БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ . BULGARIAN ACADEMY OF SCIENCES

КИБЕРНЕТИКА И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ • Том **2**, № 2
CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **2**, No 2

София . 2002 . Sofia

On Supervised and Unsupervised Discretization[1]

Gennady Agre, Stanimir Peev***

** Institute of Information Technologies, 1113 Sofia, email: agre@iinf.bas.bg*
*** Faculty of Mathematics and Informatics, Sofia University, 1000 Sofia*

Abstract: *The paper discusses the problem of supervised and unsupervised discretization of continuous attributes – an important pre-processing step for many machine learning (ML) and data mining (DM) algorithms. Two ML algorithms - Simple Bayesian Classifier (SBC) and Symbolic Nearest Mean Classifier (SNMC)) essentially using attribute discretization have been selected for empirical comparison of supervised entropy-based discretization versus unsupervised equal width and equal frequency binning discretization methods. The results of such evaluation on 13 benchmark datasets do not confirm the widespread opinion (at least for SBC) that entropy-based MDL heuristics outperforms the unsupervised methods. Based on analysis of these results a modification of the entropy-based method as well as a new supervised discretization method have been proposed. The empirical evaluation shows that both methods significantly improve the classification accuracy of both classifiers.*

Keywords: *supervised and unsupervised discretization, machine learning, data mining.*

1. Introduction

Discretization is a process of dividing the range of continuous attributes into disjoint regions (intervals) which labels can then be used to replace actual data values. Both in machine learning (ML) and data mining (DM) the discretization techniques are mainly used as a data preprocessing step, however they aim at different goals. In ML such techniques are usually applied in a classification context where the goal is to maximize the predictive accuracy. For example, it is well known fact that the simple Bayesian classifier (SBC) can significantly improve accuracy over a normal approximation [1]. Reducing the number of values for an attribute resulted from the discretization leads to accelerating the decision-tree-based classification methods especially for very large

[1] The research was partially supported by the project IIT-010051 "Advanced methods and tools for knowledge representation and processing".

datasets [2]. It should also be mentioned that many ML algorithms operates only in nominal attribute spaces [3] and that is why a preliminary discretizing of continuous attributes is needed in order to be applied to real databases.

In DM the emphasis is often not on predictive accuracy but rather on finding previously unknown and insightful patterns in the data. In such a context the goal of discretization is to find such intervals that do not hide patterns and are semantically meaningful [4]. Many discretization techniques are used for constructing a concept hierarchy – a hierarchical or multiresolution partitioning of attributes, which is useful for mining at multiple levels of abstraction [2].

The existing discretization methods can be described along several dimensions. In [1] (which is may be the most often cited work on discretization) they are three: *supervised versus unsupervised, global versus local and static versus dynamic*. The supervised methods intensively explore the class information while unsupervised ones do not use it at all. The second dimension highlights the moment when discretization is performed – the global discretization is a preprocessing step carried out *prior* the process of constructing a classifier, while local methods perform discretization *during* such a process (see, for example C4.5 [5]). The static (or *univariate* [4]) methods discretize each attribute independently (or only in conjunction with class attribute) and do not consider interactions with other attributes (e.g. all methods analyzed in [1] fall in this category). The dynamic (or *multivariate*) methods are searching for discretization intervals for all attributes simultaneously thus capturing attribute interdependencies. In [6] authors report on experiments with dynamic versions of some static discretization methods using the wrapper approach. In the classification context it has been found no significant improvement in employing dynamic discretization over its static counterpart. However, in the DM context [4] proposes a method for the multivariate discretization and reports that such a method does not destroy hidden patterns (as univariate methods do) and generates meaningful intervals.

The present paper considers discretization methods *only in the classification context*, i.e. our primary goal is to improve the generalization accuracy of a classifier on a discretized dataset. In our research we have been motivated mainly by the following facts:

1. Discussing such simple unsupervised discretization methods as equal width (EWB) and equal frequency (EFB) binning, most of authors (see e.g.[6, 7, 8]) note that not using of class attribute may potentially lead to losing the essential information due to the formation of inappropriate bin boundaries and consequently such methods will perform poorly in many situations.

2. Most of experimental evidences for better “behavior” of supervised discretization methods over unsupervised ones (e.g. [1, 8]) are based on the error rate evaluation methodology in which the discretization has been applied to the *entire* dataset that then is split into several folds used for training and testing. However, as even some of these authors have recognized in their more recent publications (see, e.g. [6]), discretizing all the data once before creating the folds allows the discretization method to have access to the testing data, which results to *optimistic error rates.*

3. Recognizing such a deficiency in the evaluation methodology the issue that “the supervised learning methods are slightly better than the unsupervised methods” [1] has not been revised and still is used as an axiom by other researchers without carrying out new experiments. However, it may be expected that the supervised methods, as

more "knowledgeable", will achieve *more advantage* from accessing to the testing data than their "blind" unsupervised counterparts, thus the "degree of optimism" in the error rate would be different for supervised and unsupervised discretization techniques.

In order to check the "axiom" that supervised methods are better than unsupervised ones for classification purposes, we have decided to compare some representatives of both methods in the correct experimental settings. Due to their simplicity, EFB and EWB methods have been selected as representatives of unsupervised discretization, and the entropy-based method (EBD) proposed by Fayyad and Irani [9] – as their supervised "opponent". This method is one of the most popular supervised discretization techniques and has been experimentally proved to outperform some other supervised methods [6, 1]. In order to illustrate the effectiveness of the mentioned above discretization algorithms they are used in conjunction with two different in their nature classification algorithms – Simple Bayesian Classifier (as it is described in [10]) and Symbolic Nearest Mean (SNMC) classifier (proposed in [11]). We have selected these algorithms since both classifiers are proved to be significantly more accurate on discretized datasets [1, 12].

The structure of this paper is as follows: the next section considers the discretization methods selected for empirical evaluation in more details. It also briefly discusses SBC and SNMC classifiers and their implementations. The third section describes experimental settings and the results of evaluation of the algorithms. The fourth section is devoted to the analysis of the results, proposes two new supervised discretization algorithms and presents the results of their empirical evaluation. The last section is reserved for a discussion and summary of this work.

2. Methods and classifiers

2.1. Discretization methods

The Equal Width Binning is the simplest unsupervised discretization method. It divides the range of an observed continuous attribute on *k* equal sized bins (intervals), where *k* – is a user-defined parameter. In the Equal Frequency Binning Method the intervals are created so, that, roughly, the frequency of each interval is constant (that is, each interval contains roughly the same number of continuous attribute values). The number of intervals *k* is also specified by the user. In our experiments, described in the next section, we selected $k = 10$ as it is recommended in [13].

The entropy-based supervised discretization (EBD) method proposed by Fayyad and Irani [9] may be seen as a divisive hierarchical clustering method used entropy measure as a criterion for recursively partitioning the values of a continuous attribute and Minimum Description Length (MDL) principle – as a stopping criterion. Given a set of examples *S*, the basic method for EBD of an attribute *A* is as follows:

1. Each value *v* of *A* can be considered as a potential interval boundary *T* and thereby can create a binary discretization (e.g. $A < v$ and $A \geq v$).

2. Given *S*, the boundary value selected is the one that maximizes the information gain resulting from subsequent partitioning. The information gain is:

$$\text{InfGain}\,(S,T) = \text{Ent}\,(S) - \text{IE}\,(S,T),$$

where IE(*S*, *T*) is the class information entropy determined by the formula:

$$\mathrm{IE}(S,T) = \frac{|S_1|}{|S|}\mathrm{Ent}(S_1) + \frac{|S_2|}{|S|}\mathrm{Ent}(S_2),$$

where $|S_1|$ and $|S_2|$ correspond to the examples of S satisfying the conditions $A < T$ and $A \geq T$ respectively. The entropy function Ent for a given set S_i is calculated based on the class distribution of the samples in the set, i.e.:

$$\mathrm{Ent}(S_i) = -\sum_{j=1}^{k} P(c_j)\log_2 P(c_j),$$

where $P(c_j)$ is the probability of class c_j in S_i, determined by the proportion of examples of class c_j in the set S_i and k is a number of classes in S_i.

3. The process of determining a new interval boundary is recursively applied to each interval produced in previous steps, until the following stopping criterion based on MDL principle is satisfied:

$$\mathrm{InfGain}(S,T) < \delta,$$

$$\delta = \frac{\log_2(n-1) + \log_2(3^k - 2) - [k\mathrm{Ent}(S) - k_1\mathrm{Ent}(S_1) - k_2\mathrm{Ent}(S_s)]}{n},$$

where k_i is the number of classes represented in the set S_i and n is a number of examples in S.

Since the described above procedure is applied independently for each interval, it is possible to achieve the final set of discretization intervals with different size – some areas in the continuous spaces will be partitioned very finely whereas others (with relatively low entropy) will be partitioned coarsely.

2.2. Classification algorithms

The Naive Bayesian Classifier is built based on a conditional independence model of each attribute given the class [14]. The probability of an unlabeled example $X = \langle A_1, \ldots, A_m \rangle$ to be classified as belonging to class c_i is given by Bayes theorem:

$$P(c_i \mid X) = \frac{P(X \mid c_i)}{P(X)}.$$

Having in mind that $P(X)$ is the same for all classes and after applying the condition independence assumption we have the following:

$$P(c_i \mid X) \propto \prod_{j=1}^{m} P(A_j \mid c_i) \cdot P(c_i).$$

This probability is computed for each class and the prediction is made for the class with the largest posterior probability.

For nominal attributes A_j conditional probabilities $P(A_j|c_i)$ are estimated as counts from the training set. The continuous attributes are assumed to be normally distributed and the corresponding conditional probabilities are estimated by the probability density function for a normal (or Gaussian) distribution [15].

A variant of Naive Bayesian algorithm, in which only nominal or discretized attributes are used, is called *Simple Bayesian Classifier* [15, 16, 10]. It has been shown that the simple Bayesian classifier outperforms his naive “brother” [1]. It has been also proved that SBC is very robust and performs well even in the face of obvious violation of the independence assumption [10].

There are several variants of SBC, which are mainly differed by the method of treating missing values and estimating the probabilities (especially for that with zero counts) [16]. In order to avoid losing potentially useful information, we treat missing values as having the value “?” at both training and testing times, and null attribute probabilities $P(A_j|c_i)$ are replaced by $P(A_j|c_i)/n$, where n is a number of training examples, as done in [17].

Symbolic Nearest Mean Classifier [11] is a prototype-based learning algorithm, which classifies an unseen example by calculating its distance to each of stored prototypes – artificial examples constructed from the training examples by a special learning method. As in case of the nearest neighbor algorithm, SNMC assigns to an unseen example the class of its nearest prototype. The classifier can work only with nominal attributes and, that is why, requires discretizing continuous attributes as a preprocessing step. For calculating distances SNMC explores the modified value difference metric (MVDM) proposed in [18] and applies it both in learning and classification phases. More exactly, in SNMC the distance $d(X,Y)$ between two examples (or prototypes) $X = <x_1, \ldots, x_m, c_i>$ and $Y = < y_1, \ldots, y_m, c_j>$ is defined as:

$$d(X,Y) = \sqrt{\sum_{k=1}^{m} \delta^2_{\mathrm{MVDM}}(x_k, y_k)},$$

$$\delta_{\mathrm{MVDM}}(x_i, y_i) = \frac{1}{2} \sum_{j=1}^{n} |\, P(c_j \mid A_i = x_i) - P(c_j \mid A_i = y_i) \,|,$$

where c_j is j-th class and n is a number of classes in training set.

The MVDM metrics is used for defining a notion of mean value of a set of nominal values, which allows SNMC to use *k*-MEANS clustering algorithm to group training examples of the same class and to create several prototypes per class.

It has been reported [11] that SNMC in average outperforms C4.5 and PEBLS on 20 benchmark databases. It has also been shown [12] how the accuracy of SNMC may be further improved. It should be mentioned that in our implementation of SNMC all missing values have been treated as having the value “?” at both training and testing sets.

3. Equal width and equal frequency binnings versus entropy-based discretization: emperical evaluation

In order to evaluate the efficiency of the selected discretization algorithms thirteen benchmark datasets containing at least one continuous attribute have been selected from the widely used UCI repository [19]. The main characteristics of these bases are presented in Table 1.

To calculate classification accuracy for application of each discretization and classification method pair to each database, we used 3-fold stratified cross-validation repeated 30 times[2] (i.e.30x3-CV). In performing cross-validation we separately

[2] Splitting each database on 2/3 for training and 1/3 for testing was used to allow comparing the results with the accuracies of other algorithms not used in our experiments but published in various papers.

Table 1. Datasets used in the empirical study. The columns are, in order: name of the database; 2-letter code used to refer to it in subsequent tables; number of examples; number of attributes; number of continuous attributes; number of classes; percentage of missing values; whether or not the dataset includes inconsistent examples (i.e. identical examples with different classes)

Dataset	**Code**	**Examples**	**Attributes**	**Continuous**	**Classes**	**Missing, %**	**Inconsistency**
Breast cancer Wisconsin	BW	699	10	10	2	2.2	No
Credit screening	CE	690	15	6	2	0.6	No
Pima diabetes	DI	768	8	8	2	0	No
Glass	GL	214	9	9	6	0	No
Glass2	G2	163	9	9	2	0	No
Heart diseases	HD	303	13	6	2	0.2	No
Horse colic	HO	300	22	7	2	24.3	Yes
Hepatitis	HE	155	19	6	2	0	No
Iris	IR	150	4	4	3	0	No
Labor negotiations	LA	57	16	8	2	35.7	No
Liver diseases	LD	345	6	6	2	0	No
Vehicle	VH	846	18	6	4	0	No
Wine	WI	178	13	13	3	0	No

discretized each training set and applied the resulted discretization intervals to the corresponding testing set. It should be mentioned that the same 90 folds were used for all discretization and classification algorithms.

The accuracy values on these folds were used for determining the statistical significance of the results by calculating *p*-values according to the one-tailed *t*-paired test [20]. The overall behavior of the algorithms was also evaluated by comparing their average accuracy across all databases and by means of Wilcoxon matched pairs signed ranks test [20]. All experiments were conducted in the environment of DaMiS – an experimental data mining system developed by the authors [21].

Table 2. Results of experiments with SBC: average accuracies and standard deviations; *p*-values; results of Wilcoxon test and number of significant wins against losses

Database	**Entopy-Based Discretization**	**Equal Frequency (*k*=10) Discretization**		**Equal Width (*k*=10) Discretization**	
	Accuracy	**Accuracy**	***p*-value**	**Accuracy**	***p*-value**
BW	97.3 ± 0.2	97.4 ± 0.1	0.005	97.3 ± 0.1	> 0.1
CE	85.7 ± 0.5	85.6 ± 0.5	> 0.1	84.6 ± 0.6	0.0005
DI	74.7 ± 0.8	74.5 ± 0.6	> 0.1	75.5 ± 0.6	0.0005
GL	66.5 ± 2.9	67.6 ± 2.6	> 0.1	62.1 ± 3.2	0.0005
G2	78.8 ± 2.4	78.3 ± 2.3	> 0.1	75.2 ± 3.1	0.0005
HD	83.0 ± 0.9	83.0 ± 0.9	> 0.1	83.3 ± 0.9	> 0.1
HE	84.3 ± 1.4	84.9 ± 1.7	0.05	85.7 ± 1.4	0.0005
HO	78.8 ± 0.9	79.0 ± 0.7	> 0.1	78.6 ± 0.6	> 0.1
IR	94.0 ± 0.8	93.8 ± 1.2	> 0.1	94.6 ± 1.2	0.05
LA	90.2 ± 4.3	93.9 ± 3.4	0.0005	93.3 ± 3.2	0.0005
LD	57.3 ± 1.2	62.7 ± 1.7	0.0005	63.3 ± 2.0	0.0005
VH	59.6 ± 0.7	62.0 ± 1.0	0.0005	60.8 ± 1.1	0.0005
WI	97.8 ± 0.8	97.0 ± 0.6	0.0005	97.0 ± 0.6	0.0005
Average	80.61	**81.51**		**80.86**	
Wilcoxon		α > 0.05		α > 0.05	
Sign. wins		5 - 1		6 – 4	

Table 3. Results of experiments with SNMC: average accuracies and standard deviations; *p*-values; results of Wilcoxon test and a number of significant wins against losses

Database	**Entropy-Based Discretization**	**Equal Frequency (*k*=10) Discretization**		**Equal Width (*k*=10) Discretization**	
	Accuracy	**Accuracy**	***p*-value**	**Accuracy**	***p*-value**
BW	97.0 ± 0.2	97.2 ± 0.2	0.025	97.2 ± 0.2	0.025
CE	85.1 ± 0.5	86.0 ± 0.7	0.0005	85.6 ± 0.4	0.005
DI	73.8 ± 1.0	74.5 ± 1.0	0.01	74.9 ± 0.8	0.0005
GL	65.6 ± 2.8	67.6 ± 3.1	0.01	60.6 ± 3.4	0.0005
G2	77.3 ± 2.3	77.5 ± 2.8	> 0.1	75.6 ± 2.6	0.05
HD	82.3 ± 1.4	82.2 ± 1.3	> 0.1	82.6 ± 1.1	> 0.1
HE	81.0 ± 2.2	81.5 ± 2.5	> 0.1	81.8 ± 2.3	> 0.1
HO	83.3 ± 1.1	83.3 ± 1.2	> 0.1	83.4 ± 0.9	> 0.1
IR	93.8 ± 1.0	94.8 ± 1.1	0.001	96.2 ± 1.1	0.0005
LA	83.8 ± 4.9	85.8 ± 4.7	0.025	89.9 ± 2.7	0.0005
LD	57.3 ± 1.5	61.5 ± 2.3	0.0005	62.2 ± 2.6	0.0005
VH	61.6 ± 1.5	65.2 ± 1.2	0.0005	64.2 ± 2.0	0.0005
WI	96.2 ± 1.2	96.7 ± 1.1	0.01	96.3 ± 1.2	> 0.1
Average	79.85	**81.06**		**80.81**	
Wilcoxon		$\boldsymbol{\alpha = 0.005}$		$\alpha > 0.05$	
Sign. wins		9 – 0		7 – 2	

Table 2 compares three discretization algorithms by measuring the classification accuracy of simple Bayesian classifier. p-values measure statistical significance of differences between each unsupervised discretization algorithm and its supervised opponent. Table 3 contains the analogous information concerning experiments with SNMC algorithm.

The average accuracy of classifiers across all databases is a measure of debatable significance, but it has often been used for evaluating the ML algorithms (see, e.g. [5, 22]). In both cases using of entropy-based discretization leads to worse results than applying the unsupervised discretization methods. More over, SNMC in combination with EFB is a more accurate algorithm that SNMC with EBD with a confidence in excess of 99.5%. Counting a number of significant wins against a number of significant losses also confirms the conclusion that on these 13 databases the selected two unsupervised discretization methods are more effective than the supervised one.

Since these experimental results contradict (at least for SBC) to the corresponding results reported in [1] and [8], we have decided to repeat the experiments in the (*incorrect*) experimental setting similar to that used in the cited works. In other words, at the beginning we discretized *the whole* database and then applied to it the 30x3-CV method. It should be mentioned, that in this case we also used the same 90 folds as used in the previous experiments. In Table 4 you can see how the accuracy of SBC is changed when the discretization process is applied *before* and *during* the cross-validation.

As one can see, the applying of incorrect error rate evaluation methodology really distorted the results. However, according to Wilcoxon test there is no significant difference for unsupervised methods. More over, the results are slightly worse, which may be considered as "overfitting" effect. In the case of EWB the decreasing of average accuracy may be explained by more global influence of outliers since the value range of each attribute is calculated on the entire dataset rather than on the base of 2/3 of pos-

Table 4. Results of experiments with SBC in different experimental settings: average accuracies and standard deviations in the cases when the discretization has been done during and before cross-validation; *p*-values; results of Wilcoxon test and a number of significant wins against losses

Database	**Entopy-Based Discretization**			**Equal Frequency (*k*=10) Discretization**			**Equal Width (*k*=10) Discretization**		
	During CV	**Before CV**	***p*-value**	**During CV**	**Before CV**	***p*-value**	**During CV**	**Before CV**	***p*-value**
BW	97.3 ± 0.2	97.3 ± 0.1	> 0.1	97.4 ± 0.1	97.4 ± 0.1	> 0.1	97.4 ± 0.1	97.3 ± 0.1	0.05
CE	85.7 ± 0.5	85.9± 0.5	0.1	85.6 ± 0.5	85.4 ± 0.5	0.05	84.6 ± 0.6	84.7 ± 0.4	> 0.1
DI	74.7 ± 0.8	78.1 ± 0.4	0.0005	74.5 ± 0.6	74.3 ± 0.6	0.05	75.5 ± 0.6	75.2 ± 0.5	0.005
GL	66.5 ± 2.9	73.3 ± 1.4	0.0005	67.6 ± 2.6	68.5 ± 2.1	0.025	62.1 ± 3.2	58.3 ± 2.7	0.0005
G2	78.8 ± 2.4	84.8 ± 1.2	0.0005	78.3 ± 2.3	78.3 ± 2.3	> 0.1	75.2 ± 3.1	77.3 ± 2.1	0.0005
HD	83.0 ± 0.9	83.3 ± 0.6	0.025	83.0 ± 0.9	82.6 ± 0.9	0.025	83.3 ± 0.9	83.1 ± 1.0	> 0.1
HE	84.3 ± 1.4	85.2 ± 1.1	0.0005	84.9 ± 1.7	85.5 ± 1.7	0.05	85.7 ± 1.4	85.7 ± 1.3	> 0.1
HO	78.8 ± 0.9	79.0 ± 0.8	> 0.1	79.0 ± 0.7	79.4 ± 0.8	0.005	78.6 ± 0.6	78.4 ± 0.6	0.025
IR	94.0 ± 0.8	94.5 ± 0.5	0.025	93.8 ± 1.2	91.9 ± 1.4	0.0005	94.6 ± 1.2	95.5 ± 1.1	0.025
LA	90.2 ± 4.3	96.6 ± 3.6	0.0005	93.9 ± 3.4	93.7 ± 3.0	> 0.1	93.3 ± 3.2	93.5 ± 3.2	0.1
LD	57.3 ± 1.2	63.2 ± 3.3	0.0005	62.7 ± 1.7	61.5 ± 1.8	0.005	63.3 ± 2.0	63.9 ± 2.0	0.1
VH	59.6 ± 0.7	62.4 ± 0.8	0.0005	62.0 ± 1.0	62.1 ± 1.1	0.1	60.8 ± 1.1	60.1 ± 1.0	> 0.1
WI	97.8 ± 0.8	99.0 ± 0.3	0.0005	97.2 ± 0.9	97.3 ± 0.8	> 0.1	97.0 ± 0.6	96.5 ± 0.6	0.001
Average	80.61	**83.27**		**81.53**	81.37		**80.88**	80.86	
Wilcoxon	***α* = 0.0005**			***α* > 0.05**			***α* > 0.05**		
Sign. wins	11 - 0			4 - 5			3 - 5		

sible attribute values. Concerning EFB method, the global application of this method leads to increase in the number of attribute values used for constructing the equal frequency intervals. Thus, when these values are not distributed uniformly (that is the case for most datasets) the result is the set of intervals with bigger variance in frequency amount.

In the case of the supervised discretization changing the evaluation framework leads to increase of average accuracy at the confidence level of 99.95%, i.e. with this level of confidence on can state that these results have been produced by *absolutely different classifiers,* which is not the case!

Thus our experiments prove the following:

1. The error rate evaluation methodology, in which the *whole* database is discretized during a preprocessing step, *is incorrect and should never be used for evaluating discretization methods in classification context.*

2. In the framework of mentioned above 13 benchmark databases and two classifiers (SBC and SNMC) two simple unsupervised discretization methods (EFB and EWB) behave *better* (especially EFB) that the entropy-based supervised discretization method proposed by Fayyad and Irani.

3. The results of all experiments for comparing the discretization methods, conducted in the framework of mentioned above incorrect evaluation methodology, *should be revised.*

4. How to improve the supervised discretization

In this section we introduce two new supervised discretization methods – the first one is a modification of the mentioned above entropy-based technique, and the second – is an attemt to combine the aglomerative hierarchical clustering approach for constructing the discretization intevals with application of MVDM metrics for mesuaring the “semantic” quality of such a dicretization.

4.1. Modified entropy-based dicretization

One of the possible reasons for comparatively weak behavior of EBD method is a too coarse granulation of the discretization intervals for some attributes caused by applying of MDL principle as a stopping criterion. In order to check this hypothesis we measured a number of discretization intervals for each continuous attribute produced by this supervised discretization method and compared it to a number of classes in the corresponding datasets[3] .

The results are summarized in Table 5 where the percent of continuous attributes, for which a number of discretization intervals is less than a number of classes, is presented. For such an attribute, for example, SBC algorithm would tend to classify every unseen example to the majority class. In its turn, SNMC algorithm would do the same or simply ignore all such attributes (for 2 classes of problems).

To empirically evaluate the mentioned above hypothesis we modified the MDL stopping criterion by adding a constraint on a minimum number of possible discretization intervals, i.e. now the recursive discretization process is stopped only if the information gain is less than the corresponding threshold *and* the number of discretization intervals is greater than or equal to the number classes. If this additional condition is not satisfied, the new – more detailed discretization minimizing the class information entropy is accepted even if such a discretization does not maximize the information gain.

Table 5. Some characteristics of the EBD discretized datasets used in the empirical study: number of classes, default accuracy and a percent of continuous attributes for which the number of discretization intervals is less than the number of classes.

Database	**Classes**	**Default Accuracy (in %)**	**Percent of cont. atts with number of intervals < number of classes**
BW	2	65.5	10
CE	2	55.5	0
DI	2	65.1	25
GL	6	35.5	100
G2	2	53.4	45
HD	2	54.4	40
HO	2	79.4	50
HE	2	63	57
IR	3	33.3	0
LA	2	64.9	25
LD	2	58	86
VH	4	25.5	27.8
WI	3	39.9	38.5

[3] The experiment was done on the entire datasets rather than on each of 2/3 part of the datasets used as training set in cross-validation methodology. However, this does not cause some significant differences since the *stratified* cross-validation was used.

4.2. MVDM-based discretization

MVDM metrics [18] has been proposed as an attempt to assess more precisely the similarity between different values of a nominal attribute in the context of supervised ML. The basic idea is that two nominal values are similar if they have the similar behaviour among examples from all classes. Since the objective of each supervised discretization method is to construct such intervals that are more "semantically" contrast to each other, MVDM metrics is a promising candidate for such purposes. In [23] a set of discretization methods combining techniques of hierarchical clustering and application of MVDM metrics have been investigated. One of such method based of agglomerative hierarchical clustering approach is presented here.

Given a set S of training examples, the basic agglomerative hierarchical discretization algorithm for an attribute A may be described as follows:

1. Pre-processing step: Sort all distinct values of A in ascending order.

2. Group all values on a set of some basic intervals $I = \{I_1, \ldots, I_k\}$ (in most algorithms each basic interval consists of a single distinct value (which may occur l times in S).

3. Select two adjacent intervals I_i and I_{i+1}, which are *the closest* to each other according to a given distance measure $D(I_k, I_j)$, and merge them into a single interval I_i.

4. Repeat Step 3 until some stopping criterion $M(I)$ is met.

5. Post-processing step (optional): Evaluate an additional discretization quality criterion $Q(I)$. If it is not satisfied – change I in an appropriate way and repeat the discretization process beginning with Step 3.

Since the adjacent values of an attribute may occur in examples belonging to different classes, we define the measure of closeness between two adjacent intervals as a weighted MVDM distance between them: $D(I_j, I_{j+1}) = (n_j + n_{j+1})\delta_{\mathrm{MVDM}}(I_j, I_{J+1})$, where n_j and n_{j+1} are the number of values (possibly duplicated) in the corresponding intervals. Such a distance aims at creating not only as more class-homogeneous but also more equally condensed intervals as possible[4].

A stopping criterion evaluates the quality of the current partitioning by calculating (for example) a function of distances between discretization intervals (see, e.g. [24]). In our algorithm the merging of intervals is stopped when the MVDM distance between the closest intervals becomes greater than a given threshold a.

To avoid the affect of overfitting, in most discretization algorithms it is desirable to add an additional post-processing step[5]. It restricts a minimum number of attribute values to be contained in a single (final) discretization interval (see e.g. [25, 24]). Such a restriction allows to process imperfect data by considering a compact but sparse group of values as noise, which can be removed.

Since such a parameter is very domain dependent, we introduced another parameter p ($0 < p = 1$) describing the quality of the discretization. A discretization $I = \{I_1, \ldots, I_k\}$ containing N (possibly duplicated) attribute values is called *p-quality satisfyable* if there does not exist an interval such that $n_m = (1 - p)N$, where n_m is a

[4] The selection of such a distance function has been motivated by an excellent behavior of EFB discretization method described in the previous section. See [23] for description of several MVDM-based distance measures and stopping criteria different from the mentioned above.

[5] The restriction that the minimum number of discretization intervals should not be less than the number of classes, that we have added to the entropy-based discretization (see previous subsection) is a kind of such post-processing step.

number of attribute values contained in interval I_m. For example, if we set p = 0.95, that any discretization, each interval of which contains more than 5% of the whole amount of attribute values, will be 0.95-quality satisfyable. It can be shown [23], that a 0.95-quality satisfyable discretization cannot consist of more than 19 intervals (which corresponds to the case of the unified distribution of values). The corresponding numbers of intervals for p = 0.9 and p = 0.75 are 9 and 3.

Introducing the described above parameter p allows us to define noisy intervals as ones that contain less than $1-p$ part of attribute values and to organize a cyclic process for finding a p-quality satisfyable discretization. If a current MVDM-based discretization, satisfying the described above stopping criterion for a given threshold ε, is not p-quality satisfyable, then only p part of its "greatest" (in the sense of amount of attribute values in it) intervals are stored and others are removed as "noisy" (or more precisely, all values in such intervals are removed). Then a next attempt to find the desired discretization by merging new adjacent intervals is done until a p-quality satisfyable discretization is found.

4.3. Experiment results

Tables 6 and 7 compare the effectiveness of the modified entropy-based and the MVDM-based discretization methods described in the previous two subsection versus the "original" entropy-based discretization proposed by Fayyad and Irani (described in details in Section 2). The fist table contains the results of the experiments with SBC algorithm and the second – with SNMC. For MVDM discretization the following parameter values were used: ε = 0.2 and p = 0.93.The experimental setting is the same as described in Section 3.

Table 6. Results of experiments with SBC algorithm: average accuracies and standard deviations; p-values; results of Wilcoxon test and a number of significant wins against losses

Database	**Entopy-Based Discretization**	**Modified Entropy-Based Discretization**		**MVDM-based Discretization**	
	Accuracy	**Accuracy**	***p*-value**	**Accuracy**	***p*-value**
BW	97.3 ± 0.2	97.3 ± 0.2	> 0.1	97.3 ± 0.2	> 0.1
CE	85.7 ± 0.5	85.6 ± 0.5	> 0.1	85.7 ± 0.6	> 0.1
DI	74.7 ± 0.8	74.3 ± 1.0	0.025	74.1 ± 1.2	0.025
GL	66.5 ± 2.9	67.6 ± 2.3	> 0.1	65.5 ± 2.7	> 0.1
G2	78.8 ± 2.4	80.1 ± 2.1	0.0005	81.7 ± 2.3	0.0005
HD	83.0 ± 0.9	83.0 ± 1.1	> 0.1	83.0 ± 0.9	> 0.1
HE	84.3 ± 1.4	84.8 ± 1.8	0.05	84.1 ± 1.4	> 0.1
HO	78.8 ± 0.9	78.7 ± 0.7	> 0.1	78.9 ± 0.9	> 0.1
IR	94.0 ± 0.8	94.1 ± 0.8	> 0.1	94.2 ± 1.2	> 0.1
LA	90.2 ± 4.3	93.4 ± 3.7	0.0005	93.3 ± 3.3	> 0.1
LD	57.3 ± 1.2	63.3 ± 2.5	0.0005	63.8 ± 1.9	0.0005
VH	59.6 ± 0.7	60.6 ± 0.9	0.0005	60.8 ± 1.1	0.0005
WI	97.8 ± 0.8	98.0 ± 0.7	0.05	95.9 ± 1.2	0.0005
Average	80.61	**81.60**		**81.41**	
Wilcoxon		***α* = 0.05**		α > 0.05	
Sign. wins		6 – 1		4 – 2	

Table 7. Results of experiments with SNMC: average accuracies and standard deviations; *p*-values; results of Wilcoxon test and a number of significant wins against losses

Database	**Entopy-Based Discretization**	**Modified Entropy-Based Discretization**		**MVDM-based Discretization**	
	Accuracy	**Accuracy**	***p*-value**	**Accuracy**	***p*-value**
BW	97.0 ± 0.2	97.0 ± 0.2	> 0.1	96.9 ± 0.3	> 0.1
CE	85.1 ± 0.5	85.1 ± 0.6	> 0.1	85.7 ± 0.5	0.0005
DI	73.8 ± 1.0	73.7 ± 0.9	> 0.1	73.9 ± 1.2	> 0.1
GL	65.6 ± 2.8	66.4 ± 3.0	> 0.1	67.3 ± 3.1	0.05
G2	77.3 ± 2.3	79.3 ± 2.3	0.0005	81.6 ± 2.2	0.0005
HD	82.3 ± 1.4	82.1 ± 1.6	> 0.1	82.4 ± 1.3	> 0.1
HE	81.0 ± 2.2	81.3 ± 1.8	> 0.1	80.8 ± 2.3	> 0.1
HO	83.3 ± 1.1	83.2 ± 1.2	> 0.1	83.2 ± 1.4	> 0.1
IR	93.8 ± 1.0	93.9 ± 1.0	> 0.1	95.0 ± 1.1	0.0005
LA	83.8 ± 4.9	87.1 ± 3.6	0.0005	86.0 ± 4.7	0.005
LD	57.3 ± 1.5	62.2 ± 3.1	0.0005	62.5 ± 2.4	0.0005
VH	61.6 ± 1.5	62.3 ± 1.7	0.025	64.0 ± 1.7	0.0005
WI	96.2 ± 1.2	95.7 ± 1.1	0.05	95.3 ± 1.3	0.01
Average	79.85	**80.71**		**81.12**	
Wilcoxon		***α* = 0.05**		***α* = 0.025**	
Sign. wins		4 – 1		7 – 1	

The results prove that the proposed modification of the entropy-based discretization has a *significantly better* behavior than the original discretization algorithm. It should be noted that this improvement practically does not depend on the classification algorithm used. However, even this improved supervised discretization algorithm statistically is not better that its unsupervised counterpart – the equal frequency binning discretization method.

It is not surprising that MVDM-based discretization is more appropriate for instance-based learning classifier, especially for those exploring MVDM metrics. Applying this discretization significantly improves the accuracy of SNMC in 7 domains and decreases it only on 1 dataset. More over, according to Wilcoxon test SNMC combining with this discretization outperforms combination of SNMC and EBD method at probability of 97.5% on all 13 datasets. A possible explanation of this fact is that by removing some "noisy" attribute values such the MVDM discretization produces a "refined" MVDM distance for each discretized attribute that is further explored by SNMC classifier.

However, it should be noted that the MVDM-based discretization also does not lead to statistically significant improvement in classification accuracy versus the equal frequency binning discretization method.

5. Conclusions

After evaluation on 13 benchmark databases and two different classification algorithms – simple Bayesian classifier and symbolic nearest mean classifier, we have empirically proved that the error rate evaluation methodology in which the *whole* database is discretized during the preprocessing step *is incorrect and should never be used for evaluating discretization methods in classification context.*

In the mentioned above experimental framework we have also shown that two simple unsupervised discretization methods (EFB and EWB) behave *better* (especially EFB) than the entropy-based supervised discretization method proposed by Fayyad and Irani.

We have proposed a modification of such entropy-based discretization method in which the stopping condition based on MDL principle is modified by additional requirement on the final number of discretization intervals, which should not be less than the number of classes. The empirical evaluation has proved that the proposed modification leads to statistically significant improvement in classification accuracy of both SBC and SNMC algorithms running on 13 discretized benchmark databases.

The MDL principle may be seen as an instance of well-known Occam's razor favoring the simpler of two models with the same training-set error because this leads to lower generalization error. In his article [26] P. Domingos found that such formulation of this principle is provably and empirically false, and our results may be considered as an additional argument for this.

We have also introduced a new supervised discretization method combining the agglomerative hierarchical clustering approach with using of MVDM metrics. This new method has been empirically proved to be significantly better (in combination with SNMC) than the original entropy-based discretization.

Our experimental evidences have allowed us to conclude that the axiom on "better behavior of supervised discretization methods over unsupervised ones" should be revised or at least be supported by more convincible experiments conducting in accordance with the correct error rate evaluation methodology.

References

1. Dougherty, J., R. Kohavi, M. Sahami. Supervised and unsupervised discretization of continuous features. – In: Machine Learning: Proceedings of the 12th Int. Conference, Morgan Kaufmann, 1995, 194-202.
2. Han, J., M. Kamber. Data Mining: Concepts and Techniques. San Francisco, Morgan Kaufmann, 2001.
3. Mitchell, T. Machine Learning. McGraw-Hill Comp., 1998.
4. Bay, S. Multivariate discretization of continuous variables for set mining. – In: KDD-2000: Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, Boston, MA: AAAI Press, 2000, 315-319.
5. Quinlan, J. R.. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1996.
6. Kohavi, R., M. Sahami. Error-Based and Entropy-Based Discretization of Continuous Features. – In: KDD-96: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Menlo Park, CA: AAAI Press, 1996, 114-119.
7. Sikonja, M. Robnik, Igor Kononenko. Discretization of continuous attributes using relief. – In: Proceedings of ERK'9 , Portoroz, Slovenia, 1995.
8. Recheldi, M., M. Rossotto. Class-driven statistical discretization of continuous attributes. – In: ECML'95: Proceedings of the 8th European Conference on Machine Learning, Springer, 1995, 335-338.
9. Fayyad, U, K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. – In: Proceedings of the 13th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1993, 1022-1027.
10. Domingos, P, M. Pazzani. Beyond independence: conditions for the optimality of the simple Bayesian classifier. – In: Machine Learning: Proceedings of the 13th Int. Conference (ICML'96), Morgan Kaufmann, 1996, 105-112.

11. Datta, P., D. Kibler. Symbolic nearest mean classifiers. – In: Proceeding of AAAI'97, AAAI Press, 1997.
12. Agre, G. An integrated prototype-based learning algorithm. – CIT: Cybernetics and Information Technologiees, Bulgarian Academy of Scienses, **1**, No 1, 2001, 56-70.
13. Weiss, S., C. Kulikowski. Computer Systems That Learn. Morgan Kaufmann, 1990.
14. Duda, R. O., P. E. Hart. Pattern Classification and Scene Analysis. New York, NY: Wiley, 1973.
15. John, G., P. Langley. Estimating continuous distributions in Bayesian classifiers. – In: Proceeding of 11th Annual Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Mateo, 1995.
16. Kohavi, R., B. Becker, D. Sommerfield. Improving simple Bayes. – In: Proceedings of ECML-97.
17. Clark, P., T. Niblett. The CN2 induction algorithm. – Machine Learning, **3**, 1989, 261-283.
18. Cost, S., S. Salzberg. A Weighted nearest neighbour algorithm for learning with symbolic features. – Machine Learning, **10**, 1993, 56-78.
19. Murphy, P., D. Aha. UCI Repository of Machine Learning Databases. 1996. http://www.ics.uci.edu/~mlearn.
20. Sincich, T. Statistics by Examples. Dellen Publishing Comp., 1990.
21. Agre, G. DaMiS: A system for data mining. – IIT Working Papers, IIT/WP-94, Issn 1310-652X, Institute of Information Technologies, 1999.
22. Domingos, P. Unifying instance-based and rule-based induction. – Machine Learning, **24**, 1996, No 2, 141-168.
23. Peev, S. Methods for Discretization of Continuous Attributes and Their Application for Solving Classification Tasks. Ms.S. Thesis, Sofia University, 2002 (in Bulgarian).
24. Li, J., H. Shen, R. Topor. An adaptive method of numerical attribute merging for quantitative association rule mining. – In: Lecture Notes in Computer Science, 1749, Springer, 1999, 41-50.
25. Holte, R. Very simple classification rules perform well on most commonly used datasets. – Machine Learning **11**, 1993, 63-90.
26. Domingos, P. The role of Occam's razor in knowledge discovery. – Data Mining and Knowledge Discovery, **3**, 1999, No 4, 1-19.

Дискретизация със и без класификационна информация

Геннадий Агре , Станимир Пеев***

* *Институт по информационни технологии, 1113 София, email: agre@iinf.bas.bg*

*** Факултет по математика и информатика – Софийски университет, 1000 София*

(Резюме)

В статията се дискутира проблемът за дискретизация на непрекъснати атрибути със или без използване на информация за класа на обучаващите примери. Дискретизацията е една важна предварителна стъпка при подготовката на данни за множество алгоритми от областта на машинното самообучение и извличане на закономерности от данни. За емпиричното сравнение бяха избрани два метода за дискретизация, неизползващи класификационната информация (метод за дискретизиране на равни по дължина интервали и метод за дискретизиране на равни по честота интервали), и един метод за дискретизация, използващ информация за класификационната ентропията. Двата метода са тествани с помощта на два класификационни алгоритъма – прост Бейсов класификатор и класификатор чрез конструиране на прототипи. Резултатите от проведеното изследване върху 13 бази от реални данни не потвърждават широко застъпеното

мнение за превъзходство на ентропийната дискретизация. След анализа на тези резултати са предложени два нови метода за дискретизация с използване на класификационната информация. Проведените експерименти доказват, че предложените методи са по-добри от ентропийната дискретизация и значително подобряват класификационната точност и на двата класификационни алгоритъма.