Improved Algorithms for Univariate Discretization of Continuous Features

Jussi Kujala and Tapio Elomaa

Institute of Software Systems
Tampere University of Technology
P. O. Box 553, FI-33101 Tampere, Finland
jussi.kujala@tut.fi elomaa@cs.tut.fi

Abstract. In discretization of a continuous variable its numerical value range is divided into a few intervals that are used in classification. For example, Naïve Bayes can benefit from this processing. A commonly-used supervised discretization method is Fayyad and Irani's recursive entropy-based splitting of a value range. The technique uses MDL as a model selection criterion to decide whether to accept the proposed split. We argue that theoretically the method is not always close to ideal for this application. Empirical experiments support our finding. We give a statistical rule that does not use the ad-hoc rule of Fayyad and Irani's approach to increase its performance. This rule, though, is quite time consuming to compute. We also demonstrate that a very simple Bayesian method performs better than MDL as a model selection criterion.

1 Introduction

A common way of handling continuous information — such as *weight* and *volume* of an object — in classifiers is to discretize the variable's value range. Discretization produces typically disjoint intervals that mutually cover the continuous value range of the attribute. Some classifiers, like Naïve Bayes (NB), actually prefer information that composes of parts that have only few possible values [1, 2]. We consider the supervised setting; i.e., a learning algorithm has access to a labeled *training set* $S = \{ (x_1, y_1), \ldots, (x_n, y_n) \}$, where instance x_i is composed of the feature values and y_i is the class label of example i. *Univariate* approaches consider one independently measured attribute at a time, while *multivariate* approaches take several (usually all) attributes into account simultaneously.

The literature on discretization algorithms is vast (see e.g., [1, 3–5] and the references therein). Many univariate and multivariate discretization algorithms have been proposed. Fayyad and Irani's [6] entropy-based discretization algorithm is arguably the most commonly used supervised discretization approach. In addition to entropy calculation the method also takes advantage of the *minimum description length* (MDL) principle, so we will call this algorithm ENT-MDL. The main reasons for the success of ENT-MDL are probably its comprehensibility and quite good performance. The other most popular discretization techniques are unsupervised approaches equal-width binning (EWB) and equal-frequency binning [7, 8, 1].

Fayyad and Irani's approach is based on recursive binary splitting of the (sub)interval at the point that appears the most promising according to the entropy measure. Whether to actually implement the suggested split is tested using a MDL model selection criterion. In this paper we show that Fayyad and Irani's MDL rule is not optimal in discretization and that it is not sound. Replacing it with a Bayesian criterion leads to an algorithm that work as well, if not better. In addition, we propose a well-founded test statistic that performs very well in practice without any ad-hoc rules attached to it. This test statistic, though, is expensive to compute.

The remainder of this paper is organized as follows. Section 2 reviews the background of this work — Naïve Bayesian classifier and discretization of continuous features. In Section 3 we recapitulate Fayyad and Irani's [6] ENT-MDL algorithm in more detail and consider its theoretical and practical properties. We then propose to replace the MDL model selection criterion with a simple Bayesian one. Section 5 puts forward a test statistic to decide on splitting. This approach does not need any ad hoc techniques to support it. In Section 6 we report on an empirical evaluation of the techniques discussed in this work. The experiments confirm that the straightforward Bayesian rule slightly outperforms the MDL rule and the test statistic can match the performance of both of these heuristics. Finally, Section 7 presents the concluding remarks of this paper.

2 Related Work and Approaches to Discretization

In general, a classifier associates a feature vector x with a class label y. Values in x are information measured from an object and y is the identity of the object that we are interested in. A *discrete* feature has a finite number of possible values, while a *continuous* feature can attain values in some infinite totally ordered set. For example, the weight of an object can attain values in the set of positive real numbers $\mathbb{R}^+$.

In this section we first recapitulate the Naïve Bayes classifier. It is a simple and effective classifier for discrete features. Naïve Bayes gives us a motivation for discretization of continuous features. We, then, briefly review previous work on discretization.

2.1 Our Motivation: Naïve Bayes Classifier

Naïve Bayes classifier uses the training set to infer from the given features x the label y we want to know. It assumes that the feature-label pairs (x, y) in the training set have been generated independently from some distribution D. NB takes advantage of *Bayesian inference* in labeling:

$$\mathbf{P}(y \mid x) \propto \mathbf{P}(x \mid y)\, \mathbf{P}(y)\,.$$

The naïvity in NB is to assume that different features in $x = (x_1, \ldots, x_d)$ are statistically independent given the class:

$$\mathbf{P}(x \mid y) = \mathbf{P}(x_1, \ldots, x_d \mid y) \approx \mathbf{P}(x_1 \mid y) \cdots \mathbf{P}(x_d \mid y)\,.$$

This simplification enables it to avoid *the curse of dimensionality*, the fact that the number of samples needed to estimate a joint distribution of several features grows exponentially in their number. Under the independence assumption we only estimate the marginal distribution of each feature, and these densities do not depend on the number of features. The trade-off is that the independence is unlikely to hold which may lead to decreased accuracy in classification.

The empirical performance of Naïve Bayes classifier has, nevertheless, been shown to be good in several experiments [1, 9]. It appears that the assumption that features are independent does not necessarily hinder the performance even when false [10]. Domingos and Pazzani [11] have argued why this is so.

2.2 Related Work on Discretization

Naïve Bayes needs to know for each feature x_i the probability of attaining a particular value v, $\mathbf{P}(x_i = v \mid y)$. For discrete features the conditional probability can be easily estimated from the training set by counting the number of labels y for which it holds that $x_i = v$. For continuous features it is an interesting question how to choose these probabilities given the training set. This problem has attained significant attention. For a comprehensive survey of the associated research see [3]. Here we only review work that is most related to ours.

For a classifier the most fundamental aim of discretization is to place the interval borders so that its predictive power is good on yet unseen examples. In discretization we could consider all features simultaneously and, for example, minimize the empirical error rate on the training set. Unfortunately multidimensional empirical error minimization is NP-complete [12–14] although polynomial time *approximation algorithm* exists [14, 15]. In general the methods for multivariate discretization are computationally expensive.

Hence simpler univariate discretization methods are actually used. Moreover, Naïve Bayes is in some sense inherently univariate, because of the assumption of the statistical independence between features. For example, Figure 1(b) demonstrates a situation in which neither of the available attributes can clarify class distribution and multivariate discretization would be beneficial. However, Naïve Bayes cannot take advantage of multivariate discretization because the marginal distributions are mixed.

Early continuous feature handling in NB assumed that each feature conforms separately to some probability distribution — e.g., normal distribution [16]. The necessary parameters were then estimated from the training set. However, sometimes features are not distributed as assumed and then the performance suffers. A continuous feature can be binned to intervals of equal width, reducing the continuous-valued estimation to a discrete one. From a statistical point of view this models a continuous feature with a piecewise uniform distribution, where

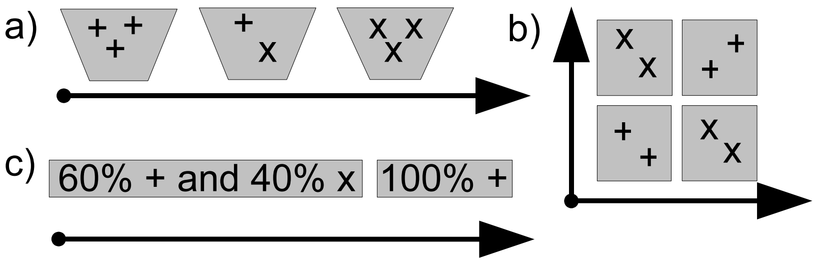

Fig. 1. The class labels are + and x. Subfigure a) shows how EWB can make suboptimal choices and b) depicts a case where univariate methods fail. Naïve Bayes cannot either take advantage of the best split in this case. In situation c) empirical error minimization fails to distinguish between two adjacent distributions, because their majority class is the same.

each uniform distribution corresponds to an interval. This is more flexible than using a more limited distribution, especially if the number of intervals can depend on the training set.

Figure 1(a) demonstrates that the unsupervised EWB is sometimes suboptimal. There is a slight performance drop if the label distribution suddenly changes "within an interval". Hence several methods have been invented to place the interval borders in a more intelligent way [7, 17, 5].

Catlett [8] proposed to apply recursive partitioning based on entropy of the observed label distribution of a discretized feature. Intuitively the entropy measures the amount of randomness of a source producing random items. In this approach an interval is split at a point that results in minimum entropy. Formally, let $\widehat{P}_I(y_i)$ be the *empirical probability* of observing the label y_i on interval I; i.e., the ratio of labels y_i to all labels in the interval I. Then the entropy of the label distribution on I is defined as:

$$H(I) = \sum_{y_i} \widehat{P}_I(y_i) \log_2 \frac{1}{\widehat{P}_I(y_i)},$$

where the sum is over all labels. The entropy of the label distribution on a feature is the sum over all intervals:

$$H(S) = \sum_{I} \frac{|I|}{|S|} H(I),$$

where $|I|$ is the number of examples in the interval I and $|S|$ is the total number of examples in the training set S.

Several heuristic rules were used to decide when to stop the recursive partitioning in Catlett's [8] approach. In ENT-MDL Fayyad and Irani [6] proposed to use a single MDL-based stopping rule. We proceed to review ENT-MDL in more detail. It is based on modeling the assumed true distribution on a feature as accurately as possible. This is in contrast to the empirical error minimization, which must be regulated, e.g., by restricting the number of final intervals and can lose information of the distribution. The problem with error-based discretization is that it cannot separate two adjacent intervals that have the same majority class, even though it might be beneficial for further processing in a classifier (see Figure 1(c)) [18].

3 Fayyad and Irani's Recursive Discretization: ENT-MDL

For a given training set we have two somewhat distinct problems:

1. How many intervals to use?
2. How to place the intervals?

ENT-MDL uses a MDL criterion to answer the first question and entropy to answer the second one. Minimizing the entropy of the label distribution for a fixed number of intervals yields a discretization in which, intuitively, the empirical label distribution is as unsurprising as possible. However, no efficient method for minimizing entropy for a feature is known. ENT-MDL uses a *heuristic*: given an interval it splits it at the point that minimizes the joint entropy of the two resulting subintervals. This heuristic is applied recursively. To address the issue of the total number of intervals Fayyad and Irani suggest that a test be done whether to actually execute a split.

This test solves a model selection problem where the candidate models are:

M_0 labels on the interval are generated independently from the same distribution.

M_i there is a distribution for the instances up to the index i, $i > 0$, and a separate one for the instances after that. The labels are generated independently.

In this case a model M_i that splits the data has always more explanatory power on the training set than M_0 which refrains from splitting. This behavior is an example of overfitting, because a more complicated model can fit the training data very well, but may not have any predictive power on instances it has not seen. ENT-MDL uses MDL to choose the model. In short, MDL selects the model that makes it possible to compress the data — in this case the class labels — the most. The compression used in ENT-MDL is a *two part code*. The first part codes the model and the second part codes the data.

More precisely, Fayyad and Irani encode the data on any interval with approximately $|I|\,H(I)$ bits using an optimal code [19]. Then the model M_0 is encoded with $k\,H(I)$ bits, where k is the number of labels on the interval I. Thus, the total bit length of the data and the model is

$$(|I| + k)H(I).$$

Similarly the models $M_i, i > 0$, are encoded with

$$k_1 H(I_1) + k_2 H(I_2) + \log_2(3^k - 2) + \log_2(|I| - 1)$$

bits, where I_1 is the first subinterval and k_1 is the number of distinct labels on it. I_2 and k_2 are defined similarly. The additional terms follow from the fact that there are two intervals.

3.1 Theoretical and Empirical Properties of ENT-MDL

The code lengths proposed by Fayyad and Irani [6] do not derive from real codes, because for example we cannot encode the model M_0 within the bits given. They suggest that M_0 is sent for example as a codebook for a Huffman code that codes each label individually. First, the expected code word length $H(I)$ for a label is different from the usual arithmetic average of the code word lengths. For example, if we have two labels with codes 1 and 0, i.e., two bits, and the probability of the former is 0.9, then $kH(I) \approx 0.94$. Second, if the item labels are coded individually, then the sum of the code word lengths for the data can be $|I|$ bits greater than $|I|H(I)$, because there are no fractional bits.

Instead we need to use a non-universal nearly optimal code for sequences, like the arithmetic code or a Huffman code that encodes sequences, and these codes in general need to know the probabilities on the labels. For $|I|$ items and k labels there are

$$P = \binom{|I| + k - 1}{k - 1}$$

different sets of probabilities (the number of ways we can allocate $|I|$ items to k bins). Hence, on average a single set of probabilities takes $\log_2 P$ bits. This, then, is a lower bound for the length of the model M_0 unless we have some additional *a priori* knowledge on the probabilities or use an approximation of the model M_0.

In general, a problem with the accurate use of two part MDL is that the user is relied on giving the code, and an optimal code for the application may be difficult to come up with.

The performance of ENT-MDL increases if it splits the range of a given feature at least once. We call this property *autocutting* and denote this method by ENT-MDL-A. It is unclear to us whether Fayyad and Irani [6] meant that autocutting should be always done with ENT-MDL. Clearly, if the rule used to determine whether to split were approximately optimal, then this kind of behavior would be unnecessary. In Section 6 (Tables 1 and 2) we see that autocutting is empirically beneficial, because it increases the average prediction accuracy and the increase is statistically significant in four test domains. Furthermore, the average accuracy does not decrease significantly in any of our test domains.

Let us give an example of a situation in which ENT-MDL makes a wrong choice.

Example 1. Let an interval I have n examples that have binary class — either 0 or 1 — and there are equal numbers of instances from both classes. The first half I_1 of I contains 30% of the 1s and the second half I_2 contains 70% of them. The entropies of these intervals are $H(I) = 1$ and $H(I_1) = H(I_2) \approx 0.88$. Let $\mathcal{H}_0$ be the hypothesis that the labels I are generated uniformly and $\mathcal{H}_1$ the hypothesis that the distribution changes at some point; i.e., we should split. ENT-MDL chooses $\mathcal{H}_0$ if the following holds:

$$\left(1 + \frac{2}{n}\right) n < 2 \cdot 0.88 \cdot \frac{n}{2} \left(1 + \frac{2}{n/2}\right) + \log_2(n-1) + \log_2(7).$$

If we use accurate values then if $n > 91$ ENT-MDL chooses to split, and does not split when $n = 91$. The probability that there is at least this discrepancy between entropies $H(I)$ and $H(I_1), H(I_2)$ is approximately of the order 0.35% *if* $\mathcal{H}_0$ is true with the label probabilities approximated from the empirical label frequencies. We estimated this probability by generating 100 000 intervals of the length n from $\mathcal{H}_0$ and computing the entropies of the intervals according to $\mathcal{H}_0$ and $\mathcal{H}_1$. Note that $\mathcal{H}_1$ always chooses the best split for the given generated labels. We then compared the difference of these entropies to those from the original data. In only 353 cases the difference was larger. Hence we should not choose $\mathcal{H}_0$, because the labels are not typical for it when compared to the hypothesis $\mathcal{H}_1$. This example is also valid if we consider MDL where the rule is based on a real code, as discussed above.

Note that we implicitly assume that we can approximate the real $\mathcal{H}_0$ with the one estimated from the item labels on the interval I. This does not affect our results to a great extent, because $\mathcal{H}_0$ is a simple hypothesis, hence it overfits only slightly. We also tested altering the frequencies for $\mathcal{H}_0$ and observed that the results were the same.

In experiments this kind of case appears to happen for example in the UCI Bupa Liver Database. For this domain a decrease of 6.2 percentage units in prediction accuracy results when ENT-MDL is used instead of ENT-MDL-A. We verified this behavior by manually checking for this domain that MDL did not split when it really should have.

4 Simple Bayesian Methods in Discretization: ENT-BAY

Let us study replacing the MDL criterion used in ENT-MDL with a Bayesian method. Bayesian model selection is a well known and much used tool. It unifies formal reasoning and intuitive prior knowledge of the user in a convenient manner. Given models $M_0, \ldots, M_n$ the Bayesian approach selects the model M_i that maximizes the posterior probability of having generated the data S:

$$\mathbf{P}(S \mid M_i)\,\mathbf{P}(M_i)\,,$$

where $\mathbf{P}(M_i)$ is the *a priori* probability of the model M_i given by the user. Two part MDL can be seen as a special case of the Bayesian approach in which $\mathbf{P}(M_i)$ is obtained from the code length for the model M_i.

In our case, the model M_0 corresponds to the no-split decision and models M_i with $i > 0$ correspond to the cases where the interval is split at instance with index i. We can, of course, set the priors in several ways. In subsequent empirical evaluation we study the following straightforward way. We assign a prior 0.95 to M_0 and the remaining probability mass is divided evenly to the other models. We call this method ENT-BAY95.

Having to assign the priors is both an advantage and a drawback. Priors offer flexibility, because they are intuitive and user can set them according to the needs of the problem. On the other hand, there are no true priors and selecting them can be a nuisance. It is worth noting that Fayyad and Irani [6] too consider a Bayesian test, but prefer MDL, because they view the selection of priors to be too arbitrary. We show in our empirical studies that the simple prior given above performs well in all tested problems. Thus, it can be used if the user does not wish to select the prior himself. If the user chooses to customize the prior distribution, then we presume that the results would be even better.

5 Using a Test Statistic to Decide on the Splits

A problem with the discretization schemes described above is that they can be improved with the ad-hoc technique of autocutting. This means that when used as such the schemes do not work as well as they should.

We demonstrate an alternative approach to decide whether to split: using a *test statistic* derived from the data. In statistics using such an approach is a standard method. In Section 6 we see that this approach works better than the previous discretization methods without autocutting. In discretization a χ^2-distributed test statistic has been used in the ChiMerge algorithm to decide whether to merge adjacent intervals together [20]. We give a test statistic which, when $\mathcal{H}_0$ is approximately true, tells us how likely it is that the best split produces $\mathcal{H}_1$. If we find the situation unlikely, then we can reject $\mathcal{H}_0$ and execute the split. We call this method ENT-TEST.

The test statistic is derived as follows. Denote the labels on the current interval with a vector $\boldsymbol{y}$. Let $\mathbf{P}(\boldsymbol{y} \mid \mathcal{H}_0(\boldsymbol{y}))$ be the probability of generating $\boldsymbol{y}$ according to hypothesis $\mathcal{H}_0$ when the parameters are estimated from $\boldsymbol{y}$ itself. Similarly let $\mathbf{P}(\boldsymbol{y} \mid \mathcal{H}_1(\boldsymbol{y}))$ be the probability according to $\mathcal{H}_1$. Now we need to know the probability of obtaining a pair $\langle \mathbf{P}(\boldsymbol{y}' \mid \mathcal{H}_0(\boldsymbol{y}')), \mathbf{P}(\boldsymbol{y}' \mid \mathcal{H}_1(\boldsymbol{y}'))\rangle$, where $\boldsymbol{y}' \sim \mathcal{H}_0$, that is less likely than the actual pair $\langle \mathbf{P}(\boldsymbol{y} \mid \mathcal{H}_0(\boldsymbol{y})), \mathbf{P}(\boldsymbol{y} \mid \mathcal{H}_1(\boldsymbol{y}))\rangle$. We have two problems:

1. How to generate $\boldsymbol{y}' \sim \mathcal{H}_0$ given that we do not know the *exact* probabilities of the labels under the null hypothesis $\mathcal{H}_0$?
2. How to define “less likely”?

We answer these questions by approximating that $\mathcal{H}_0$ is a permutation on the class labels that we have seen. Because $\mathcal{H}_0$ is a very simple hypothesis this estimation from the empirical data is likely to be close enough to the “truth” for our purposes and additionally $\mathbf{P}(\boldsymbol{y}' \mid \mathcal{H}_0(\boldsymbol{y}'))$ becomes a constant. Then we only need to compute $\sum_{\boldsymbol{y}' \in Y'} \mathbf{P}(\boldsymbol{y}')$, where Y' is the set of $\boldsymbol{y}'$s such that $\mathbf{P}(\boldsymbol{y}' \mid \mathcal{H}_1(\boldsymbol{y}')) < \mathbf{P}(\boldsymbol{y} \mid \mathcal{H}_1(\boldsymbol{y}))$ and $\mathbf{P}(\boldsymbol{y}')$ is the probability of $\boldsymbol{y}'$ according to $\mathcal{H}_0$. Unfortunately we do not know how to solve this problem efficiently. We resort to sampling from $\mathcal{H}_0$, i.e., generating vectors of data $\boldsymbol{y}'$ from $\mathcal{H}_0$. This is expensive, because we need to generate many vectors if we want to remove the effect of randomness from sampling.

Of course, this method gives a likelihood value and in empirical experiments we need to decide how small the likelihood value can be before splitting. In experiments we chose to split if the likelihood was below 10%. The number of samples drawn from $\mathcal{H}_0$ was fifty. The results of these experiments are given in Table 1 and Table 2. We can see that the significance value of 10% gives a good performance with respect to ENT-MDL. It is worth noticing is that ENT-TEST does not depend on autocutting to improve the performance. However, unless we can do the significance test efficiently, this method is limited to cases in which enough computational power is available to handle the sampling. It is an interesting open question whether a more efficient method to calculate the likelihood exists.

Why do we use such a complicated distribution? We could assume the number of a particular label in a partitioned interval to be normally distributed. Its parameters could, then, be taken from the unpartitioned interval. We can use the normal distribution, because it approximates quite well the multinomial one, which is the real distribution for the number of labels when the number of trials is fixed. Then these normally distributed values for both subintervals could be joined to form a variable that is χ^2-distributed; i.e., it is a sum of normalized normally distributed values squared. However, there is a flaw in this approach. The problem is that this works for a split that is in a fixed location on the interval, but in our case the hypothesis $\mathcal{H}_1$ selects the one that is the best according to its criteria. Hence, the numbers of the different labels do not conform to our assumption on their distribution.

An alternative approach to a test statistic is to simply use a test set. Unfortunately, in empirical tests this approach did not perform well. As the small number of samples in small intervals is probably to blame, the k-set validation could be more useful. However, we have not experimented with this approach yet.

6 Empirical Evaluation

We evaluate EWB, ENT-MDL, ENT-BAY, and ENT-TEST on 16 domains from the UCI machine learning repository. Also versions of ENT-MDL and ENT-BAY that carry out autocutting are included in this comparison. For each domain we randomly split the data to a training set and test set, with two-thirds being in the training set and the rest in the test set. We iterate the procedure thirty times for each domain. For an interval I the probability $\mathbf{P}(I \mid y)$ was estimated using Laplacian correction; i.e., each interval has one additional training example with label y.

The average prediction accuracies are given in Table 1 and statistical significance tests using t-test[1] with confidence level 0.95 are in Table 2. From these results we see that autocutting benefits both ENT-MDL and ENT-BAY. The resulting increase in average accuracy over all 16 test domains is 0.8 percentage units for ENT-MDL and 0.6 percentage units for ENT-BAY. Unsupervised EWB is the clear loser in these experiments, but still it is able to win in some domains.

[1] The assumptions behind the t-test are violated and as Dietterich [21] argues this can result in inaccurate significance measurements. However, we also used the Wilcoxon signed-rank test, which has fewer assumptions, and the results were identical.

Table 1. Performance of discretization algorithms on Naïve Bayes. The average classification accuracy over 30 repetitions of randomized training set selection for 16 UCI domains is shown. Also the average over all 16 domains is given.

	EWB	MDL	MDL-A	BAY95	BAY95-A	TEST-10%
Iris	94.5	94.0	93.5	93.9	94.0	94.7
Glass	60.7	63.7	67.3	68.8	69.4	69.0
Bupa	61.6	57.1	63.3	57.4	62.2	60.0
Pima	75.0	74.7	74.1	75.5	74.5	74.2
Ecoli	83.5	84.9	85.0	85.9	85.5	84.9
Segmentation	79.0	81.3	84.0	81.6	83.2	82.3
Wine	97.1	98.3	98.3	98.3	98.3	98.1
Australian	85.2	85.3	85.0	85.2	85.7	85.5
German	71.8	71.9	73.2	71.4	73.9	74.7
Iono	85.9	89.8	89.2	90.3	88.3	89.8
Sonar	74.4	75.1	75.4	74.4	77.8	75.6
Wisconsin	97.4	97.6	97.4	97.6	97.5	97.4
Letter	61.2	73.6	73.6	73.5	73.6	73.5
Abalone	58.0	58.7	58.3	58.4	58.2	58.4
Vehicle	60.1	58.4	59.2	61.7	61.4	62.0
Page	92.3	93.4	93.4	93.4	93.5	93.2
Average	77.4	78.6	79.4	79.2	79.8	79.6

In these ones the numerical values of attributes are probably important. The inefficient ENT-TEST is better than pure ENT-MDL or ENT-BAY, and performs approximately the same when autocutting is factored in. It also has the least number of statistically significant losses against the other algorithms. The two entropy-based approaches ENT-MDL and ENT-BAY have quite similar overall performance. However, ENT-BAY is slightly better than ENT-MDL and wins more often against EWB.

7 Conclusions

In this paper we discussed the flaws in the theoretical justification of Fayyad and Irani's [6] entropy-based recursive discretization algorithm. The MDL criterion used to stop the recursive partitioning is not based on real codes. We proposed to replace the MDL criterion with an extremely simple Bayesian model selection criterion. In empirical evaluation the Bayesian approach has similar, though, slightly better overall performance than the MDL approach. Of course, the success of discretization algorithms varies from domain to domain. The Bayesian approach has the advantage of being simpler than the MDL approach and, furthermore, can be easily customized by the user.

We also put forward a test statistic to decide on partitioning. This approach does not need heuristic techniques to improve its performance like the other entropy-based techniques do. Empirical evaluation shows this approach to have

Table 2. Number of statistically significant wins using the *t*-test with 0.95 confidence level. The figure in a cell denotes the number of wins (out of 16) that the discretization algorithm mentioned on the row obtains with respect to the one on the column.

	EWB	MDL	MDL-A	BAY95	BAY95-A	TEST-10%
EWB	●	2	2	1	0	0
MDL	5	●	0	0	1	0
MDL-A	8	4	●	3	0	1
BAY95	8	2	2	●	1	1
BAY95-A	9	5	2	3	●	0
TEST-10%	9	4	2	2	1	●

a comparative performance with the heuristic approaches, but unfortunately it is expensive to compute.

In this work we have demonstrated that better working new efficient heuristic approaches to discretization and (inefficient) well-founded approaches can be developed. In the long run would be interesting to find solutions to the discretization problem that are at the same time *efficient* and *theoretically justified*.

Acknowledgments

This work has been financially supported by Tampere Graduate School in Information Science and Engineering (TISE), Academy of Finland, and Nokia Foundation.

References

1. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In Prieditis, A., Russell, S., eds.: Proc. 12th International Conference on Machine Learning, San Francisco, CA, Morgan Kaufmann (1995) 194–202
2. Hsu, C.N., Huang, H.J., Wong, T.T.: Implications of the Dirichlet assumption for discretization of continuous variables in naive Bayesian classifiers. Machine Learning **53** (2003) 235–263
3. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: An enabling technique. Data Mining and Knowledge Discovery **6** (2002) 393–423
4. Yang, Y., Webb, G.I.: A comparative study of discretization methods for naive-Bayes classifiers. In: Proc. Pacific Rim Knowledge Acquisition Workshop (PKAW). (2002) 159–173
5. Elomaa, T., Rousu, J.: Efficient multisplitting revisited: Optima-preserving elimination of partition candidates. Data Mining and Knowledge Discovery **8** (2004) 97–126
6. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proc. 13th International Joint Conference on Artificial Intelligence, San Francisco, CA, Morgan Kaufmann (1993) 1022–1027

7. Wong, A., Chiu, D.: Synthesizing statistical knowledge from incomplete mixed-mode data. IEEE Transactions on Pattern Analysis **9** (1987) 796–805
8. Catlett, J.: On changing continuous attributes into ordered discrete attributes. In Kodratoff, Y., ed.: Machine Learning — EWSL-91, Proc. 5th European Working Session on Learning. Volume 482 of Lecture Notes in Computer Science., Berlin, Heidelberg, Springer-Verlag (1991) 164–178
9. Hand, D.J., Yu, K.: Idiot Bayes? not so stupid after all. International Statistical Review **69** (2001) 385–398
10. Rish, I.: An empirical study of the naive Bayes classifier. In: IJCAI-01 workshop on "Empirical Methods in AI". (2001)
11. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning **29** (1997) 103–130
12. Chlebus, B.S., Nguyen, S.H.: On finding optimal discretizations for two attributes. In Polkowski, L., Skowron, A., eds.: Rough Sets and Current Trends in Computing, Proc. First International Conference. Volume 1424 of LNAI., Heidelberg, Springer (1998) 537–544
13. Elomaa, T., Rousu, J.: On decision boundaries of naïve Bayes in continuous domains. In N. Lavrač et al., ed.: Knowledge Discovery in Databases: PKDD 2003, Proc. 7th European Conference. Volume 2838 of LNAI., Berlin, Heidelberg, Springer-Verlag (2003) 144–155
14. Călinescu, G., Dumitrescu, A., Karloff, H., Wan, P.J.: Separating points by axis-parallel lines. International Journal of Computational Geometry & Applications **15** (2005) 575–590
15. Elomaa, T., Kujala, J., Rousu, J.: Approximation algorithms for minimizing empirical error by axis-parallel hyperplanes. In J. Gama et al., ed.: Machine Learning: ECML 2005, Proc. 16th European Conference. Volume 3720 of LNAI., Berlin, Heidelberg, Springer-Verlag (2005) 547–555
16. John, G., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: Proc. 11th Annual Conference on Uncertainty in Artificial Intelligence, San Francisco, CA, Morgan Kaufmann (1995) 338–345
17. Fayyad, U.M., Irani, K.B.: On the handling of continuous-valued attributes in decision tree generation. Machine Learning **8** (1992) 87–102
18. Kohavi, R., Sahami, M.: Error-based and entropy-based discretization of continuous features. In Simoudis, E., Han, J.W., Fayyad, U., eds.: Proc. 2nd International Conference on Knowledge Discovery and Data Mining, Menlo Park, CA, AAAI Press (1996) 114–119
19. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, New York, NY (1991)
20. Kerber, R.: Chimerge: Discretization of numeric attributes. In: Proc. 10th National Conference on Artificial Intelligence, Cambridge, MA, MIT Press (1992) 123–128
21. Dietterich, T.G.: Approximate statistical test for comparing supervised classification learning algorithms. Neural Computation **10** (1998) 1895–1923