Making Better Use of Global Discretization

Eibe Frank and Ian H. Witten
Department of Computer Science
University of Waikato
Hamilton, New Zealand
{eibe, ihw}@cs.waikato.ac.nz

Abstract

Before applying learning algorithms to datasets, practitioners often globally discretize any numeric attributes. If the algorithm cannot handle numeric attributes directly, prior discretization is essential. Even if it can, prior discretization often accelerates induction, and may produce simpler and more accurate classifiers.

As it is generally done, global discretization denies the learning algorithm any chance of taking advantage of the ordering information implicit in numeric attributes. However, a simple transformation of discretized data preserves this information in a form that learners can use. We show that, compared to using the discretized data directly, this transformation significantly increases the accuracy of decision trees built by C4.5, decision lists built by PART, and decision tables built using the wrapper method, on several benchmark datasets. Moreover, it can significantly reduce the size of the resulting classifiers.

This simple technique makes global discretization an even more useful tool for data preprocessing.

1 Introduction

Algorithms that transform numeric attributes into discrete ones are useful for several reasons. Most importantly, they enable learning schemes that can only handle nominal attributes to process numeric data. But this is not the only situation where they are useful. Often, it is worthwhile to discretize even if the learning scheme is able to process numeric data directly—as can most schemes that learn decision trees or lists. There are two reasons for this. First, discretization accelerates learning because nominal attributes are generally processed faster than numeric ones (Catlett, 1991)—assuming, of course, that the discretization itself is accomplished quickly. Second, it reduces the likelihood of overfitting by narrowing the space of possible hypotheses that the learning scheme can investigate, thereby lowering the chance of finding a complex hypothesis that fits the training data particularly well just by chance. The resulting classifiers are often significantly less complex and sometimes more accurate than classifiers learned from the raw numeric data.

Catlett (1991) presents a supervised method for global discretization and discusses the speed-up that can be achieved by applying it before building a decision tree.[1] His method is improved by Fayyad and Irani (1993), who show how to prevent the discretization from becoming too fine-grained by using the minimum description length principle to determine the appropriate granularity. Dougherty, Kohavi and Sahami (1995) compare several supervised and unsupervised methods and conclude that Fayyad and Irani's method produces the most accurate classifiers. Kohavi and Sahami (1996) extend the comparison to include three new methods, and corroborate this result.[2]

The trouble with global discretization, as it is normally used, is that the learning algorithm cannot take advantage of the ordering information implicit in numeric attributes because it treats the different discretized val-

[1] Note, however, that he sorts the data for each attribute at each node of the decision tree. This can be avoided by careful book-keeping: it is only actually necessary to sort the data once for each attribute.

[2] Neither of these two comparisons include ChiMerge, a theoretically well-founded discretization method based on the chi-squared test (Kerber, 1992).

ues as though they were completely independent. For example, if the decision-tree inducer C4.5 (Quinlan, 1992) decides to split on a pre-discretized attribute, it generates a multiway branch. In contrast, internal discretization implicitly takes advantage of ordering information by using successive binary splits instead.

However, there is a very simple transformation of discretized data that preserves the ordering information in a form that learners can use. This paper investigates the effect of this transformation on common classification models. Using a set of benchmark datasets from the UCI repository, we show that it often leads to more accurate decision trees, decision lists, and decision tables. In addition, it significantly reduces the size of these classifiers in several domains.

Section 2 describes the transformation and gives examples of its application. Section 3 applies decision-tree, decision-list, and decision-table inducers to the datasets with numeric attributes globally discretized, both with and without the transformation, and compares the results. The decision-tree and decision-list inducers are also applied to the raw datasets, using internal discretization. Section 4 discusses related work, and Section 5 gives some concluding remarks.

2 Using Ordering Information in Discretized Attributes

Global discretization transforms a numeric attribute A into an attribute A^* with values $\{V_1, V_2, \ldots, V_n\}$, where each value V_i of the new attribute represents a range of numeric values of the original attribute. Virtually all supervised learning schemes treat discretized attributes in the same way as nominal ones. This means that potentially useful information is discarded—in other words, the learning algorithm is being deprived of valuable domain knowledge—for the fact is that discretized attribute values should be treated as ordered entities. In most learning schemes this loss is completely unnecessary: the ordering information can be exploited in decision trees, lists and tables without any change to the learning algorithm itself.

2.1 Decision Trees

Consider a decision tree learner that constructs trees with univariate tests. During learning, it has to select tests for each node of the tree structure, given the data present at that node. Exploiting the ordering information implicit in a discretized attribute A^* simply

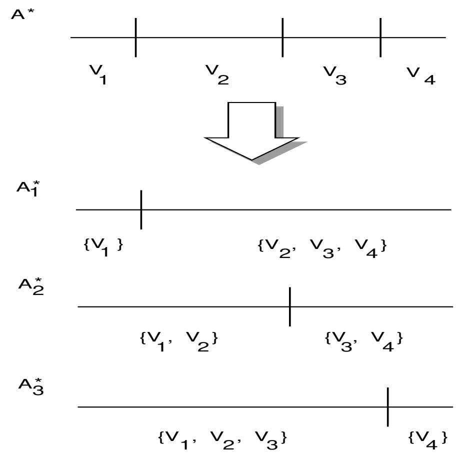

Figure 1: Transformation of a discretized attribute with four values into three binary attributes

amounts to investigating tests of the form $A^* \leq V_i$ instead of $A^* = V_i$. Although this seems to imply that the inner workings of the learning scheme have to be changed to enable it to deal with ordered attributes of this type, the problem can be avoided by transforming the attributes before applying the learning scheme.

For each discretized attribute with n values, $n-1$ boolean ones are introduced, one for each of the attribute's first $n-1$ values, and the original discretized attribute is discarded. The ith boolean attribute represents the test $A^* \leq V_i$ (Figure 1). Figure 2b shows the effect of applying Fayyad and Irani's (1993) discretization method in conjunction with this transformation on a pruned decision tree produced by C4.5 (Quinlan, 1992) for the ionosphere dataset. The tree produced by C4.5 on the raw dataset, using its standard method of internal discretization, is shown in Figure 2a for comparison. As we will see in Section 3, the tree in Figure 2b is significantly more accurate than that of Figure 2a; it is also significantly smaller and more accurate than the tree built from the discretized data without applying the transformation.

Another way of interpreting this strategy is that it provides a way of combining *global* and *local* information during learning. The discretization method preselects candidate cutpoints based on global information, while the learner decides which of these cut-points is most appropriate in the current local context.

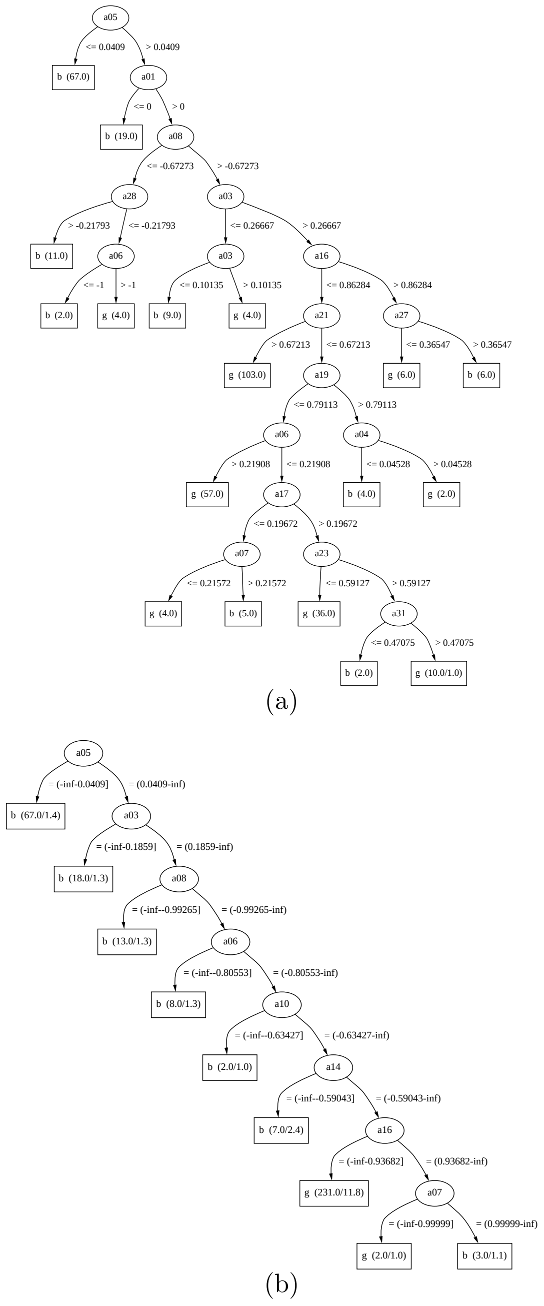

Figure 2: Decision tree for (a) raw, and (b) discretized and transformed ionosphere dataset

2.2 Decision Lists

Decision lists (Rivest, 1987) can also take advantage of ordering information. A decision list is an ordered set of rules. During classification, a test instance is assigned to the class of the first rule whose premise it satisfies. The premise of each rule consists of a conjunction of attribute-value tests. For nominal attributes, these tests have the form $A = V_i$. The ordering information implicit in discretized attributes can easily be exploited by applying tests of the form $A^* \leq V_i$. Again, the learning algorithm does not have to be changed to allow this type of test to be included: the attribute transformation of Section 2.1 can be used instead.

2.3 Decision Tables

Decision tables, like decision lists, are sets of rules. A decision table enumerates all possible combinations of attribute-value tests for a selected set of attributes, together with a class assignment for each combination. The size of the table increases exponentially with the number of attributes included: more specifically, if n_j is the number of values for attribute j, the table contains $\prod_j n_j$ rows. This means that the instance space becomes very fragmented when many attributes, with many possible values, are included in the table. The learning problem is to find a small subset of attributes, each with a small number of possible values, while maintaining high accuracy on the training data.

Decision tables can be constructed for numeric data if it is discretized *a priori*.[3] However, a discretized attribute may be only partially informative when used in conjunction with other attributes. In that case, if the discretization is too fine-grained, the instance space can become overly fragmented, resulting in suboptimal performance. The problem is easily solved by applying the transformation discussed in Section 2.1. If an attribute is partially informative, only those binary attributes derived from the original discretized attribute that are relevant in the current context will be included in the decision table, thereby minimizing fragmentation.

3 Experimental Evaluation

In this section, we will see that significantly more accurate decision trees, lists, and tables can be produced by exploiting the ordering information implicit

[3] Kohavi's (1995) method does handle numeric attributes, but only in a very limited way.

Table 1: Datasets used for the experiments

Dataset	Size	Numeric	Nominal	Classes
anneal	898	6	32	5
australian	690	6	9	2
autos	205	15	10	6
balance-scale	625	4	0	3
breast-w	699	9	0	2
german	1000	7	13	2
glass (G2)	163	9	0	2
glass	214	9	0	6
heart-c	303	6	7	2
heart-h	294	6	7	2
heart-statlog	270	13	0	2
hepatitis	155	6	13	2
horse-colic	368	7	15	2
hypothyroid	3772	7	22	4
ionosphere	351	34	0	2
iris	150	4	0	3
labor	57	8	8	2
lymph	148	3	15	4
pima-indians	768	8	0	2
segment	2310	19	0	7
sick	3772	7	22	2
sonar	208	60	0	2
vehicle	846	18	0	4
vowel	990	10	3	11
waveform	5000	40	0	3
zoo	101	1	15	7

in discretized numeric attributes. All results are based on the 26 UCI benchmark datasets listed in Table 1, each of which contains at least one numeric attribute. Fayyad and Irani's (Fayyad & Irani, 1993) discretization method was employed throughout.

3.1 Decision Trees

We begin with decision trees, and compare pruned trees generated by C4.5[4] in three different ways. First, trees are built from the discretized data transformed using the procedure described in Section 2.1 (ORD). Second, they are built directly from the discretized data (DISC). Third, they are built by C4.5 from the raw data (RAW).

The results are listed in Table 2. This shows the percentage of correct classifications, averaged over ten ten-fold cross-validation runs, along with the standard deviation of the ten. Also shown is the average tree size measured by the number of nodes in it. The same folds were used for each scheme. In all experiments with discretization, the training data was discretized separately for each fold—otherwise the cross-validation estimates would be optimistically biased (Kohavi & Sahami, 1996).

Results for DISC and RAW are marked with ∘ if they show significant improvement over the corresponding results for ORD, and with • if they show significant degradation. Throughout, we speak of results being "significantly different" if the difference is statistically significant at the 1% level according to a paired two-sided t-test, each pair of data points consisting of the estimates obtained in one ten-fold cross-validation run for the two learning schemes being compared.

Table 3 shows how the different methods compare with each other. Each entry indicates the number of datasets for which the method associated with its column is significantly more accurate than the method associated with its row.

As Table 3 shows, it is clearly advantageous to make use of the ordering information in discretized attributes. ORD is significantly more accurate than DISC on six datasets (second row, first column), whereas the inverse is never true (first row, second column). It is interesting to see that ORD and RAW are almost neck and neck in terms of accuracy: ORD is more accurate on four datasets, RAW on five. It is also apparent that RAW builds more accurate trees than DISC in eight cases, whereas the reverse is true in only two. This is consistent with the results of Quinlan (Quinlan, 1996), who found that RAW has a strong advantage over DISC in terms of accuracy. As Table 2 and Table 3 show, this advantage largely vanishes when the ordering information is exploited.

Table 3 also shows that the trees built by ORD are generally smaller than those built by DISC: the former builds significantly smaller trees for twelve datasets and significantly larger ones for only three. The advantage of using ORD is even more pronounced when compared to RAW. On sixteen datasets ORD generates smaller trees, and larger ones on only four.

3.2 Decision Lists

As explained in Section 2.2, decision lists can exploit ordering information in the same way as decision trees. The empirical results presented here were obtained using the rule learner PART, which has been shown to perform comparably to other state-of-the-art rule learning methods (Frank & Witten, 1998).

Tables 4 and 5 summarize results obtained using the

[4] In all our experiments we used C4.5 Revision 8 (Quinlan, 1996).

Table 2: C4.5: Percentage of correct classifications and size of trees, with standard deviations, for ordered (ORD), discretized (DISC), and raw data (RAW)

Dataset	Accuracy ORD	Accuracy DISC	Accuracy RAW	Size ORD	Size DISC	Size RAW
anneal	98.8±0.2	98.8±0.1	98.7±0.3	46.9±0.7	51.3±1.1 •	48.0±1.4
australian	86.5±0.3	86.0±0.5	85.5±0.7 •	21.1±1.7	22.6±2.2	32.5±3.2 •
autos	75.9±2.4	74.5±1.8	80.0±2.5 ○	76.2±2.9	101.5±4.6 •	62.4±2.2 ○
balance-scale	75.7±0.9	75.5±0.9	77.6±0.9 ○	33.6±2.3	41.3±2.1 •	82.2±2.9 •
breast-w	95.4±0.4	95.0±0.5	94.9±0.4	18.1±1.3	20.5±1.2 •	24.6±1.3 •
german	72.2±0.8	71.8±0.7	71.1±1.1	90.6±4.9	89.1±7.2	124.4±6.0 •
glass (G2)	77.4±2.5	77.4±2.4	78.1±1.8	14.1±0.7	11.8±0.8 ○	23.7±1.6 •
glass	70.2±2.0	72.0±1.2	68.2±2.4	28.5±1.1	37.1±1.1 •	45.5±1.3 •
heart-c	77.3±1.8	77.8±1.3	76.7±1.7	33.2±2.1	31.4±2.0 ○	43.5±2.5 •
heart-h	79.3±1.0	79.3±1.0	79.8±0.8	9.0±1.8	9.1±1.9	10.8±0.9
heart-statlog	81.4±1.3	81.5±1.4	78.3±1.9 •	24.1±2.1	23.9±2.0	34.9±2.6 •
hepatitis	78.5±1.3	79.3±2.3	79.7±1.2	10.7±1.8	10.7±1.7	17.8±1.2 •
horse-colic	85.3±0.3	85.3±0.3	85.4±0.3	8.5±0.7	8.5±0.7	8.7±0.7
hypothyroid	99.5±0.0	99.2±0.1 •	99.5±0.0	23.2±0.4	46.6±1.3 •	27.9±0.3 •
ionosphere	93.0±0.4	89.8±1.2 •	89.4±1.3 •	17.4±0.5	25.1±2.1 •	27.1±0.9 •
iris	93.7±1.1	92.9±0.9 •	94.4±0.6	6.1±0.3	6.5±0.5	8.3±0.5 •
labor	78.0±3.4	77.5±3.3	77.2±4.1	6.5±0.5	6.4±0.5	7.0±0.8
lymph	74.9±2.0	75.8±2.0	75.8±2.9	25.5±1.3	25.6±1.3	27.7±1.2 •
pima-indians	73.9±0.8	73.8±1.0	74.5±1.4	25.3±2.1	23.6±2.1	42.3±4.1 •
segment	96.0±0.3	94.1±0.3 •	96.7±0.3 ○	88.0±1.7	336.7±9.4 •	82.0±2.6 ○
sick	97.8±0.1	97.8±0.1	98.7±0.2 ○	29.1±0.7	32.2±0.9 •	48.7±2.1 •
sonar	76.0±2.2	76.0±2.2	75.0±3.0	28.5±1.6	28.5±1.6	28.0±0.8
vehicle	70.3±1.0	69.7±1.2	72.8±1.1 ○	122.2±2.0	200.7±5.4 •	139.4±5.0 •
vowel	78.5±0.9	76.2±0.7 •	79.6±1.3	298.4±14.4	370.4±10.4 •	216.2±7.1 ○
waveform	77.0±0.6	74.7±0.4 •	75.3±0.7 •	651.2±12.2	628.2±15.7 ○	590.1±9.9 ○
zoo	91.1±1.2	90.8±1.5	91.1±1.2	15.1±0.3	16.6±0.5 •	15.2±0.4

Table 3: Results of paired t-tests (p=0.01) for C4.5: number indicates how often method in column significantly outperforms method in row

	Accuracy ORD	Accuracy DISC	Accuracy RAW	Size ORD	Size DISC	Size RAW
ORD	–	**0**	**5**	–	**3**	**4**
DISC	**6**	–	8	**12**	–	8
RAW	**4**	2	–	**16**	13	–

same experimental procedure as in Section 3.1. ORD denotes results obtained by running PART on the discretized and transformed data—thereby incorporating ordering information in the learning process—and DISC stands for PART run on the discretized data directly. Results for PART using the raw numeric data are included as RAW.

Table 5 shows that, just as with decision trees, decision lists built from the discretized and transformed data are preferable to those generated from the discretized data directly. ORD is significantly more accurate than DISC on five datasets, whereas DISC is never significantly better than ORD. ORD is also generally the better choice when the number of rules is a critical factor. It produces significantly fewer rules than DISC on ten datasets, and significantly more on only two.

The situation is less clear-cut when ORD is compared to RAW. In four cases, PART builds significantly more accurate classifiers from the raw data than from the discretized and transformed data. In only one case is the decision list significantly more accurate for the latter type of data. Size is also not necessarily an argument for applying pre-processing. In several cases—for example waveform, vowel, vehicle, and segment—RAW generates significantly fewer rules than ORD.

Table 4: PART: Percentage of correct classifications and size, with standard deviations, for ordered (ORD), discretized (DISC), and raw data (RAW)

Dataset	Accuracy ORD	Accuracy DISC	Accuracy RAW	Size ORD	Size DISC	Size RAW
anneal	98.6±0.2	98.7±0.4	98.4±0.3	15.4±0.5	16.8±0.4 •	14.6±0.5
australian	84.8±0.5	84.6±0.6	84.3±1.2	21.6±1.3	21.3±1.2	30.5±1.5 •
autos	73.5±3.0	71.8±2.3	74.5±1.1	20.2±0.4	23.6±0.8 •	20.5±1.0
balance-scale	77.0±0.8	77.5±0.7	82.3±1.2 ◦	13.4±0.7	14.4±1.0	38.7±1.2 •
breast-w	95.4±0.5	95.3±0.7	94.9±0.4	11.2±0.6	9.7±0.7 ◦	10.0±0.7 ◦
german	71.0±1.3	71.3±0.9	70.0±1.4	57.6±1.7	58.7±1.8	69.6±1.3 •
glass (G2)	77.7±2.5	79.5±2.2	80.0±4.0	5.0±0.0	5.6±0.5 •	6.8±0.4 •
glass	70.6±2.1	70.9±1.8	69.8±2.3	11.5±0.5	12.6±0.5 •	15.2±0.8 •
heart-c	79.6±1.5	80.5±1.7	78.1±1.6	15.9±0.7	15.8±0.8	19.3±0.8 •
heart-h	79.8±1.5	79.8±1.5	80.4±1.6	7.7±0.5	7.8±0.6	8.9±0.9 •
heart-statlog	82.1±0.9	82.1±0.9	78.9±1.3 •	13.9±0.9	14.0±0.7	17.7±1.1 •
hepatitis	80.0±2.8	79.8±3.6	80.2±1.9	7.4±0.8	7.3±0.8	8.3±0.7
horse-colic	83.9±1.0	83.9±1.0	84.4±0.8	7.4±0.7	7.4±0.7	9.7±0.8 •
hypothyroid	99.4±0.1	99.4±0.1	99.5±0.1	8.2±0.4	14.2±0.4 •	10.4±0.5 •
ionosphere	91.4±1.3	89.9±0.7 •	90.6±1.3	7.7±0.5	10.0±0.5 •	7.6±0.5
iris	93.7±1.0	92.7±0.8 •	93.7±1.6	3.6±0.5	4.5±0.5 •	4.0±0.5
labor	76.0±4.0	75.8±4.1	77.3±3.9	3.7±0.5	3.7±0.5	3.6±0.5
lymph	76.9±2.1	77.7±3.1	76.5±2.7	10.8±0.6	10.8±0.6	11.7±0.7
pima-indians	73.5±0.8	73.4±1.3	73.6±0.5	22.2±2.7	22.4±1.7	7.3±0.5 ◦
segment	96.1±0.3	93.6±0.4 •	96.6±0.3 ◦	31.0±0.8	59.3±2.5 •	27.5±0.7 ◦
sick	97.8±0.1	97.9±0.0	98.6±0.1 ◦	15.1±0.7	14.9±0.9	19.1±0.9 •
sonar	77.1±3.7	77.2±3.5	76.5±2.3	12.5±0.5	12.5±0.5	7.5±0.7 ◦
vehicle	71.1±0.9	68.2±1.7 •	72.4±0.8 ◦	60.7±1.6	65.3±1.8 •	33.2±1.2 ◦
vowel	77.7±1.0	75.3±0.9 •	78.1±1.1	93.5±2.1	116.4±0.8 •	66.0±1.4 ◦
waveform	77.3±0.5	76.6±0.6	78.0±0.5	283.0±4.3	263.9±4.4 ◦	87.9±2.4 ◦
zoo	92.2±1.2	92.0±1.1	92.2±1.2	8.0±0.0	8.0±0.0	8.0±0.0

Table 5: Results of paired t-tests (p=0.01) for PART: number indicates how often method in column significantly outperforms method in row

	Accuracy ORD	Accuracy DISC	Accuracy RAW	Size ORD	Size DISC	Size RAW
ORD	–	**0**	**4**	–	**2**	**7**
DISC	**5**	–	8	**10**	–	10
RAW	**1**	2	–	**11**	9	–

3.3 Decision Tables

For our experiments on decision tables, we employed a learning algorithm that uses the wrapper method in conjunction with a best first search for selecting the attributes in the table (Kohavi, 1995). Our implementation of the wrapper employs leave-one-out cross-validation and a stopping criterion of five consecutive fully expanded non-improving nodes in the search space.

Table 6 and Table 7 show the results. The size of a decision table is measured by the number of entries in it. Only combinations of attribute-value tests that cover at least one instance of the training data are counted as entries.

Table 7 shows that ORD produces more accurate tables than DISC on seven datasets. Unlike decision trees and lists, however, for decision tables there is a dataset (glass) for which ORD is significantly less accurate—although the relative difference in accuracy is rather small. Table 7 also shows that ORD generates significantly smaller tables than DISC for eight datasets, and significantly larger ones for only two. Moreover, on one of these two datasets (waveform) it is also significantly more accurate, indicating that there are some discretized attributes in this dataset that are only partially informative, and therefore omitted from the table built by DISC. As discussed in Section 2.3, ORD successfully reduces the fragmentation of the instance space, sometimes with a positive effect on accuracy

Table 6: Decision tables: Percentage of correct classifications and size, with standard deviations, for ordered (ORD) and discretized (DISC) data

Dataset	Accuracy ORD	Accuracy DISC	Size ORD	Size DISC
anneal	99.3±0.2	98.6±0.1 •	46.8±2.1	119.8±4.6 •
australian	85.1±0.9	85.2±0.8	43.7±7.8	45.9±7.5
autos	77.4±1.1	77.5±1.8	77.6±1.7	77.6±3.9
balance-scale	78.1±0.5	74.5±0.9 •	41.6±5.5	42.7±5.6
breast-w	95.3±0.5	94.7±0.7	32.7±3.3	46.1±6.4 •
german	71.6±0.9	71.6±0.8	122.7±21.6	120.4±21.4
glass (G2)	78.2±2.0	77.9±1.8	11.7±1.0	11.9±0.7
glass	68.1±1.2	69.6±0.7 ○	23.5±3.4	36.8±2.1 •
heart-c	78.5±2.0	78.6±1.7	37.3±4.2	42.8±6.5 •
heart-h	79.6±1.0	79.6±1.1	14.6±1.6	15.0±1.8
heart-statlog	82.6±1.4	82.5±1.3	21.5±4.7	21.6±4.8
hepatitis	80.4±2.9	80.2±2.3	39.1±5.2	36.8±4.7
horse-colic	82.7±0.8	82.7±0.8	62.0±8.1	62.0±8.1
hypothyroid	99.6±0.1	99.4±0.0 •	72.7±2.7	69.2±5.5
ionosphere	89.5±1.3	89.6±1.2	22.9±1.4	28.4±1.4 •
iris	93.2±0.8	92.7±0.5 •	4.9±0.5	5.5±0.5 •
labor	83.3±3.6	83.8±3.6	10.7±1.1	10.7±1.0
lymph	74.3±2.4	74.9±1.8	27.1±3.0	25.7±3.1
pima-indians	73.5±0.8	74.0±1.3	44.9±7.0	41.4±8.6
segment	94.3±0.4	92.1±0.5 •	143.0±11.2	302.0±11.6 •
sick	97.6±0.1	97.6±0.1	92.8±7.0	71.6±9.1 ○
sonar	73.7±2.3	73.5±2.5	35.4±2.5	35.0±2.5
vehicle	68.2±1.4	65.1±0.7 •	89.1±7.0	100.2±9.5
vowel	70.2±1.1	70.7±1.6	354.6±14.6	355.3±7.3
waveform	76.7±0.4	73.7±0.6 •	423.3±18.4	270.3±56.0 ○
zoo	91.4±1.5	90.0±1.4	12.7±0.4	15.7±0.4 •

Table 7: Results of paired t-tests (p=0.01) for Decision tables: number indicates how often method in column significantly outperforms method in row

	Accuracy ORD	Accuracy DISC	Size ORD	Size DISC
ORD	–	**1**	–	**2**
DISC	**7**	–	**8**	–

(anneal, iris, segment).

Note that the time complexity of learning decision tables with best first search is not linear in the number of attributes.[5] Hence ORD can be significantly slower than DISC, depending on the number of attributes generated by the transformation. For C4.5 and PART the time complexity is linear, and the difference between ORD and DISC is roughly proportional to the average number of values in the discretized attributes. This difference is usually negligible compared to the time needed for the discretization, which involves sorting the training instances once for each numeric attribute.

[5]However, there are fast algorithms for learning decision tables that are linear in the number of attributes (Kohavi & Sommerfield, 1998).

4 Related Work

Work on methods for global discretization has already been discussed in Section 1. However, there is another line of research, concerned with the combination of global and local information in the learning process, that is closely related to the work presented in this paper.

Pazzani (1998) introduces the notion of globally predictive tests for rule learning systems. In order to incorporate them into rules, he introduces a bias that forces the learning scheme to choose only those tests that are globally predictive. A test is defined to be globally predictive of a certain class if the probability of observing this class after the test has been performed is greater than the prior probability of the class. On some datasets this restriction significantly increases accuracy; however, on others the effect is detrimental. Therefore the afore-mentioned bias is weakened by allowing locally predictive rules, but only if they are significantly more accurate than the globally predictive alternatives. This combination significantly increases accuracy on three out of fifteen datasets, and never significantly decreases it. It is also claimed that this procedure improves the comprehensibility of the resulting rule sets, although this claim is not validated by experiments.

Vilalta *et al.* (1997) argue that C4.5rules (Quinlan, 1992) uses a kind of global data analysis to combat the fragmentation problem in C4.5's decision trees. Each of the rules derived from a decision tree is pruned individually using all the available training data—in other words, tests are deleted from a rule if this makes the rule more globally predictive.

5 Conclusions

This paper introduces a simple transformation of discretized attributes that allows learning schemes for decision trees, lists, and tables to make full use of the ordering information present in those attributes. Implemented as a pre-processing step that takes place after discretization but before applying the learning scheme, it can be used in conjunction with existing learning algorithms without any modifications being necessary.

The empirical results we have presented show that the transformation is indeed useful when used in conjunction with decision trees produced by C4.5, decision lists produced by PART, and decision tables produced using the wrapper method. Compared to using the discretized data directly, it significantly increases the accuracy of the classifiers in several cases, rarely decreasing it. (In our experiments, a significant decrease occurred only once.) Moreover, classifiers are generally significantly smaller if the transformation is applied.

In decision trees and lists, the same effect can be achieved by coding discretized attributes as integers. In this case, if the generated classifier is to be comprehensible for the user, it should be post-processed by replacing the artificial integer-valued tests with tests on values of the original numeric attribute. The same kind of integer coding can also be used to exploit ordering information in instance-based learning algorithms. Of course, if a learning scheme supports ordered attributes directly, any transformation is superfluous: the effect is achieved by declaring discretized attributes to be of type "ordered". Whichever way it is done, our experiments show that it pays to give the learning scheme the opportunity to exploit the ordering information in discretized attributes.

References

Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. In Kodratoff, Y. (Ed.), *Proceedings of the European Working Session on Learning* (pp. 164–178). Berlin: Springer-Verlag.

Dougherty, J., Kohavi, R. & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning* (pp. 194–202). Morgan Kaufmann.

Fayyad, U. M. & Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artifical Intelligence* (pp. 1022–1027). Morgan Kaufmann.

Frank, E. & Witten, I. H. (1998). Generating accurate rule sets without global optimization. In *Proceedings of the 15th International Conference on Machine Learning* (pp. 144–151). San Francisco, CA: Morgan Kaufmann.

Kerber, R. (1992). Discretization of numeric attributes. In *Proceedings of the 10th National Conference on Artificial Intelligence* (pp. 123–128). Menlo Park, CA: AAAI Press/MIT Press.

Kohavi, R. (1995). The power of decision tables. In Lavrac, N. & Wrobel, S. (Eds.), *Machine Learning: ECML-95* (pp. 174–189). Berlin: Springer-Verlag.

Kohavi, R. & Sahami, M. (1996). Error-based and entropy-based discretization of continuous features. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (pp. 114–119). Menlo Park: AAAI Press.

Kohavi, R. & Sommerfield, D. (1998). Targeting business users with decision table classifiers. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining.* AAAI Press.

Pazzani, M. J. (1998). Learning with globally predictive tests. In *Proceedings of the 1st International Conference on Discovery Science*. Springer-Verlag.

Quinlan, J. (1992). *C4.5: Programs for Machine Learning*. Los Altos, CA: Morgan Kaufmann.

Quinlan, J. (1996). Improved use of continuous attribute in C4.5. *Journal of Artificial Intelligence Research*, *4*, 77–90.

Rivest, R. L. (1987). Learning decision lists. *Machine Learning*, *2*, 229–246.

Vilalta, R., Blix, G. & Rendell, L. (1997). Global data analysis and the fragmentation problem in decision tree induction. In *Proceedings of the 9th European Conference on Machine Learning* (pp. 312–327). Heidelberg: Springer-Verlag.