Evaluating the performance of cost-based discretization versus entropy- and error-based discretization

Peer-reviewed author version

JANSSENS, Davy; BRIJS, Tom; VANHOOF, Koen & WETS, Geert (2006) Evaluating the performance of cost-based discretization versus entropy- and error-based discretization. In: COMPUTERS & OPERATIONS RESEARCH, 33(11). p. 3107-3123.

DOI: 10.1016/j.cor.2005.01.022
Handle: http://hdl.handle.net/1942/1511

Evaluating the performance of Cost-based Discretization versus Entropy- and Error-based Discretization

Davy Janssens
Tom Brijs
Koen Vanhoof
Geert Wets*

E-mail: **{davy.janssens; tom.brijs; koen.vanhoof; geert.wets}@luc.ac.be**

Limburgs Universitair Centrum
Transportation Research Institute
Universitaire Campus
Gebouw D
B-3590 Diepenbeek
Belgium

* Corresponding author. Tel.: +32 (0)11 26.86.49 Fax: +32(0)11 26.87.00

Evaluating the performance of Cost-based Discretization versus Entropy- and Error-based Discretization

Statement of Scope and Purpose

Given its importance, many researchers have already contributed to the issue of discretization in the past. However, to the best of our knowledge, no efforts have been made yet to include the concept of misclassification costs to find an optimal multi-split for discretization purposes. For this reason, this new concept is introduced and explored in this article by means of operations research techniques.

Abstract

Discretization is defined as the process that divides continuous numeric values into intervals of discrete categorical values. In this article, the concept of cost-based discretization as a preprocessing step to the induction of a classifier is introduced in order to obtain an optimal multi-interval splitting for each numeric attribute. Cost-based discretization is particularly useful in the case where the cost of making errors is not equal. A transparent description of the method and the steps involved in cost-based discretization are given. Furthermore, its performance against two other well-known methods, i.e. entropy-based discretization and pure error-based discretization is examined. To this end, experiments on several datasets, taken from the UCI Repository on Machine Learning were carried out. In order to compare the different methods, the area under the Receiver Operating Characteristic (ROC) graph was used and tested on its level of significance. For most datasets the results show that cost-based discretization outperforms entropy- and error-based discretization.

Keywords: Discretization, ROC-curve, cost-sensitive learning

1. Introduction

Discretization is defined as the process that divides continuous numeric values into intervals of discrete categorical values [1]. Given its importance, many researchers have already contributed to the issue of discretization in the past [2-4]. However also in more recent publications, the topic still remains an intriguing research domain [5-7]. Many algorithms which focus on learning decision trees from data, such as C4.5 [8] and CART [9], originally have not been designed to handle continuous numeric attributes very well. These methods are designed to construct decision trees by recursively selecting an attribute to split the instance space in smaller subgroups where, in the case of C4.5, the number of splits per attribute is dependent on the number of distinct attribute values, which for a continuous attribute would result in too many splits. This may lead to overfitting, with less accurate performance of the classifier on unseen data as a result. Therefore, during the construction of the decision tree, continuous attributes are divided into discrete categorical values by grouping some continuous values together. The number of intervals subsequently determines the number of splits per attribute. However, instead of discretizing continuous valued attributes on-the-fly (i.e. during decision tree construction), discretization can also be carried out as a pre-processing step before the induction of the tree. In this case, discretization itself may be considered as a form of knowledge discovery in that critical values in a continuous domain may be revealed [10]. Furthermore, according to Catlett [11], for very large datasets, discretization as a pre-processing step significantly reduces the time to induce a classifier.

To the best of our knowledge, no efforts have been made yet to include the concept of misclassification costs to find an optimal multi-split. This is however very important in the case where the cost of making errors is not equal. Therefore, the objective of this paper is to introduce the concept of cost-based discretization and to evaluate its performance against two other well-known discretization methods, i.e. entropy- and error-based discretization. This paper is organized as follows. In section 2 a brief overview of the existing literature on discretization is provided. From a conceptual point of view, the effectiveness of cost-based discretization in finding the critical cutpoints that minimize an overall cost function is explained in section 3 and the methodology behind cost-based discretization is shown by means of an example. In section 4 an empirical evaluation of these methods is carried out on several datasets, taken from the UCI Repository on Machine Learning [12]. Finally, some conclusions and recommendations for further research are presented in section 5.

2. Discretization Methods

In essence, the process of discretization involves the grouping of continuous values into a number of discrete intervals. However, the decision which continuous values to group together, how many intervals to generate, and thus where to position the interval cutpoints on the continuous scale of attribute values is not always identical for the different discretization methods. Therefore, a brief literature overview of previous research on discretization is presented. This overview can be characterized along five different axes: the type of evaluation function being used, global versus local, static versus dynamic, supervised versus unsupervised and top-down versus bottom-up discretization.

Evaluation Function

Since discretization involves grouping continuous values into discrete intervals, all discretization methods differ with respect to how they measure the quality of the partitioning. Error-based methods, such as for example Maass [13], evaluate candidate cutpoints against an error function and explore a search space of boundary points to minimize the sum of false positive (FP) and false negative (FN) errors on the training set. In other words, given a fixed number of intervals, error-based discretization aims at finding the best discretization that minimizes the total number of errors (FP and FN) made by grouping together particular continuous values into an interval. Entropy-based methods, such as for example Fayyad and Irani [2], are among the most commonly used discretization measures in the literature. These methods use entropy measures to evaluate candidate cutpoints. This means that an entropy-based method will use the class information entropy of candidate partitions to select boundaries for discretization. Class information entropy is a measure of purity and it measures the amount of information which would be needed to specify to which class an instance belongs. It considers one big interval containing all known values of a feature and then recursively partitions this interval into smaller subintervals until some stopping criterion, for example MDLP (Minimum Description Length Principle) [14] or an optimal number of intervals is achieved. Other evaluation measures include Gini, dissimilarity and the Hellinger measure. A detailed description of the method can be found in Fayyad and Irani [2].

Global versus Local Discretization

The distinction between global [15] and local [8] discretization methods is dependent on when discretization is performed. Global discretization handles discretization of each numeric attribute as a

pre-processing step, i.e. before induction of a classifier whereas local methods, like C4.5 carry out discretization on-the-fly (during induction). Empirical results have indicated that global discretization methods often produced superior results compared to local methods since the former use the entire value domain of a numeric attribute for discretization, whereas local methods produce intervals that are applied to subpartitions of the instance space [16].

STATIC VERSUS DYNAMIC DISCRETIZATION

The distinction between static [11,2,3,17] and dynamic [4,5] methods depends on whether the method takes feature interactions into account. Static methods, such as binning, entropy-based partitioning and the 1R algorithm, determine the number of partitions for each attribute independent of the other features. In contrast, dynamic methods conduct a search through the space of possible *k* partitions for all features simultaneously, thereby capturing interdependencies in feature discretization.

SUPERVISED VERSUS UNSUPERVISED DISCRETIZATION

Another distinction can be made dependent on whether the method takes class information into account to find proper intervals or not. Several discretization methods, such as equal width interval binning or equal frequency binning, do not make use of class membership information during the discretization process. These methods are referred to as unsupervised methods [18]. In contrast, discretization methods that use class labels for carrying out discretization are referred to as supervised methods [17,2]. Previous research has indicated that supervised methods are better than unsupervised methods [16].

TOP-DOWN VERSUS BOTTOM-UP DISCRETIZATION

Finally, the distinction between top-down [2] and bottom-up [19] discretization methods can be made. Top-down methods consider one big interval containing all known values of a feature and then partition this interval into smaller and smaller subintervals until a certain stopping criterion, for example Minimum Description Length (MDLP), or optimal number of intervals is achieved. In contrast, bottom-up methods initially consider a number of intervals, determined by the set of boundary points, to combine these intervals during execution until a certain stopping criterion, such as a χ^2 threshold, or optimal number of intervals is achieved.

Later on in the text, we will position our own developed cost-based discretization method within this existing framework. For now, we continue with a conceptual overview of our method.

3. Cost-based Discretization

The objective of our cost-based discretization approach is to take into account the cost of making errors instead of just minimizing the total sum of errors, such as in error-based discretization. By means of the introduction of a misclassification cost matrix, candidate cutpoints are then evaluated against a cost function (instead of an error function) to minimize the overall misclassification cost of false positive and false negative errors. The specification of this cost function is dependent on the costs assigned to the different error types (FP and FN).

In order to understand our contribution of cost-based discretization, it is shown in the next section that the intervals produced by error-based discretization cannot be optimal in a situation where the costs of FP and FN errors are unequal.

3.1 Finding Optimal Cutpoints for Cost-Based Discretization

Suppose we have an attribute A and a binary target variable with class values ‘X’ and ‘Y’. ‘X’ and ‘Y’ have equally sized frequency distributions but the second distribution is shifted in a way that they have a nonempty intersection (see figure 1). Finding the optimal discretization in this case would then involve the identification of all boundary points.

<INSERT FIGURE 1 HERE>

Intuitively, a boundary point is a value *V* in between two sorted attribute values *U* and *W* such that all examples having attribute value *U* have a different class label compared to the examples having attribute value *W*, or *U* and *W* have a different class frequency distribution.

Formally, the concept of a boundary point is defined as: "A value *T* in the range of the attribute *A* is a boundary point if in the sequence of examples sorted by the value of *A*, there exist two examples s_1, $s_2 \in S$, having different classes, such that $val_A(s_1) < T < val_A(s_2)$; and there exists no other example s' $\in$ S such that $val_A(s_1) < val_A(s') < val_A(s_2)$." [20]

Among all identified boundary points (which are not all shown on figure 1 for clarity), $C_1..C_5$ are important candidate cutpoints for error-based discretization. In this example, when the attribute A has a value in the interval [C_1, C_2] or [C_2, C_3] 'X' is the predicted class label, otherwise 'Y'. Therefore, the error-based discretization method, aiming at minimizing the total sum of errors, will merge [C_1, C_2] and [C_2, C_3] into [C_1, C_3] with label 'X', and [C_3, C_4] and [C_4, C_5] into [C_3, C_5] with label 'Y' respectively. However, in the cost-based discretizer, the goal is to minimize the total cost of misclassifications instead of the total sum of errors. In order to calculate this cost, a misclassification cost is assigned to every error type (FP and FN). For instance, assume that misclassifying 'X' is twice as costly as misclassifying 'Y'. In that case, given the candidate cutpoints for error-based discretization, the cost-based discretizer will merge [C_1, C_2], [C_2, C_3] and [C_3, C_4] into [C_1, C_4] due to the fact that the total number of X cases in [C_3, C_4] multiplied by 2 is larger than the number of Y cases in the same interval. The remaining two intervals [C_1, C_4] and [C_4, C_5] will minimize the total misclassification cost, given the positions of the cutpoints.

However, it is clear that the optimal solution for cost-based discretization has not yet been reached. The optimal solution is given by [C_1, a] and [a, C_5] where 'a' is the intersection point where it holds that |ak|=|kl|. In other words, the cost-based discretization technique will select a different boundary point to serve as the cutpoint for the two intervals, namely that particular attribute value after which the misclassification cost of 'X' by predicting the remaining attribute values to belong to class 'Y' is less than the misclassification cost of 'Y'.

3.2 Methodology

In order to illustrate the methodology behind cost-based discretization, in this section a hypothetical example of a continuous numeric attribute with 15 values is considered. The distribution of the different attribute values together with their class values is given in table 1.

<INSERT TABLE 1 HERE>

In a first step, the method will sort the attribute values and will try to identify all boundary points.
For the example cited above, 7 boundary points were determined. The position of the different boundary points is illustrated in figure 2.

<INSERT FIGURE 2 HERE>

These boundary points will serve as potential cutpoints for our final discretization. In previous work [20] it has been proven that it is sufficient to consider boundary points as potential cutpoints, because optimal splits always fall on boundary points.

As stated before, in order to calculate this cost a misclassification cost to every error type (FP and FN) is assigned. For instance, assume that misclassifying 'X' is twice as costly as misclassifying 'Y'. The minimal cost can then be calculated by multiplying the false positive cost (respectively, false negative) by the false positive (respectively, false negative) errors made as a result of assigning one of both classes to the interval and by picking the minimal cost of both assignments. For instance, suppose we want to calculate the minimum cost in the interval 1-6. Assigning the class value 'X' to the interval 1-6 results in 3 errors. The assumption was made that misclassifying 'X' is twice as costly as misclassifying 'Y', so the total cost will be: 3 * 2 = 6. Assigning the class value 'Y' to the interval 1-6 results in 5 errors, so the total cost will be: 5 * 1 = 5. This means that for this interval the minimum cost is 5. The procedure for finding the minimum costs for the other intervals is similar and is shown in table 2. Important to notice however is that for a real-world dataset, it might be difficult to determine exact cost parameters. Therefore, cost values of FP and FN only reflect their relative importance against each other and also may depend on the user's domain knowledge about the problem.

<INSERT TABLE 2 HERE>

The next step will be to set a maximum number of intervals (*n*) and to put the minimum costs of table 2 in a network, whose size depends on the value of *n*. This value is a maximum value and as our method chooses the total minimal cost of the network, the algorithm will still be able to choose less intervals than the number specified by the user.

Suppose that in our example the value of *n* is set to 3, it is then possible to construct a network like the one shown in figure 3 (not all costs are included for the sake of visibility).

<INSERT FIGURE 3 HERE>

The optimisation problem can then be formulated as follows:

$$\text{Minimize } S = \sum_j a_{ij} * x_{ij} + \sum_{j,k \geq j} b_{jk} * y_{jk} + \sum_k c_{kl} * z_{kl}$$

Subject to $\sum_j x_{ij} = 1$ and $x_{ij}; y_{jk}; z_{kl} \in \{0,1\}$

$\sum_k z_{kl} = 1$ $\quad i \in \{1\}$

$\forall j : \sum_{k \geq j} y_{jk} = x_{ij}$ $\quad j \in \{1,..,7\}$

$\forall k : \sum_{j \leq k} y_{jk} = z_{kl}$ $\quad k \in \{1,..,7\}$

$\quad l \in \{7\}$

This is a typical formulation for the shortest path network, which is a well-known problem in operations research [21]. The values x_{ij}, y_{jk} and z_{kl} are Boolean and represent whether the path is chosen or not chosen. The values a_{ij}, b_{jk} and c_{kl} represent the different costs to take a particular path. The position of cutpoints can be determined by solving this shortest path problem by means of integer programming. The actual size of the network for a particular dataset and its corresponding optimisation problem depends on the number of intervals *(n)* and the number of boundary points for the attribute to be discretized.

A full understanding of this methodology enables us now to position our discretization method along the five axes, as they were presented in section 2. The cost-based discretization method presented in this paper, is an error-based, global, static, supervised method combining a top-down and bottom-up approach. However, it is not just an error-based method. As said before, by means of the introduction

of a misclassification cost matrix, boundary points are evaluated against a cost function (instead of an error function) to minimize the overall misclassification cost of false positive and false negative errors instead of just the total sum of errors. It is a global method, since discretization is carried out as a pre-processing step to induction. Furthermore, cost-based discretization is static, since we discretize each attribute separately. It is supervised, since we use class information to find an optimal interval partitioning. Finally, it combines a top-down with a bottom-up approach since all the boundary points are evaluated simultaneously by an integer programming approach.

By increasing the error-cost of a particular class (e.g. class 'X' in the example), the frequency of this class is leveraged so that this can result in different minimum costs and in another positioning of the final cutpoints. Our method should therefore perform better than error-based discretization because this method suffers from a weakness which was identified by Kohavi and Sahami [10], where they showed that the error-based discretization method will never generate two adjacent intervals when in both intervals a particular class prevails, even when the class frequency distributions differ in both intervals. Kohavi and Sahami [10] state that the reason is that two adjacent intervals can always be collapsed into one interval with no degradation in the error.

In the next section, it will be validated whether this theoretical assumption can be verified and whether our method performs better than entropy- and error-based discretization.

4. Empirical Evaluation

4.1 Approach

In our experimental study, we have chosen 7 datasets, taken from the UCI Repository on Machine Learning [12]. Each dataset has several continuous features and the target attribute is always a 2-class nominal attribute. Per dataset, all numeric attributes were discretized separately for different misclassification costs ranging from false positive cost parameter 1 (pure error-based) to 8 (false positive errors are severely punished relative to false negative errors). For the sake of simplicity, this cost parameter is called the *discretization cost*. For the maximum number of intervals (parameter *n*) we have followed the recommendations made by Elomaa & Rousu [22] to keep the value of *n* relatively low. For our experiments we have arbitrarily set the value of *n* to 8. When *n* is not allowed to be too

high, this will have a positive impact on the interpretability of the classification tree after induction, as the tree is prevented from growing too wide. Furthermore, small and narrow trees are less vulnerable to overfitting. In addition, as cost-based discretization finds the total minimum cost of the network, the method is able to choose less intervals than the maximum number specified.

In order to compare the performance of the different methods, we used repeated 10-fold cross validation and induced a C4.5 classifier on the discretized data. C4.5 [8] constructs classification trees by recursively splitting the instance space in smaller subgroups until the subgroup contains only instances from the same class (a pure node), or the subgroup contains instances from different classes (unpure) but the number of instances in that node is too small to be split further. Typically, the tree is allowed to grow its full size after which it is pruned back upwards in order to increase its generalisation power and to reduce overfitting. In contrast to CART [9], which produces binary splits on the attributes, C4.5 creates multiple branches per split, i.e. one for each interval after discretization of that attribute.

Per method, 8 models were built, by increasing the FP cost, as well from 1 to 8. This parameter is called the *misclassification cost*. It should be clear for the reader that a higher discretization cost results in a different position of the final cutpoints, while a higher FP misclassification cost will result in a lower FP error rate (equivalent with a higher TN rate) and in a higher FN error rate (equivalent with a lower TP rate). The FP error rate and the TP rate will be used to evaluate the different methods. However, as explained before, both (discretization and misclassification cost) are introduced to cope with situations where the cost of making errors is not equal.

Varying the class misclassification cost in the cost matrix, will allow us to define for each inducer a Receiver Operating Characteristic (ROC) curve [23]. ROC analysis uses what is called a ROC space to give a graphical representation of the classifiers performance independently of class distributions or error costs. This ROC space is a coordinate system where the rate of true positives is plotted on the Y-axis and the rate of false positives is plotted on the X-axis. The true positive rate is defined as the fraction of positive cases classified correctly relative to the total number of positive examples. The

false positive rate is defined as the fraction of negative cases classified erroneously relative to the number of all negative examples.

Our ROC curves, were averaged over the ten train and test partitions. From a visual perspective, one point in the ROC curve (representing one classifier with given parameters) is better than another if it is located more to the north-west (TP is higher, FP is lower or both) on the ROC graph [23]. For our cost-based method, we have chosen, for the sake of visibility, to represent the classifier, with its corresponding discretization cost, which performs best. Furthermore, statistical hypothesis testing was applied to compare the relative performance of the different models. A detailed procedure about how this was done, is described in the next section.

4.2 Comparing ROC Curves

The difficulty of comparing several ROC curves is that, generally speaking, one ROC curve does not completely dominate another (the first curve does not lie entirely above the second one), but intersects at one or more points.

This is shown in figure 4 for the Bupa liver disorders dataset by means of example. ROC curves for the error, entropy and cost-based discretization methods were represented in the figure, since these are the methods under evaluation, along with the alternative of not discretizing prior to induction. In the latter case, discretization is of course carried out while inducing the C4.5 classifier.

<INSERT FIGURE 4 HERE>

To be able to compare the performance of different classifiers with ROC curves measured on the same data, a single number measure which reflects the performance of the classifiers is needed. The area under the ROC curve (AUC) is generally accepted as the preferred single number measure. Because random guessing produces the diagonal line between (0,0) and (1,1), which has an area of 0.5, no realistic classifier should have an AUC less than 0.5. Trapezoidal integration was used to calculate the AUC, according to the formula [24]:

AUC= $\sum_i \left\{ \left(1 - b_i \cdot \Delta a\right) + \frac{1}{2}\left[\Delta\left(1-b\right) \cdot \Delta a\right] \right\}$, where a =FP-rate; 1- b = TP-rate;

$\Delta\left(1-b\right) = \left(1-b_i\right) - \left(1-b_{i-1}\right)$ and $\Delta a = a_i - a_{i-1}$

For the example shown above, the AUC was respectively 0.6661, 0.6199, 0.6974 and 0.7207 for the not discretized, entropy-, error- and cost-based discretization options.

In order to compare classifiers, it is necessary to estimate the standard error of the area under the curve, SE(AUC). The method for doing this, which is applicable to an empirically derived curve, is to use the standard error of the Wilcoxon statistic, SE(W) [24]:

$$\text{SE (AUC)} \approx \text{SE (W)} = \sqrt{\frac{q(1-q) + (C_p - 1)(Q_1 - q^2) + (C_n - 1)(Q_2 - q^2)}{C_p C_n}} \text{, where} \quad (1)$$

θ is the area under the curve, C_p and C_n are the number of positive and negative examples respectively, and $Q_1 = \frac{q}{(2-q)}$ and $Q_2 = \frac{2q^2}{(1+q)}$.

The SE (AUC) for the bupa liver disorders dataset was respectively 0.0299, 0.0308, 0.0291 and 0.0283 for the different discretization alternatives.

To assess whether the differences between the AUCs computed from the same data set are statistically significant, hypothesis testing can be employed. Hanley & McNeil [25] define the following test statistic:

$$Z = \frac{AUC_1 - AUC_2}{\sqrt{se_1^2 + se_2^2 - 2r \cdot se_1 \ se_2}} \text{, where} \quad (2)$$

se_1 and se_2 are the standard errors (Equation 1) for AUC_1 and AUC_2 respectively, and r is a value which represents the correlation between the two areas.

One should take into account this correlation coefficient because when computed from the same data, AUC_1 and AUC_2 are very likely to be correlated. The value r is a function of the average value of two intermediate correlation coefficients and of the average areas. The intermediate coefficients are the correlations between the two classifiers' certainty values for objects with negative decision and positive

decision, respectively. These coefficients can be computed using Kendall's (τ) measure of correlation [26]. For a tabulation of *r*, we refer to Hanley & McNeil [25].

Z is standard normally distributed under the hypothesis that the two areas are equal, and can be used to test -under a certain level of significance- whether the two areas are statistically likely to be different. Therefore, one should calculate the critical value of Z and depending on the selected significance level α, reject or not reject the hypothesis that both areas are equal. The Z-values for the bupa liver disorders dataset were respectively 2.5505, 4.6161 and 1.107 for the comparison of the cost-based discretization method with the not discretized, entropy- and error-based discretization options. In our discussion of the results (see section 4.3), p-values were used to determine whether different areas are statistically significant. The p-values for the example shown above were respectively 0.011, 3.98E-06 and 0.268 for the different comparisons. The null hypothesis that both areas are equal was rejected when the statistical test showed a p-value below 0.05.

4.3 Discussion of the results

According to the logic presented above, the empirical results for all the datasets are summarized in table 3. In order to validate whether the differences between the different areas under the ROC-graph for the classifiers are statistically significant, pairwise comparisons were conducted. When the difference between AUC_1 en AUC_2 shows a positive sign, this means that the area under the ROC curve for the first method is larger than the area under the ROC curve for the second method under consideration. The opposite is true for negative signs. One method can only said to be better than another if the level of significance (<0.05) is reached. In these cases, the p-values were indicated in bold.

<INSERT TABLE 3 HERE>

GLOBAL RESULTS

Since we are especially interested in evaluating the performance of cost-based discretization against the other discretization methods, our main focus should be on the right-hand side of table 3 (last three columns). At first glance, the results appear to reveal some interesting insights. As we can see, for

cost-based discretization, 8 times (out of 10) cost-based discretization has proven to be significantly better than the other discretization methods. This is a very good result, all the more because the other discretization methods were not able to achieve a similar number. Error-based discretization outperformed only in 2 times (out of 6), and entropy-based discretization and not discretizing prior to induction did only slightly better by dominating in 4 times (out of 10). Furthermore, only in 2 times out of 21 observations (i.e. the Pima and Euthyroid dataset), cost-based discretization is dominated by another discretization method.

RESULTS PER DATASET

Another possibility is to have a look at the results per dataset. For the Australian (Australian Credit Screening) dataset, cost-based discretization dominates the entropy and the error-based methods, but cost-based discretization was not able to show a significant difference (neither better, nor worse) with respect to the option of not discretizing. For the Bupa (Bupa liver disorders) dataset, cost-based discretization performs remarkably better than Entropy-based discretization (extremely low p-value). The reason is that the latter only discretized one of the six attributes and for all the other attributes collapsed their attribute values into a single interval. This proved not to be a good approach, because a lot of valuable classification information incorporated in the other variables is therefore lost. Unfortunately, we were unable to prove that for this dataset cost-based discretization does significantly better than Error-based discretization. However, there is still a significant difference with the option of not discretizing prior to induction. From the Breast (Breast Cancer Wisconsin) dataset, we can learn that cost-based discretization performs significantly better in relation to all tree methods under evaluation. The Cleve (Cleveland Heart Disease) dataset does not yield any statistical significant differences between the different classifiers. Therefore, none of the methods really prevails for this dataset. For the Ionosphere dataset all three discretization methods have proven to be extremely significant compared to the option of not discretizing prior to induction. This is of course because the latter yield very poor results since FP- versus TP-combinations are located very much to the north-east. Quite a remarkable result from our research is that the Pima dataset (Pima Indian Diabetes) is the only dataset in which Entropy-based discretization clearly dominates the other methods. Finally, there is only one dataset (Euthyroid dataset) in which local discretization generates better results than

discretization prior to induction. This confirms our statement that dealing with discretization as a pre-processing step can significantly improve a classifier's performance.

As cost-based discretization only just missed the minimum level of significance for the Pima dataset (p-value=0.051, versus error-based discretization), also in this dataset our method shows to be a legitimate second best alternative.

DISCRETIZATION COSTS DETERMINE CLASSIFIERS PERFORMANCE

It was already shown in figure 4 that the classifier with a low discretization cost performs best for the Bupa liver disorders dataset. A similar pattern can be found for the other datasets as well. This is shown in table 4. This table shows the discretization cost which achieves the largest AUC for the cost-based discretization method.

<INSERT TABLE 4 HERE>

Important to notice is that there is never a discretization cost higher than 4 which leads to the best results for the cost-based discretization method. This can be explained by the fact that applying a high error cost to a particular class, actually leverages the frequency of that class excessively, as this class is considered to be more important (due to the high cost assigned to it). When a particular class is excessively leveraged, this will of course lead to a less appropriate position of the actual cutpoints, and finally also to a poorer performance of the C4.5 classifier.

5. Conclusion

In this article, the concept of cost-based discretization was introduced. The method was empirically evaluated against two other important discretization methods, i.e. entropy and error-based discretization. Validation of the cost-based discretization approach was carried out on several UCI repository datasets. After the datasets were discretized, ROC analysis was used to evaluate the performance of the different classification trees. To be able to make a valid assessment which method performs best, the area under the ROC curve and p-values were used as criteria to reflect the performance of the classifier.

Although cost-based discretization did not dominate other discretization methods all along the line, the empirical results showed that for most datasets cost-based discretization outperformed entropy and error-based discretization. Furthermore, it was shown and explained why the best results for the cost-based discretization method are usually obtained with relatively low discretization costs. Finally, we were able to confirm the statement that global discretization produces better results than local discretization.

Further research is still needed to better understand why it is not always the same discretization cost that performs best over the datasets. The fact that class distributions differ significantly for the different datasets and that different patterns may be incorporated in the datasets are plausible explanations but further research should still validate this.

References

[1] Lee C., Shin D-G. A Context-Sensitive Discretization of Numeric Attributes for Classification Learning. Proceedings of the Eleventh European Conference on Artificial Intelligence. Amsterdam: John Wiley & Sons, 1994. p. 428-432.

[2] Fayyad U., Irani K. Multi-interval discretization of continuous-valued attributes for classification learning. Proceedings of the Thirteenth Int. Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann, 1993. p. 1022-1027.

[3] Pfahringer B. Compression-based discretization of continuous attributes. Proceedings of the Twelfth International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 1995. p. 456-463.

[4] Fulton T., Kasif S., Salzberg S. Efficient algorithms for finding multi-way splits for decision trees. Proceedings of the Twelfth Int. Conference on Machine Learning. San Francisco: Morgan Kaufmann, 1995. p. 244-251.

[5] Bay S. D. Multivariate Discretization for Set Mining. Knowledge and Information Systems 2001; 3(4): 491-512.

[6] Elomaa T., Rousu J. Preprocessing opportunities in optimal numerical range partitioning. Proceedings of the First IEEE International Conference on Data Mining. IEEE Computer Society Press, 2001. p. 115-122.

[7] Cantú-Paz E. Supervised and Unsupervised Discretization Methods for Evolutionary Algorithms. Proceedings Genetic and Evolutionary Computation Conference 2001. San Francisco: Morgan Kaufmann, 2001. p. 213-216.

[8] Quinlan J.R. C4.5: Programs for Machine Learning: Los Altos : Morgan Kaufmann, 1993.

[9] Breiman L., Friedman J.H., Olshen R.A., Stone C.J. Classification and Regression Trees. Belmont, CA: Wadsworth, 1984.

[10] Kohavi R., Sahami M. Error-based and Entropy-Based Discretization of Continuous Features. Proceedings of the Second Int. Conference on Knowledge & Data Mining. Menlo Park: AAAI Press, 1996. p. 114-119.

[11] Catlett J. On changing continuous attributes into ordered discrete attributes. Proceedings of the Fifth European Working Session on Learning, Berlin: Springer-Verlag, 1991. p. 164-178.

[12] Blake C.L., Merz C.J. UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[13] Maass W. Efficient agnostic PAC-learning with simple hypotheses. Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory. New York: ACM Press, 1994. p. 67-75.

[14] Rissanen J. Stochastic Complexity in Statistical Inquiry, World Scientific, 1989.

[15] Chmielewski M.R., Grzymala-Busse J.W. Global discretization of continuous attributes as preprocessing for machine learning. In Third International Workshop on Rough Sets and Soft Computing, 1994. p. 294-301.

[16] Dougherty J., Kohavi R., Sahami M. Supervised and unsupervised discretization of continous features. Proceedings of the Twelfth Int. Conference on Machine Learning. San Francisco: Morgan Kaufmann, 1995. p. 194-202.

[17] Holte R. Very simple classification rules perform well on most commonly used datasets. Machine Learning 1993; 11: 63-90.

[18] Van de Merckt T. Decision Trees in Numerical Attributes Spaces. In Proceedings of the Thirteenth Int. Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1993. p.1016-1021.

[19] Kerber R. Chimerge: Discretization of numeric attributes. In Proceedings of the Tenth Nat. Conference on Artificial Intelligence, MIT Press, 1992. p. 123-128.

[20] Fayyad U., Irani K. On the handling of continuous-valued attributes in decision tree generation. Machine Learning 1992; 8: 87-102.

[21] Hillier F., Lieberman G. Introduction to Operations Research, 6th edition, McGraw Hill, New York, 1995.

[22] Elomaa T., Rousu J. Finding Optimal Multi-Splits for Numerical Attributes in Decision Tree Learning, Technical Report, NC-TR-96-041, University of Helsinki, 1996.

[23] Provost F., Fawcett T. Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. Proceedings of the Third Int. Conference on Knowledge Discovery and Data Mining. Menlo Park: AAAI Press, 1997. p. 43-48.

[24] Bradley A.P. The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. Pattern Recognition 1997; 30 (7): 1145-1159.

[25] Hanley J.A., McNeil B.J. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. Radiology 1983; 148: 839-843.

[26] Kendall M.G. A new measure of rank correlation. Biometrika 1938; 30: 81-92.

List of Figures and Tables

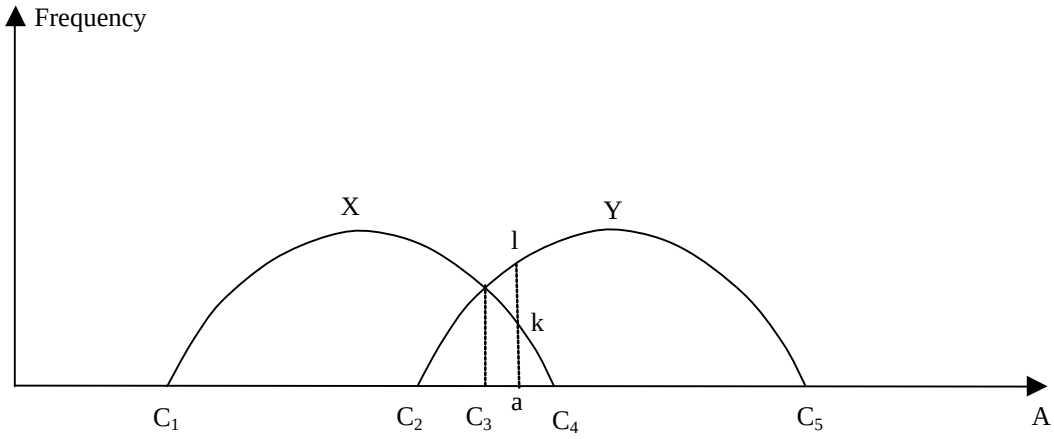

Figure 1: Cutpoints and class distribution for a continuous attribute A

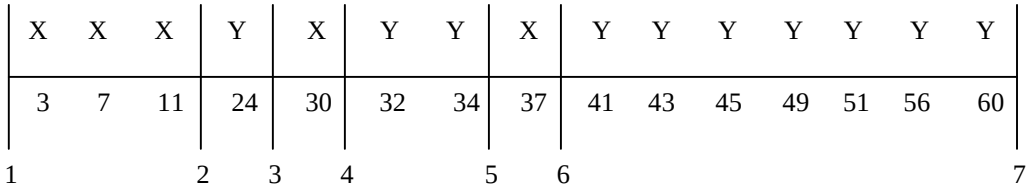

Figure 2: Sorted attribute values and distribution of class values with possible boundary points

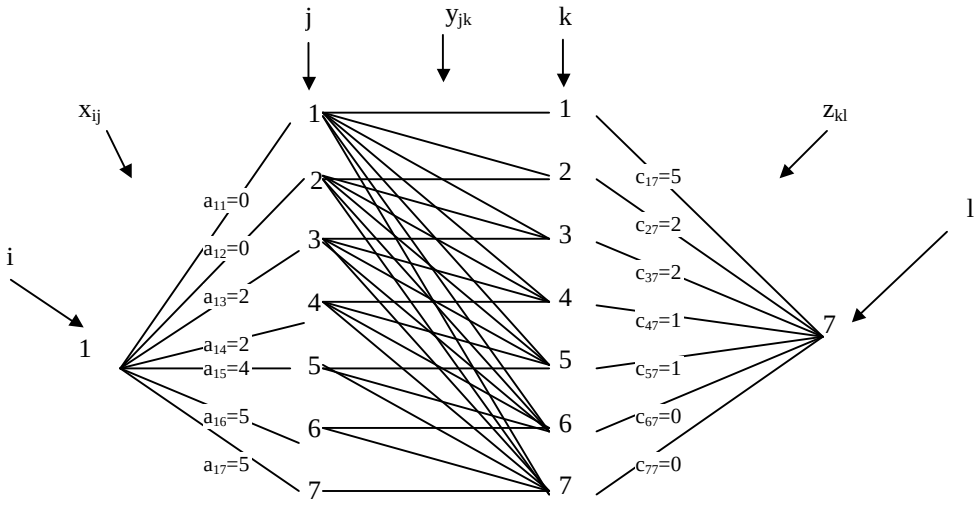

Figure 3: Shortest route network

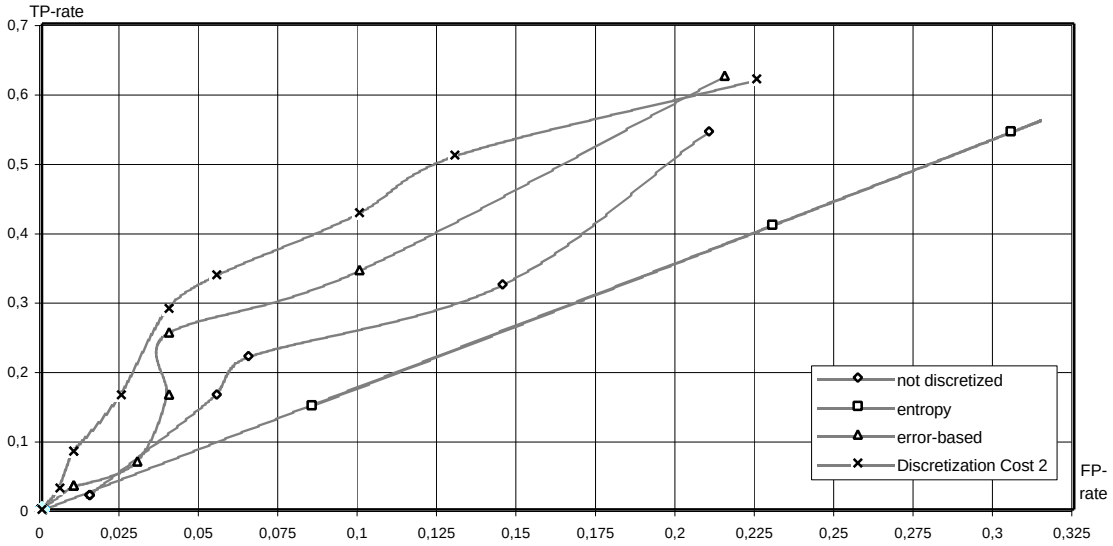

Figure 4: ROC-curve for the Bupa liver disorders dataset

Table 1: Example of cost-based discretization

Attribute values	Class values	Attribute values	Class values	Attribute values	Class values
49	Y	51	Y	60	Y
37	X	3	X	32	Y
41	Y	7	X	34	Y
11	X	43	Y	30	X
24	Y	56	Y	45	Y

Table 2: Intervals with the corresponding minimum costs

Interval	Min.Cost	Interval	Min. Cost	Interval	Min. Cost
1-2	0	2-4	1	3-7	2
1-3	2	2-5	1	4-5	0
1-4	2	2-6	2	4-6	1
1-5	4	2-7	2	4-7	1
1-6	5	3-4	0	5-6	0
1-7	5	3-5	1	5-7	1
2-3	0	3-6	2	6-7	0

Table 3: Overview of the pairwise comparisons for all datasets

	Entropy vs Not discretized		**Error vs Not discretized**		**Entropy vs Error**		**Cost vs Not discretized**		**Cost vs Entropy**		**Cost vs Error**	
	AUC1-AUC2	***p-value***	***AUC1-AUC2***	***p-value***	***AUC1-AUC2***	***p-value***	***AUC1-AUC2***	***p-value***	***AUC1-AUC2***	***p-value***	***AUC1-AUC2***	***p-value***
Australian	-0.01	0.25	-0.0063	0.46	-0.0037	0.68	0.01	0.20	0.02	**0.02**	0.02	**0.045**
Bupa	-0.05	**0.04**	0.03	0.15	-0.08	**4.48E-04**	0.05	**0.01**	0.10	**3.98E-06**	0.02	0.27
Breast	-0.0033	0.54	0.002	0.70	-0.005	0.32	0.01	**0.02**	0.02	**0.003**	0.009	**0.044**
Cleve	-0.0055	0.82	0.0033	0.89	-0.009	0.71	-0.007	0.75	-0.002	0.93	-0.01	0.64
Ionosphere	0.17	**2.14E-08**	0.19	**1.14E-10**	-0.02	0.35	0.18	**4.51E-09**	0.007	0.78	-0.01	0.52
Pima	0.03	**0.006**	-0.02	0.22	0.05	**7.65E-05**	0.009	0.47	-0.02	**0.04**	0.03	0.051
Euthyroid	-0.03	**0.01**	-0.03	**0.01**	0.0006	0.96	-0.05	**2.06E-05**	-0.02	0.06	-0.02	0.071

Table 4: The discretization cost for the cost-based discretization method which performs best per dataset

Dataset	Breast	Credit	Euthyroid	Ionosphere	Cleve	Pima	Bupa
Discretization cost	4	3	4	4	3	4	2