

Table 2. Cut-points from different methods on *Adult*

For the *age* attribute, 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. Similarly for the *capital loss* attribute, all methods return a single cutpoint, and the cutpoint returned by both Projection and MVD are almost identical. For the *capital gain* attribute, there is some difference in terms of the cutpoint values returned by all three methods. Using KNN we were able to get even better cut-points than the other two methods. It divides the entire range into three intervals, i.e., $(\$0, \$7298)$ low gain, which has 1981 people, *($7299,$9998)* moderate gain, having 920 people and $(\$9999, MAX)$ high gain, having 1134 people. For the above three attributes we get many of the same association rules that Bay reports in his paper. Details are provided in our technical report[11].

The *hours/week* attribute is one where we get significantly different cut-points from MVD. 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/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 refers to blue-collar workers. More differences among the different methods is detailed in our technical report[11].

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 large datasets (both in terms of dimensionality and number of records), our approaches take on the order of a few seconds in running time. Our benefits over MVD in terms of execution time stem from the fact that we use PCA to reduce the dimensionality of the problem and the fact that we compute one set of cut-points on each principal component and project the resulting cut-points onto the original dimension(s) simultaneously.

Qualitative Results Using 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[2]. We note that most of these classifiers use a supervised discretization algorithm whereas our approach is unsupervised. For evaluating our approaches, once the discretization has been performed, we append the class labels back to the discretized datasets and run C4.5. All results use 10-fold cross-validation.

Table 3 shows the mis-classification rates of our approaches (last two columns) as compared to seven other different classifiers (first seven columns). First, on viewing the results it is clear that our methods coupled with C4.5 often outperform the other approaches (including C4.5 itself) and especially so on high dimensional datasets (Musk(1) and Musk(2)). The Bupa dataset is the only one on which our methods perform marginally worse, and this may be attributed to the fact that the correlation structure of this dataset is weak[12].

Experiments with Missing Data: Our first experiment compares the impact of missing data on the classification results on three of the datasets. We randomly eliminated a certain percentage of the data and then adopted the approach described in Section 2. Table 4 documents these results. One can observe that the classification error is not affected much by the missing data, even if there is 30% of the data missing. This indicates that our discretization approach can tolerate missing data quite well.

In the second experiment, we randomly eliminated a percentage of the categorical components from the dataset and

[2]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

then predicted the missing values using the discretized intervals. For each interval, we identify frequent association rules. We next use these rules to predict the missing values [8, 15]. We compared this strategy, referred as PCA-based, against three strawman methods: **(1)Dominant Value:** Under this scheme, the missing value is predicted by the most dominant value for each attribute in an interval. **(2)Discretization w/o PCA:** Under this scheme we perform equi-width discretization, which is unsupervised and does not count the correlation as against our approach. **(3)Random:** Missing values are predicted by randomly picking a possible value of a specific attribute.

	Adult			**Credit1**			**Credit2**		
Missing(%)	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 PCA	22	15	29	15	10	10	20	18	11
Random	37	40	34	40	33	35	39	36	33

Table 5. Missing Value Prediction Accuracy (%)

Table 5 shows the accuracy of all four schemes on different datasets, in which all results are averaged over 10 different runs. It is clear that the PCA-based scheme has the highest accuracy among all four. Wherease the w/o PCA scheme has the lowest accuracy, which might be caused by not considering the inter-attribute correlation. Such a difference also validates the importance of preserving correlation when discretizing data of high dimensionality.

Compression of Datasets: In this section we evaluate the compressibility that can be achieved by discretization. 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 6 shows the results of compression on various datasets. As we can see from the results, on most datasets we achieve a compression factor around 3, and in some cases the results are even better.

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
Shuttle	1153518	478500	2.4

Table 6. Compression Results

4 Conclusions and Future Work

In this article we propose correlation preserving discretization, an efficient method that can effectively discretize continuous attributes even in high dimensional datasets. The approach 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, as a preprocessing step for classification as well as for frequent association mining. We show that the resulting datasets can be easily used to store data in a compressed fashion ready to use for different data mining tasks. We also propose an extension to the algorithm so that it can deal with missing values effectively and validate this aspect as well.

References

[1] Stephen D. Bay. Multivariate discretization for set mining. *Knowledge and Information Systems*, 3(4):491–512, 2001.
[2] J. Catlett. Changing continuous attributes into ordered discrete attributes. In *Proceedings of European Working Session on Learning*, 1991.
[3] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *ICML*, 1995.
[4] Usama. M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI*, 1993.
[5] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.
[6] Randy Kerber. Chimerge: Discretizaion of numeric attributes. In *National Conference on AI*, 1991.
[7] Jae-On Kim and Charles W. Mueller. *Factor Analysis : Statistical Methods and Practical Issues*. SAGE Publications, 1978.
[8] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *KDD*, pages 80–86, 1998.
[9] Marcus-Christopher Ludl and Gerhard Widmer. Relative unsupervised discretization for association rule mining. In *PKDD*, 2000.
[10] Wolfgang Maass. Efficient agnostic PAC-learning with simple hypotheses. In *COLT*, 1994.
[11] Sameep Mehta, Srinivasan Parthasarathy, and Hui Yang. Correlation preserving discretization. In *OSU-CISRC-12/03-TR69*, December 2003.
[12] Srinivasan Parthasarathy and Charu C. Aggarwal. On the use of conceptual reconstruction for mining massively incomplete data sets. In *TKDE*, 2003.
[13] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
[14] R. Subramonian, R. Venkata, and J. Chen. A visual interactive framework for attribute discretization. In *Proceedings of KDD'97*, 1997.
[15] Hui Yang and Srinivasan Parthasarathy. On the use of constrained association rules for web log mining. In *Proceedings of WEBKDD 2002*, 2002.