Syracuse University
SURFACE

Electrical Engineering and Computer Science | L.C. Smith College of Engineering and Computer Science

11-2-2009

A Clustering based Discretization for Supervised Learning

Ankit Gupta

Kishan Mehrotra
Syracuse University, mehrotra@syr.edu

Chilukuri K. Mohan
Syracuse University, ckmohan@syr.edu

Follow this and additional works at: http://surface.syr.edu/eecs

Part of the Computer and Systems Architecture Commons

Recommended Citation
Gupta, Ankit; Mehrotra, Kishan; and Mohan, Chilukuri K., "A Clustering based Discretization for Supervised Learning" (2009). *Electrical Engineering and Computer Science.* Paper 3.
http://surface.syr.edu/eecs/3

This Report is brought to you for free and open access by the L.C. Smith College of Engineering and Computer Science at SURFACE. It has been accepted for inclusion in Electrical Engineering and Computer Science by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

Department of Electrical Engineering and Computer Science

Technical Report

SYR-EECS-2009-03 Nov. 2, 2009

A Clustering based Discretization for Supervised Learning

Ankit Gupta ankitgupta.iitk@gmail.com
Kishan G. Mehrotra mehrotra@syr.edu
Chilukuri Mohan mohan@syr.edu

ABSTRACT: We address the problem of discretization of continuous variables for machine learning classification algorithms. Existing procedures do not use interdependence between the variables towards this goal. Our proposed method uses clustering to exploit such interdependence. Numerical results show that this improves the classification performance in almost all cases. Even if an existing algorithm can successfully operate with continuous variables, better performance is obtained if variables are first discretized. An additional advantage of discretization is that it reduces the overall time-complexity.

KEYWORDS: Discretization, Clustering, Binning, Supervised Learning

Syracuse University - Department of EECS,
4-206 CST, Syracuse, NY 13244
(P) 315.443.2652 (F) 315.443.2583
http://eecs.syr.edu

A Clustering based Discretization for Supervised Learning

Ankit Gupta[a], Kishan G. Mehrotra[*,b], Chilukuri Mohan[b]

[a] *Department of Electrical Engineering, Indian Institute Of Technology, Kanpur, India 208016*

[b] *Department of Electrical Engineering & Computer Science, 4-106 Center for Science & Technology, Syracuse University, Syracuse, NY 13244-4100, USA*

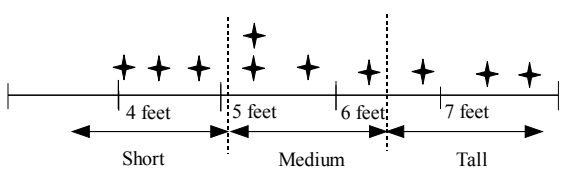
Abstract

We address the problem of discretization of continuous variables for machine learning classification algorithms. Existing procedures do not use interdependence between the variables towards this goal. Our proposed method uses clustering to exploit such interdependence. Numerical results show that this improves the classification performance in almost all cases. Even if an existing algorithm can successfully operate with continuous variables, better performance is obtained if variables are first discretized. An additional advantage of discretization is that it reduces the overall time-complexity.

Key words:
Discretization, Clustering, Binning, Supervised Learning

1. Introduction

Discretization is the mapping of a continuous variable into discrete space, and is frequently performed in the analysis of sociological data to assist comprehension, grouping together multiple values of a continuous attribute, and partitioning the continuous domain into non-overlapping intervals. An example is the transformation of a continuous variable such as height (of a human) from a numerical measure (for example, 6'3") into tall / short / medium categories.



The ensuing compression and compaction of data facilitates formulation of comprehensible rules such as "short(x) and heavy(x) → overweight(x)". Thus, the central data mining task of extracting small rules from quantifiable data is facilitated by the discretization process applied to continuous variables (e.g., height and weight). Discretization facilitates the application of several machine learning algorithms to problems where attributes are in continuous space.

*Corresponding Author
Email addresses: ankitgupta.iitk@gmail.com (Ankit Gupta), mehrotra@syr.edu (Kishan G. Mehrotra), mohan@syr.edu (Chilukuri Mohan)

In addition, the need for discretization of continuous variables in machine learning arises because some machine leaning algorithms are designed to operate best with discrete variables. There are several ways by which discretization methods can be classified: *Splitting* vs. *Merging*, *Global* vs. *Local*, *Supervised* vs. *Unsupervised*, and *Static* vs. *Dynamic*.

- Splitting method, as exemplified by entropy based partitioning [3], is a top down approach of discretization in which we start with an empty set of cut points and gradually divide the interval and subintervals to obtain the discretization. In contrast, the merging method is a bottom up approach in which we consider all possible cut points and then eliminate these cut points by merging intervals (see *Chi-Merge* [1], *Chi2* [2]).

- Local methods, such as C4.5 [15], produce partitions that are applied to localized regions of instance space. Global methods [6], on the other hand, use the entire instance space and forms a mesh over the entire n-dimensional continuous instance space, where each feature is partitioned into regions independent of other attributes.

- Static discretization methods require some parameter, k, indicating the maximum number of desired intervals in discretizing a feature. These methods include binning, entropy-based partitioning [3, 7, 8] and 1R-algorithm [9] which perform one discretization pass of the data for each feature and determine the value of k for each feature independent of other features. However, *dynamic* methods conduct a search through the space of possible k values for all features simultaneously, thereby capturing interdependencies in feature discretization.

- Unsupervised methods such as *equal width* and *equal frequency* interval binning carry out discretization without the knowledge of class label, whereas the supervised methods [1, 2, 3] utilize the class information to carry out the discretization. The simplest discretization procedure is to divide the range of continuous variable into *equal width* or *equal frequency* intervals using a user defined parameter, k. However the weakness of these unsupervised methods is in cases when observations are not distributed equally. Supervised methods such as Chi-Merge [1], Chi2 [2] use χ^2 statistical measure to merge adjacent intervals. Fayyad & Irani developed a concept of entropy based partitioning in [3]. Dougherty, Kohavi, and Sahami in [4] made comparison of the uniform binning, the discretization presented by Holte [9] and an entropy based method proposed by Fayyad and Irani [3], using two induction algorithms: C4.5 and a Naive-Bayesian classifier and reported that the entropy based disretization was the most promising one. Also, entropy based discretization performs better than the error based discretization as discussed in [5].

The algorithm presented in this paper is a splitting, supervised, global and static method. The underlying idea of our work is to use inter-attribute dependencies contrasting with previous work in which each attribute is discretized independent of the other attributes. On the other hand exploitation of this interdependence is hard to achieve due to the reason that the classical measures of dependence do not apply when one variable is discrete and the other is continuous. Clustering provides a natural criterion for exploiting this dependence.

Mehrotra *et al* [11] observed that a cluster-based classification generally performs better in multi-class problems. Monti and Cooper [10] developed a latent variable model for the unsupervised multivariate discretization in order to account for the interdependencies in the data. In the absence of the class label, the possible interaction of the attribute being discretized with other attributes was utilized to create a pseudo-class and unsupervised discretization problem was converted to a supervised one.

In our algorithm we extend this approach to address supervised discretization tasks. We first cluster the data (using K-means clustering and Shared Nearest Neighbor clustering [16] techniques) to account for the interdependencies among different attributes and use cluster-ids along with the provided class information to discretize the data using Minimum Entropy with Minimum Description Length (ME-MDL) [3] as the stopping criterion.

In section 2 we describe our algorithm and in section 3 we explain the experiments conducted and compare the results of our algorithm with Minimum Entropy-Maximum Description Length (ME-MDL) and finally conclude in section 4.

2. Algorithm

Our algorithm is based on clustering i.e., partitioning data into a set of subsets so that the intra-cluster distances are small and inter-cluster distances are large. The clustering technique does not utilize class identification information, but instances belonging to the same cluster should ideally belong to the same class. However, in real world complex problems a class gets separated into two or more clusters. Therefore, we propose to discretize data with the knowledge of both classes and clusters.

We have used two clustering algorithms – the K-means clustering approach with Euclidean distance metric as the similarity measure and the Shared Nearest Neighbor (SNN) clustering algorithm of Ertoż et al [16] . In the K-means clustering approach the number of clusters are kept *approximately*[1] equal to the number of classes. Each instance is assigned to a particular cluster and this is labeled as the *'pseudo class'*. However, as pointed out in Ertoż et al, the K-means clustering algorithm has difficulty with outliers and performs poorly when the clusters are of different shapes, sizes, and density. In addition in an implementation of the K-means clustering we need to specify the number of clusters. Therefore, as an alternative, we have implemented the SNN clustering algorithm as well.

Clustering is a natural process in which similar objects tend to group together. It provides us with the intrinsic grouping of the unlabeled data. Thus, the cluster-id captures the interdependencies in the data. So the problem has now been changed to classifying data (with both continuous & discrete attributes) having two class features, one provided with data, C and other, pseudo class C'. We then apply a generalized version of minimum entropy method [3] as explained below. The definitions in sections 2.1 and 2.2 have directly been taken from [3].

2.1. Criteria for Discretization

The idea behind entropy based discretization is to search for the partition of the value range of continuous feature so as to minimize the uncertainty of the class variable conditioned on the discretized feature variable. This approach results in two intervals and is applied recursively to each subsequent sub-interval until the stopping criterion (explained in 2.3) is met. Let S be the set of N examples. For each continuous valued attribute A we select the best cut point T_a from its range of values by evaluating every candidate cut point in the range of values. The examples are first sorted by increasing value of attribute A and the mid-point between each successive pair

[1] We have considered larger values for the number of clusters than the number of classes and have observed that the performance is generally robust for SVM and Naive Bayes Classifiers (see Tables 2 and 3 in Section 3). We have evaluated the performance when number of clusters is equal to number of classes + 2, 3, 4 and 5. For the SVM classifier, classification performance decreases as the number of clusters increase, whereas for the Naive Bayes classifier, we found that when # of clusters = # of clusters + 3, performance further improves.

of examples in the sorted sequence is evaluated as a potential cut point. Thus, for each continuous valued attribute, $(N-1)$ evaluations will take place (assuming that examples do not have identical attribute values). For each evaluation of candidate cut point T, the data is partitioned into two sets and the class entropy of the resulting partition is computed (corresponding to C as well as C'). The discretization procedure is performed locally at each step.

Let there be k classes in class feature C and k^* classes in class feature C'. Let $\mathrm{P}(C_i, S)$ be the proportion of examples in S that belong to C_i. Similarly, let $\mathrm{P}(C'_i, S)$ be the proportion of classes in C'_i. Then the class entropy (measure of the amount of information needed, in bits, to specify the classes in S) is defined as:

$$\begin{aligned}
\mathrm{Ent}(S, C) &= \sum_{i=1}^{k} \mathrm{P}(C_i, S) \log_2[\mathrm{P}(C_i, S)] \\
\mathrm{Ent}(S, C') &= \sum_{i=1}^{k^*} \mathrm{P}(C'_i, S) \log_2[\mathrm{P}(C'_i, S)]
\end{aligned}$$

Definition 1: For an example set S, an attribute A, and a cut point T: Let S_1 be the subset of examples in S with A-values $\leq T$ and $S_2 = S - S_1$. The information entropy induced by partition T on class feature C is defined as $\mathrm{E}(A, T; S, C)$:

$$\mathrm{E}(A, T; S, C) = \frac{|S_1|}{|S|}\mathrm{Ent}(S_1, C) + \frac{|S_2|}{|S|}\mathrm{Ent}(S_2, C)$$

and similarly on class feature C', $\mathrm{E}(A, T; S, C')$ is defined as:

$$\mathrm{E}(A, T; S, C') = \frac{|S_1|}{|S|}\mathrm{Ent}(S_1, C') + \frac{|S_2|}{|S|}\mathrm{Ent}(S_2, C')$$

A binary discretization for A is determined by selecting the cut point T_a for which the linear combination[2]of the above two terms, $\xi(A, T; S, \alpha, \beta)$, is minimum among all the candidate cut points, where α and β are weights assigned to class C and C' respectively.

$$\begin{aligned}
\xi(A, T; S, \alpha, \beta) &= \alpha[\mathrm{E}(A, T; S, C)] + \beta[\mathrm{E}(A, T; S, C')] \\
\xi(A, T_a; S) &= \arg\min_T[\xi(A, T; S, \alpha, \beta)]
\end{aligned}$$

The entropy based discretization in [3] can be considered to be a special case , with $\alpha = 1$ and $\beta = 0$. Unsupervised discretization (Monty & Copper[10]) is a special case with $\alpha = 0$ and $\beta = 1$, as no information of class label is provided. In our algorithm, we have assigned equal weightage to both classes and chosen $\alpha = \beta = 0.5$.

2.2. Discussion of Cut Points

For each attribute the number of cut points being considered is $(N-1)$. This is computationally very expensive as the machine learning algorithms are often applied to training data, where N is

[2]The reason behind considering a linear combination is that the two class features, C and C' are independent of each other

very large. Therefore, only a subset of the range of values are considered as potential cut points and these turn out to be the *boundary points* (defined below) of the class.

Definition 2: For attribute A, a value T is a boundary point iff in the sequence of examples sorted by the value of A, there exist two examples $e_1, e_2 \in S$, having different classes, such that $A(e_1) < T < A(e_2)$; and there exists no other example $e' \in S$ such that $A(e_1) < A(e') < A(e_2)$.

It is shown in [12] that the value T_a for attribute A that minimizes the combined class entropy $\xi(A, T; S, \alpha, \beta)$ for a training set S must always be a value between two examples of different classes in the sequence of sorted examples. Therefore, the potential cut points reduce from $(N-1)$ to just the boundary points of each class. In our algorithm, the set of potential cut points is the *union* of the boundary points of class C and pseudo class C'.

2.3. Stopping Criterion

The stopping criterion is based on the Minimum Description Length (MDL) principle. The minimum description length of an object is defined to be the minimum number of bits required to uniquely specify that object out of the universe of all objects. The MDL principle was originally introduced by Rissanen [13] and has later been adopted by others.

Let k be the number of distinct classes in C and k^* be the number of distinct pseudo classes in C'. Let the cut point T divide the attribute A into two sets of instances, S_1 and S_2 . Let k_1 and k_2 be the number of distinct elements in S_1 and S_2 respectively, corresponding to class C and let k_1^* and k_2^* be corresponding to class C'. Then the gain corresponding to each class is defined as $\mathrm{G}(A, T; S, C)$ and $\mathrm{G}(A, T; S, C')$, and net gain is defined as $\Omega(A, T; S)$[3]:

$$\begin{aligned}
\mathrm{G}(A, T; S, C) &= \mathrm{Ent}(S, C) - \mathrm{E}(A, T; S, C) \\
\mathrm{G}(A, T; S, C') &= \mathrm{Ent}(S, C') - \mathrm{E}(A, T; S, C') \\
\Omega(A, T; S) &= \frac{1}{2}[\mathrm{G}(A, T; S, C) + \mathrm{G}(A, T; S, C')]
\end{aligned}$$

According to MDL criterion [3], the partition induced by cut point T is accepted if and only if:

$$\Omega(A, T; S) > \frac{1}{2}[\frac{\log_2(N-1)}{N} + \frac{\Delta(A, T; S, C)}{N}] + \frac{1}{2}[\frac{\log_2(N-1)}{N} + \frac{\Delta(A, T; S, C')}{N}]$$

where,

$$\Delta(A, T; S, C) = \log_2(3^k - 1) - [k\mathrm{Ent}(S, C) - k_1\mathrm{Ent}(S_1, C) - k_2\mathrm{Ent}(S_2, C)]$$
$$\Delta(A, T; S, C') = \log_2(3^{k^*} - 1) - [k^*\mathrm{Ent}(S, C') - k_1^*\mathrm{Ent}(S_1, C') - k_2^*\mathrm{Ent}(S_2, C')]$$

3. Experiments and Results

In the experimental study, we have compared the performance (prediction rate) of the discretization method ME-MDL (minimum entropy) with our algorithm and also compared the above

[3]Equal weights assigned to both class C and C', $\alpha = \beta = 0.5$

two with the case when no discretization is done on the data. The analysis is done taking the # of clusters = # of classes + j, where j = 0, 1, 2, 3, 4, 5 but for brevity we present just two cases when $j = 0, 1$. Three different classifiers are used for this: Support Vector Machine using Radial Basis Function[4] , the Naive Bayes classifier and the Maximum Entropy classifier [17]. The Naive Bayes induction algorithm computes the posterior probability of the classes, given the data, assuming independence between features for each class. The probability of the nominal features are estimated using counts and using Kernel Smoothing Density Function.

Table 1: Description of the data sets used in our study

	Number of Attributes				
Dataset	Disc.	Cont.	Size of Dataset	No. of Classes	Majority Accuracy
Iris	0	4	150	3	33.33
Blood	0	4	748	2	76.20
Liver	0	6	345	2	57.97
Heart	5	5	270	2	55.55
Diabetes	0	8	768	2	65.10
Australian	8	6	690	2	55.50
Cleve	7	6	303	2	54.45
Crx	9	6	653	2	54.67
Wine	0	13	178	3	39.88
Breast	0	31	569	2	62.74
Ionosphere	1	32	351	2	64.10

A measure of the quality of the clusters in a given data is the Davies Bouldin index. The Davies Bouldin index [14], DB, is defined as:

$$DB = \frac{1}{n} \sum_{i=1, i \neq j}^{n} max(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)})$$

where n is the number of clusters, σ_i is the average distance of all patterns in cluster i to their cluster center c_i, σ_j is the average distance of all patterns in cluster j to their cluster center c_j, and $d(c_i, c_j)$ is the distance between cluster centers c_i and c_j. Small values of DB correspond to clusters that are compact, and whose centers are far away from each other.

We chose 11 datasets from U.C. Irvine Machine Learning Repository that had atleast one continuous feature. Table 1 describes the datasets in terms of the number and types of features and instances[5] and also gives the Majority Accuracy (percentage of instances having the majority

[4]SVM was applied with Linear and Quadratic kernels. In both of these cases, the number of iterations exceeded the preassigned maximum no. of iterations and time taken to execute is extremely large as compared to RBF-Kernel. The time for the heart dataset increased by a factor of 4 and for the liver dataset, by a factor of 3.

[5]The instances having missing values have been removed from the datasets, therefore, the number indicate the number of data points with no missing values.

class). Firstly, a classifier was tested with the undiscretized data. The continuous feature(s) are then discretized using both ME-MDL and our algorithm and the prediction accuracies of the classifier using this discrete data are recorded. The DB index is calculated for the undiscretized data, as well as the discretized data taking the class information as the clusters, with the data being first normalized between 0 and 1. Tables 2, 3, and 4 contain the accuracies of the SVM, Naive Bayes, and Maximum Entropy classifiers respectively. For all three methods, we report the performance on the discretized as well as on the undiscretized data. Code for the Maximum Entropy classifier is taken from Yan [18]. Table 5 compares the execution time for SVM, NaiveBayes (NB) and Maximum Entropy (MaxEnt). Table 6 gives the DB index for both undiscretized and discretized data. Table 7 depicts the number of discretization levels in Crx and Australian Datasets. Figure 2, 3, 4 shows the performance of SVM, Naive Bayes and Maximum Entropy classifier for different cases. All results were computed using leave-one out cross validation.

Table 2: Predictive Accuracy for the SVM with RBF kernel

		Discretized Attributes			
			Clustering + ME-MDL		
			K-Means		SNN
Dataset	Undiscr. Data	ME-MDL	(A)	(B)	(C)
Iris	95.33	95.33	**98.00**	97.33	93.33
Blood	79.55	81.42	81.42	81.55	**81.82**
Liver	68.41	74.20	**77.10**	70.14	72.17
Heart	75.56	77.41	75.93	77.04	**80.00**
Diabetes	74.48	84.64	**87.63**	83.85	86.33
Austraian	81.88	81.88	83.04	82.03	**87.68**
Cleve	77.56	83.17	87.13	**88.45**	**88.45**
Crx	82.70	86.22	87.44	86.83	**87.90**
Wine	82.02	90.45	**91.01**	88.76	88.76
Breast	75.04	71.00	**78.21**	73.11	77.86
Ionosphere	90.03	66.38	91.17	**91.74**	90.88

(A):Number of clusters = Number of classes
(B):Number of clusters = Number of classes+1
(C):Number of Iterations = 100 & Minimum Cluster Size = 5

For SVM, as illustrated in Table 2, in all datasets our method, when used with SNN clustering or K-Means clustering with number of clusters equal to the number of classes or number of classes + 1, does better than ME-MDL. Only in heart dataset, the performance goes down from 77.41% to 77.04% when K-means clustering is used, however in this case, SNN clustering does better than ME-MDL. It is difficult to asses the better clustering algorithm because in 7 out of 11 cases K-Means clustering is better than SNN clustering.

For the Naive Bayes classifier (see Table 3), in 8 out of 11 cases, our method does better than ME-MDL and in addition, for one dataset (heart) the performance is equal to ME-MDL. Likewise, in all 11 cases, the performance of Maximum Entropy classifier (see Table 4) is better either for K-Means clustering or for SNN clustering.

On comparing the three classifiers, it is observed that Naive Bayes does best in 7 out of 11 datasets and, in one dataset (iris), it is jointly the best, along with SVM. This goes on to show that

simple classification rules work well on most commonly used datasets as observed by Holte [9].

Table 3: Predictive Accuracy for the Naive Bayes with Kernel Smoothing Density Function

		Discretized Attributes			
			Clustering + ME-MDL		
			K-Means		SNN
Dataset	Undiscr. Data	ME-MDL	(A)	(B)	(C)
Iris	96.00	96.67	96.00	**98.00**	96.67
Blood	74.87	74.33	74.60	74.35	**76.87**
Liver	64.35	68.89	71.30	**73.62**	70.43
Heart	79.26	**81.48**	80.37	**81.48**	80.00
Diabetes	73.31	85.02	**88.54**	87.24	**88.54**
Australian	67.55	91.30	89.71	86.68	**92.32**
Cleve	79.21	85.81	89.11	**90.76**	86.14
Crx	67.99	87.75	86.68	88.21	**89.13**
Wine	97.19	**98.31**	96.63	97.31	95.51
Breast	94.20	95.31	95.73	94.73	**96.31**
Ionosphere	91.45	**93.16**	87.75	88.03	86.89

(A):Number of clusters = Number of classes
(B):Number of clusters = Number of classes+1
(C):Number of Iterations = 100 & Minimum Cluster Size = 5

Table 4: Predictive Accuracy for the Maximum Entropy Classifier

		Discretized Attributes			
			Clustering + ME-MDL		
			K-Means		SNN
Dataset	Undiscr. Data	ME-MDL	(A)	(B)	(C)
Iris	95.33	78.00	80.67	**87.33**	86.67
Blood	77.41	75.27	**75.95**	74.87	75.33
Liver	67.54	62.32	55.36	63.19	**69.57**
Heart	76.36	79.26	**80.00**	**80.00**	79.26
Diabetes	67.71	65.23	**74.61**	65.36	**74.61**
Australian	81.01	87.68	**91.30**	**91.30**	84.64
Cleve	76.24	82.18	80.53	**82.84**	80.53
Crx	79.33	77.95	75.50	75.34	**82.85**
Wine	91.01	93.19	94.94	**95.51**	**95.51**
Breast	67.61	96.49	**97.54**	96.66	97.01
Ionosphere	80.34	82.62	**83.48**	80.34	75.21

(A):Number of clusters = Number of classes
(B):Number of clusters = Number of classes+1
(C):Number of Iterations = 100 & Minimum Cluster Size = 5

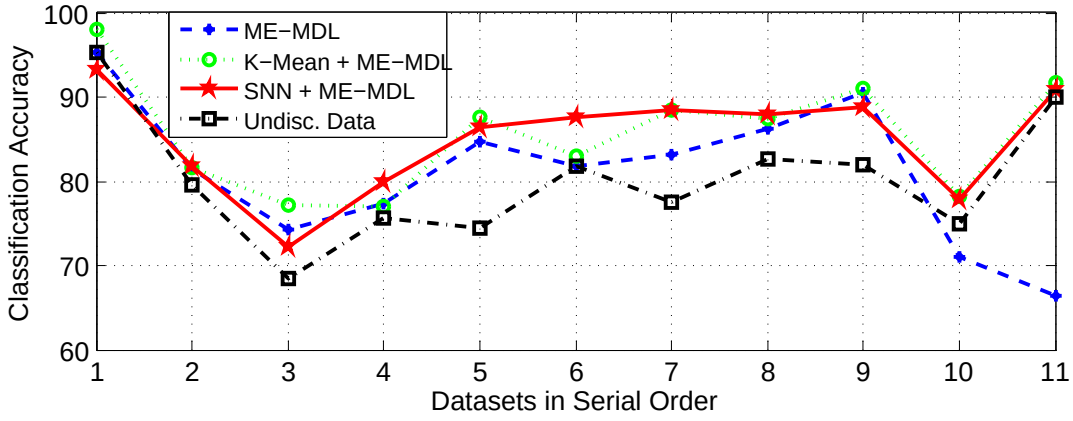


Figure 2: Performance comparison of SVM classification for undiscretized data with discretized data. The X-axis shows the datasets in serial order in accordance to Table 2. The Y-axis shows the prediction accuracy of the classifiers for different techniques.

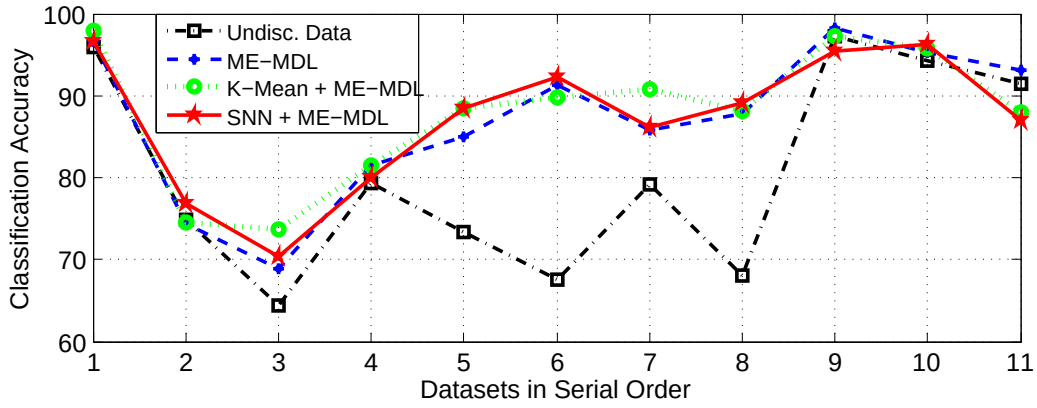


Figure 3: Performance comparison of NB classification for undiscretized data with discretized data. The X-axis shows the datasets in serial order in accordance to Table 2. The Y-axis shows the prediction accuracy of the classifiers for different techniques.

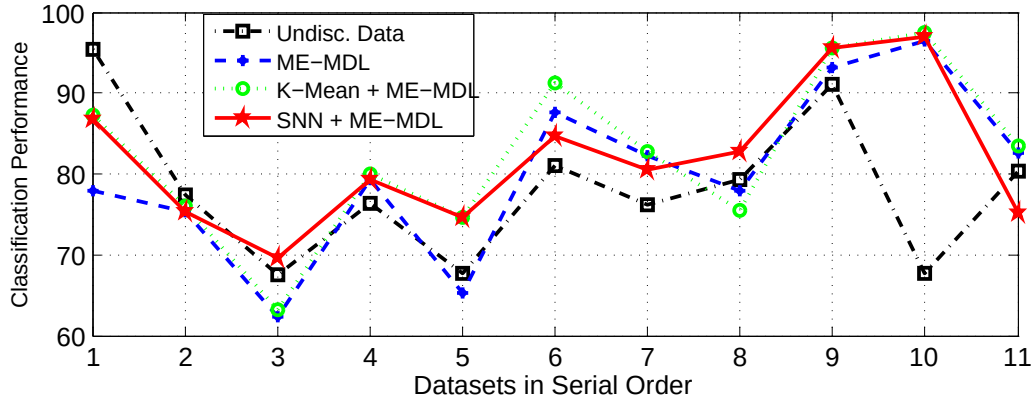


Figure 4: Performance comparison of MaxEnt classification for undiscretized data with discretized data. The X-axis shows the datasets in serial order in accordance to Table 2. The Y-axis shows the prediction accuracy of the classifiers for different techniques.

In Table 5, we have reported only the computational time for classification, because the time required for Clustering and ME-MDL is negligible compared to the amount of time required for the classification of data. It is observed that time taken by Naive Bayes and Maximum Entropy classifier is approximately same for the undiscretized, as well as the discretized version of a particular dataset. However for SVM, time taken to classify undiscretized data is much larger in comparision to its discretized version.

Table 5: Computational time, measured in seconds, for SVM with RBF kernel, Naive Bayes, and Maximum Entropy classification methods

			Discretized Data			
				Clustering + ME-MDL		
				K-Means		SNN
Dataset	Classifier	Undiscr. Data	ME-MDL	(A)	(B)	(C)
Iris	SVM	88.9	27.1	94.2	106.8	20.2
	NB	15.4	15.7	15.4	15.5	15.9
	MaxEnt	146.2	146.9	146.3	147.1	146.0
Liver	SVM	429.5	36.9	156.7	45.3	36.1
	NB	33.6	32.9	32.8	33.2	31.5
	MaxEnt	270.8	217.1	280.2	321.79	218.1
Heart	SVM	73.1	19.1	17.8	17.5	16.0
	NB	45.6	44.4	44.5	44.3	43.1
	MaxEnt	211.1	207.8	215.2	255.8	185.9
Australian	SVM	16438.0	1338.0	1584.0	1212.0	8937.5
	NB	178.9	178.7	178.5	178.6	181.5
	MaxEnt	1280.6	1303.0	1116.0	1128.1	990.5
Cleve	SVM	52.5	24.7	24.9	24.6	25.9
	NB	68.9	68.9	69.3	68.2	67.4
	MaxEnt	222.2	199.8	271.0	272.5	268.0
Crx	SVM	3282.0	2256.0	2352.0	1188.0	10133.0
	NB	186.9	187.2	186.4	187.2	185.4
	MaxEnt	655.6	463.0	841.2	774.4	512.0
Wine	SVM	7.2	42.9	44.7	38.0	237.3
	NB	66.4	66.4	65.6	66.0	63.5
	MaxEnt	180.0	167.0	194.5	219.9	174.5
Breast	SVM	3750.0	1338.0	1806.0	2058.0	903.1
	NB	433.5	432.4	435.0	430.9	437.4
	MaxEnt	1212.0	1111.2	1240.5	1324.5	1226.0
Ionosphere	SVM	1812.0	276.0	1458.0	678.0	1380.1
	NB	292.3	291.3	291.2	300.0	293.0
	MaxEnt	355.1	340.0	387.0	394.5	350.9

(A):Number of clusters = Number of classes
(B):Number of clusters = Number of classes+1
(C):Number of Iterations = 100 & Minimum Cluster Size = 5

As noted above, we observe that discretization has two advantages:

1. Superior classification performance
2. Less time complexity

Improvement in classification performance is contrary to our expectation, because one would expect that when we discretize an attribute, there is some loss of information. A possible explanation is

that in the discretized version, the dataset is more compact, as measured using the Davies Bouldin index. In Table 6, DB index in column 2 for undiscretized data is larger than similar index for discretized data.

Table 6: Davies Bouldin Index

		Discretized Data			
			Clustering + ME-MDL		
			K-Means		SNN
Dataset	Undiscr. Attributes	ME-MDL	(A)	(B)	(C)
Iris	1.0776	0.9144	0.9678	0.9156	0.8862
Blood	4.1826	3.4250	3.9726	3.6517	3.7857
Liver	9.2670	3.7544	8.1713	10.3014	7.8327
Diabetes	3.9106	3.0541	2.8984	3.0633	3.1711
Australian	3.0471	2.6289	2.6066	2.6131	2.6231
Cleve	2.7530	2.2820	2.1682	2.0814	2.0813
Crx	3.3672	3.4067	3.2950	3.3182	3.3972
Wine	1.4629	1.2210	1.2399	1.2499	1.2351
Breast	1.6120	1.2382	1.2063	1.2294	1.2701

(A):Number of clusters = Number of classes
(B):Number of clusters = Number of classes+1
(C):Number of Iterations = 100 & Minimum Cluster Size = 5

It could be argued that better classification is obtained because the number of levels of discretization is higher in cluster based method. This is refuted by the results in Table 7 where we report the discretization levels for the Crx and Australian Dataset (both having 6 continuous attributes). We observe that in all cases, namely MEMDL and Clustering based MEMDL, all six continuous variables are discretized at approximately the same number of levels.

4. Conclusion

The results in the paper show that even if the machine learning algorithm can be applied directly to problem with continuous variables, it is still possible to improve the performance of the algorithm and also reduce the computation time considerably by using discretized variables.

We also observe that discretization of continuous variables simultaneously using the class information and cluster based 'pseudo class' information generally performs better than based on the class information alone.

If we wish to use the K-Means clustering, then we do not have a precise and clear answer for "How many clusters should be used in our scheme ?". Based on experiments on 11 datasets we observe that if the number of desired clusters is equal to the number of classes or number of classes + 1, then the classification performance is better than ME-MDL. Incremental changes, for better or worse, are noted when the number of clusters is 2 or more than the number of classes. We plan to study this question in more detail in our future investigation. As noted earlier, in some instances, SNN clustering gives better performance than the K-Means clustering. This is, perhaps, due to the nature of clusters formed for the datasets under consideration.

Table 7: Discretization levels

CRX DATASET							
Algorithm	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**Total No. of levels**
MEMDL	2	2	5	7	5	4	25
K-Mean Clustering + ME-MDL							
(A)	1	2	3	5	3	1	15
(B)	2	2	4	5	4	4	21
(C)	2	2	4	6	4	4	22
(D)	2	2	4	5	4	4	21
(E)	2	2	4	6	3	4	21
(F)	2	2	5	6	3	5	23
SNN Clustering + ME-MDL							
(G)	2	2	5	8	3	4	24

AUSTRALIAN DATASET							
Algorithm	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**Total No. of levels**
MEMDL	2	2	5	7	5	4	25
K-Mean Clustering + ME-MDL							
(A)	2	2	5	6	3	5	23
(B)	2	2	5	8	4	5	26
(C)	2	3	5	8	4	4	26
(D)	2	2	2	6	3	5	20
(E)	3	3	6	6	4	5	27
(F)	2	2	5	6	3	6	24
SNN Clustering + ME-MDL							
(G)	2	2	5	5	3	4	21

(A):Number of clusters = Number of classes
(B):Number of clusters = Number if classes+1
(C):Number of clusters = Number if classes+2
(D):Number of clusters = Number if classes+3
(E):Number of clusters = Number if classes+4
(F):Number of clusters = Number if classes+5
(G): Number of Iterations = 100 Minimum Cluster Size = 5

References

[1] Kerber, R. (1992), Chimerge: Discretization of numeric attributes, in *Proceeding of Tenth National Conference on Artificial Intelligence*, MIT Press, pp. 123-128.

[2] Huan, L., and Setino R. (1997), Feature Selection via Discretization, *IEEE Transactions on Knowledge and Data Engineering*, Vol 9, No. 4, July/August.

[3] Fayyad, U.M., and Irani, K.B. (1993), Multi Interval discretization of continuous-valued attributes for classification learning, in *Proceeding of the 13th International Joint Conference on Artificial Intelligence*, 1022-1027, Morgan Kauffmann.

[4] Dougherty, J., Kohavi, R., and Sahami, M. (1995), Supervised and Unsupervised discretization of continuous features. *Machine Learning* 10(1), 57-78.

[5] Kohavi, R., Sahami, M. (1996), Error based and Entropy based discretization of Continuous features, in *Proceeding KDD-96* .

[6] Chmielewski M.R., Busse Jersy W. (1996), Global Discretization of continuous attributes as preprocessing for Machine Learning, *International Journal of Approximate Reasoning* 15, 319-331.

[7] Pfahringer B. (1995), Compression based discretization of continuous attributes, in *Proceeding of 12th international conference on Machine Learning*, pages 456-463.

[8] Catlett J. (1991b), On changing continuous attributes into ordered discrete attributes, in Y.Kodratoff,ed., in *Proceeding of the European Working Session on Learning*, Berlin Germany: Springer-Verlag, pp. 164-178.

[9] Holte, R.C. (1993), Very Simple Classification rules work well on most commonly used datasets, *Machine Learning* 11, 63-90.

[10] Monti S., Cooper G.F., (1999), A Latent Variable model for multivariate discretization, *The Seventh International Workshop on Artificial Intelligence and statistics.*

[11] Mehrotra K. G., Ozgencil N. E., McCracken N. (2007), Squeezing the last drop: Cluster based classification algorithm, in *Statistics and Probability Letters 77*, 1288-1299.

[12] Fayyad, U.M. (1991), On the induction of decision trees for Multiple Concept Learning, PhD dissertation, EECS Dept.,The University of Michigan.

[13] Rissanen, J. (1978), Modelling by shortest data description, *Automatica, Vol.14*, pp. 465-471.

[14] Davies, D.L., Bouldin, D.W., (1979), A cluster separation measure, IEEE Trans. Pattern Anal. Machine Intelligence, 1(4), 224-227.

[15] Quinlan J.R. (1993), C4.5 Programs for Machine Learning, Morgan Kaufmann, Los Altos, California.

[16] Ertoẗ L., Steinbach M., and Kumar V., (2002) A shared nearest neighbor clustering algorithm and its applications, *Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining.*

[17] Burger A. L., Pietra, S. D., and Pietra, V. D. (1996) A maximum entropy approach to natural language processing, *Comp. Linguistics*, 39–71.

[18] Yan, R., MATLABArsenal, *http://www.informedia.cs.cmu.edu/yanrong/MATLABArsenal/MATLABArsenal.htm*