In Armand Prieditis & Stuart Russell, eds., *Machine Learning: Proceedings of the Twelfth International Conference*, 1995, Morgan Kaufmann Publishers, San Francisco, CA.

Supervised and Unsupervised Discretization of Continuous Features

James Dougherty **Ron Kohavi** **Mehran Sahami**
Computer Science Department
Stanford University
Stanford, CA. 94305
{jfd,ronnyk,sahami}@CS.Stanford.EDU

Abstract

Many supervised machine learning algorithms require a discrete feature space. In this paper, we review previous work on continuous feature discretization, identify defining characteristics of the methods, and conduct an empirical evaluation of several methods. We compare binning, an unsupervised discretization method, to entropy-based and purity-based methods, which are supervised algorithms. We found that the performance of the Naive-Bayes algorithm significantly improved when features were discretized using an entropy-based method. In fact, over the 16 tested datasets, the discretized version of Naive-Bayes slightly outperformed C4.5 on average. We also show that in some cases, the performance of the C4.5 induction algorithm significantly improved if features were discretized in advance; in our experiments, the performance never significantly degraded, an interesting phenomenon considering the fact that C4.5 is capable of locally discretizing features.

1 Introduction

Many algorithms developed in the machine learning community focus on learning in nominal feature spaces (Michalski & Stepp 1983, Kohavi 1994). However, many real-world classification tasks exist that involve continuous features where such algorithms could not be applied unless the continuous features are first discretized. Continuous variable discretization has received significant attention in the machine learning community only recently. Often, uniform binning of the data is used to produce the necessary data transformations for a learning algorithm, and no careful study of how this discretization affects the learning process is performed (Weiss & Kulikowski 1991). In decision tree methods, such as C4.5 (Quinlan 1993), continuous values are discretized during the learning process. The advantages of discretizing during the learning process have not yet been shown. In this paper, we include such a comparison.

Other reasons for variable discretization, aside from the algorithmic requirements mentioned above, include increasing the speed of induction algorithms (Catlett 1991*b*) and viewing General Logic Diagrams (Michalski 1978) of the induced classifier. In this paper, we address the effects of discretization on learning accuracy by comparing a range of discretization methods using C4.5 and a Naive Bayes classifier. The Naive-Bayes classifier is the one implemented in $\mathcal{MLC}$++ (Kohavi, John, Long, Manley & Pfleger 1994), which is described in Langley, Iba & Thompson (1992).

There are three different axes by which discretization methods can be classified: *global* vs. *local*, *supervised* vs. *unsupervised*, and *static* vs. *dynamic*.

Local methods, as exemplified by C4.5, produce partitions that are applied to localized regions of the instance space. Global methods (Chmielewski & Grzymala-Busse 1994), such as binning, produce a *mesh* over the entire n-dimensional continuous instance space, where each feature is partitioned into regions independent of the other attributes. The mesh contains $\prod_{i=1}^{n} k_i$ regions, where k_i is the number of partitions of the ith feature.

Several discretization methods, such as equal width interval binning, do not make use of instance labels in the discretization process. In analogy to supervised versus unsupervised learning methods, we refer to these as *unsupervised discretization* methods. In contrast, discretization methods that utilize the class labels are referred to as *supervised discretization* methods.

We believe that differentiating static and dynamic discretization is also important. Many discretization methods require some parameter, k, indicating the maximum number of intervals to produce in discretizing a feature. Static methods, such as binning, entropy-based partitioning (Catlett 1991*b*, Fayyad & Irani 1993, Pfahringer 1995), and the 1R algorithm (Holte 1993), perform one discretization pass of the data for each feature and determine the value of k for each feature independent of the other features. Dynamic methods conduct a search through the space of possible k values for all features simultaneously, thereby capturing interdependencies in feature discretization. While we believe such methods are a promising avenue of research, we do not pursue these methods in this paper.

We present work related to feature discretization in Section 2. In Section 3, we describe in detail the methods we used in our comparative study of discretization techniques. We explain our experiments and results in Section 4. Section 5 and 6 are reserved for a discussion and summary of this work.

2 Related Work

The simplest discretization method, *Equal Interval Width*, merely divides the range of observed values for a variable into k equal sized bins, where k is a user-supplied parameter. As Catlett (1991*a*) points out, this type of discretization is vulnerable to outliers that may drastically skew the range. A related method, *Equal Frequency Intervals*, divides a continuous variable into k bins where (given m instances) each bin contains m/k (possibly duplicated) adjacent values.

Since these unsupervised methods do not utilize instance labels in setting partition boundaries, it is likely that classification information will be lost by binning as a result of combining values that are strongly associated with different classes into the same bin (Kerber 1992). In some cases this could make effective classification much more difficult.

A variation of equal frequency intervals—maximal marginal entropy— adjusts the boundaries to decrease entropy at each interval (Chmielewski & Grzymala-Busse 1994, Wong & Chiu 1987).

Holte (1993) presented a simple example of a supervised discretization method. His 1R algorithm attempts to divide the domain of every continuous variable into pure bins, each containing a strong majority of one particular class with the constraint that each bin must include at least some prespecified number of instances. This method appears to work reasonably well when used in conjunction with the 1R induction algorithm.

The ChiMerge system (Kerber 1992) provides a statistically justified heuristic method for supervised discretization. This algorithm begins by placing each observed real value into its own interval and proceeds by using the χ^2 test to determine when adjacent intervals should be merged. This method tests the hypothesis that the two adjacent intervals are statistically independent by making an empirical measure of the expected frequency of the classes represented in each of the intervals. The extent of the merging process is controlled by the use of a χ^2 *threshold*, indicating the maximum χ^2 value that warrants merging two intervals. The author reports that on random data a very high threshold must be set to avoid creating too many intervals.

Another method for using statistical tests as a means of determining discretization intervals, *StatDisc*, has been proposed by Richeldi & Rossotto (1995). Similar in flavor to ChiMerge, this bottom-up method creates a hierarchy of discretization intervals using the Φ measure as a criterion for merging intervals. StatDisc is more general than ChiMerge, however, in that it considers merging up to N adjacent intervals at a time (where N is a user-set parameter), rather than just two adjacent intervals at a time as in ChiMerge. Merging of intervals continues until some Φ threshold is achieved. The final hierarchy of discretizations can then be explored and a suitable final discretization automatically selected.

A number of entropy-based methods have recently come to the forefront of work on discretization. Chiu, Cheung & Wong (1990) have proposed a hierarchical discretization method based on maximizing the Shannon entropy over the discretized space. This method uses a hill-climbing search to find a suitable initial partition of the continuous space into k bins along each axis and then re-applies this method to particular intervals to obtain finer intervals. This method has been applied primarily to an information synthesis task yet it bears strong similarities to work in discretization by machine learning researchers.

Catlett (1991*b*) has explored the use of entropy-based discretization in decision tree domains as a means of achieving an impressive increase in the speed of induction on very large data sets with many continuous features. His D-2 discretizer uses several conditions as criteria for stopping the recursive formation of partitions for each attribute: a minimum number of samples in one partition, a maximum number of partitions, and a minimum information gain.

Fayyad & Irani (1993) use a recursive entropy min-

	Global	Local
Supervised	1RD (Holte) Adaptive Quantizers ChiMerge (Kerber) D-2 (Catlett) Fayyad and Irani / Ting Supervised MCC Predictive Value Max.	Vector Quantization Hierarchical Maximum Entropy Fayyad and Irani C4.5
Unsupervised	Equal width interval Equal freq. interval Unsupervised MCC	k-means clustering

Table 1: Summary of discretization methods

imization heuristic for discretization and couple this with a *Minimum Description Length* criterion (Rissanen 1986) to control the number of intervals produced over the continuous space. In the original paper, this method was applied locally at each node during tree generation. The method was found to be quite promising as a global discretization method (Ting 1994), and in this paper the method is used for global discretization.

Pfahringer (1995) uses entropy to select a large number of candidate split-points and employs a best-first search with a Minimum Description Length heuristic to determine a good discretization.

Adaptive Quantizers (Chan, Batur & Srinivasan 1991) is a method combining supervised and unsupervised discretization. One begins with a binary equal width interval partitioning of the continuous feature. A set of classification rules are then induced on the discretized data (using an ID3-like algorithm) and tested for accuracy in predicting discretized outputs. The interval that has the lowest prediction accuracy is then split into two partitions of equal width and the induction and evaluation processes are repeated until some performance criteria is obtained. While this method does appear to overcome some of the limitations of unsupervised binning, it has a high computational cost as the rule induction process must be repeated numerous times. Furthermore, the method makes an implicit assumption that high accuracy can be attained. For example, on random data, the system might make many splits and a post-processing step needs to be added.

Bridging the gap between supervised and unsupervised methods for discretization, Van de Merckt (1993) developed two methods under the general heading of *Monothetic Contrast Criterions* (MCC). The first criterion, dubbed *unsupervised* by the author, makes use of an unsupervised clustering algorithm that seeks to find the partition boundaries that "produce the greatest contrast" according to a given contrast function. The second method, referred to as *mixed supervised/unsupervised*, simply redefines the objective function to be maximized by dividing the previous contrast function by the entropy of a proposed partition. Since calculating the entropy for the candidate partition requires class label information, this method can be thought of as supervised. Chmielewski & Grzymala-Busse (1994) have taken a similar approach using a cluster-based method to find candidate interval boundaries and then applying an entropy-based *consistency* function from the theory of *Rough Sets* to evaluate these intervals.

The *Predicative Value Maximization* algorithm (Weiss, Galen & Tadepalli 1990) makes use of a supervised discretization method by finding partition boundaries with locally maximal *predictive values*—those most likely to make correct classification decisions. The search for such boundaries begins at a coarse level and is refined over time to find locally optimal partition boundaries.

Dynamic programming methods have been applied to find interval boundaries for continuous features (Fulton, Kasif & Salzberg 1994). In such methods, each pass over the observed values of the data can identify a new partition on the continuous space based on the intervals already identified up to that point. This general framework allows for a wide variety of impurity functions to be used to measure the quality of candidate splitting points. Maass (1994) has recently introduced a dynamic programming algorithm which finds the minimum training set error partitioning of a continuous feature in $O(m(log\ m + k^2))$ time, where k is the number of intervals and m is the number of instances. This method has yet to be tested experimentally.

Vector Quantization (Kohonen 1989) is also related to the notion of discretization. This method attempts to partition an N-dimensional continuous space into a *Voronoi Tessellation* and then represent the set of points in each region by the region into which it falls. This discretization method creates local regions and is thus a local discretization method. Alternatively, it can be thought of as a complete *instance space* discretization as opposed to the *feature space* discretizations discussed here.

Table 1 shows a summary of these discretization methods, identified by the global/local and supervised/unsupervised dimensions. All the methods presented are static discretizers.

3 Methods

In our study, we consider three methods of discretization in depth: equal width intervals, 1RD, the method proposed by Holte for the 1R algorithm, and the entropy minimization heuristic (Fayyad & Irani 1993, Catlett 1991*b*).

3.1 Equal Width Interval Binning

Equal width interval binning is perhaps the simplest method to discretize data and has often been applied as a means for producing nominal values from continuous ones. It involves sorting the observed values of a continuous feature and dividing the range of observed values for the variable into k equally sized bins, where k is a parameter supplied by the user. If a variable x is observed to have values bounded by x_{min} and x_{max} then this method computes the bin width

$$\delta = \frac{x_{max} - x_{min}}{k}$$

and constructs bin boundaries, or *thresholds*, at $x_{min} + i\delta$ where $i = 1, ..., k-1$. The method is applied to each continuous feature independently. It makes no use of instance class information whatsoever and is thus an unsupervised discretization method.

3.2 Holte's 1R Discretizer

Holte (1993) describes a simple classifier that induces one-level decision trees, sometimes called decision stumps (Iba & Langley 1992). In order to properly deal with domains that contain continuous valued features, a simple supervised discretization method is given. This method, referred to here as 1RD (One-Rule Discretizer), sorts the observed values of a continuous feature and attempts to greedily divide the domain of the feature into bins that each contain only instances of one particular class. Since such a scheme could possibly lead to one bin for each observed real value, the algorithm is constrained to forms bins of at least some minimum size (except the rightmost bin). Holte suggests a minimum bin size of 6 based on an empirical analysis of 1R on a number of classification tasks, so our experiments used this value as well. Given the minimum bin size, each discretization interval is made as "pure" as possible by selecting cut-points such that moving a partition boundary to add an observed value to a particular bin cannot make the count of the dominant class in that bin greater.

3.3 Recursive Minimal Entropy Partitioning

A method for discretizing continuous attributes based on a minimal entropy heuristic, presented in Catlett (1991*b*) and Fayyad & Irani (1993), is also used in our experimental study. This supervised algorithm uses the class information entropy of candidate partitions to select bin boundaries for discretization. Our notation closely follows the notation of Fayyad and Irani. If we are given a set of instances S, a feature A, and a partition boundary T, the class information entropy of the partition induced by T, denoted $E(A, T; S)$ is given by:

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

For a given feature A, the boundary T_{min} which minimizes the entropy function over all possible partition boundaries is selected as a binary discretization boundary. This method can then be applied recursively to both of the partitions induced by T_{min} until some stopping condition is achieved, thus creating multiple intervals on the feature A.

Fayyad and Irani make use of the *Minimal Description Length Principle* to determine a stopping criteria for their recursive discretization strategy. Recursive partitioning within a set of values S stops iff

$$Gain(A, T; S) < \frac{log_2(N-1)}{N} + \frac{\Delta(A, T; S)}{N},$$

where N is the number of instances in the set S,

$$Gain(A, T; S) = Ent(S) - E(A, T; S),$$

$$\Delta(A,\ T; S) = log_2(3^k - 2) - [k \cdot Ent(S) - k_1 \cdot Ent(S_1) - k_2 \cdot Ent(S_2)],$$

and k_i is the number of class labels represented in the set S_i. Since the partitions along each branch of the recursive discretization are evaluated independently using this criteria, some areas in the continuous spaces will be partitioned very finely whereas others (which have relatively low entropy) will be partitioned coarsely.

	Dataset	Features		Train	Test	Majority
		continuous	nominal	sizes		Accuracy
1	anneal	6	32	898	CV-5	76.17±0.10
2	australian	6	8	690	CV-5	55.51±0.18
3	breast	10	0	699	CV-5	65.52±0.14
4	cleve	6	7	303	CV-5	54.46±0.22
5	crx	6	9	690	CV-5	55.51±0.18
6	diabetes	8	0	768	CV-5	65.10±0.16
7	german	24	0	1000	CV-5	70.00±0.00
8	glass	9	0	214	CV-5	35.51±0.45
9	glass2	9	0	163	CV-5	53.37±0.56
10	heart	13	0	270	CV-5	55.56±0.00
11	hepatitis	6	13	155	CV-5	79.35±0.79
12	horse-colic	7	15	368	CV-5	63.04±0.25
13	hypothyroid	7	18	2108	1055	95.45±0.05
14	iris	4	0	150	CV-5	33.33±0.00
15	sick-euthyroid	7	18	2108	1055	90.89±0.06
16	vehicle	18	0	846	CV-5	25.53±0.09

Table 2: Datasets and baseline accuracy

4 Results

In our experimental study, we compare the discretization methods in Section 3 as a preprocessing step to the C4.5 algorithm and a Naive-Bayes classifier. The C4.5 induction algorithm is a state-of-the-art top-down method for inducing decision trees. The Naive-Bayes induction algorithm computes the posterior probability of the classes given the data, assuming independence between the features for each class. The probabilities for nominal features are estimated using counts, and a Gaussian distribution is assumed for continuous features.

The number of bins, k, in the equal width interval discretization was set to both $k = 10$ and $k = \max\{1, 2 \cdot \log \ell\}$, where l is the number of distinct observed values for each attribute. The heuristic was chosen based on examining S-plus's histogram binning algorithm (Spector 1994).

We chose sixteen datasets from the U.C. Irvine repository (Murphy & Aha 1994) that each had at least one continuous feature. For the datasets that had more than 3000 test instances, we ran a single train/test experiment and report the theoretical standard deviation estimated using the Binomial model (Kohavi 1995). For the remaining datasets, we ran five-fold cross-validation and report the standard deviation of the cross-validation.

Table 2 describes the datasets with the last column showing the accuracy of predicting the majority class on the test set. Table 3 shows the accuracies of the C4.5 induction algorithm (Quinlan 1993) using the different discretization methods. Table 4 shows the accuracies of the Naive-Bayes induction algorithm. Figure 1 shows a line plot of two discretization methods: $\log \ell$-binning and entropy. We plotted the difference between the accuracy after discretization and the induction algorithm's original accuracy.

5 Discussion

Our experiments reveal that all discretization methods for the Naive-Bayes classifier lead to a large average increase in accuracy. Specifically, the best method—entropy—improves performance on all but three datasets, where the loss is insignificant. On seven out of 16, the entropy discretization method provides a significant increase in accuracy. We attribute this disparity in accuracy to the shortcomings of the Gaussian distribution assumption that is inappropriate in some domains. As observed by Richeldi & Rossotto (1995), discretization of a continuous feature can roughly approximate the class distribution for the feature and thus help to overcome the normality assumption used for continuous features in the Naive-Bayesian classifier we used.

C4.5's performance was significantly improved on two datasets—cleve and diabetes—using the entropy discretization method and did not significantly degrade on any dataset, although it did decrease slightly on some. The entropy-based discretization is a global method and does not suffer from data fragmentation (Pagallo & Haussler 1990). Since there is no significant

	Dataset	C4.5				
		Continuous	Bin-log ℓ	Entropy	1RD	Ten Bins
1	anneal	91.65±1.60	90.32±1.06	89.65±1.00	87.20±1.66	89.87±1.30
2	australian	85.36±0.74	84.06±0.97	85.65±1.82	85.22±1.35	84.20±1.20
3	breast	94.71±0.37	94.85±1.28	94.42±0.89	94.99±0.68	94.57±0.97
4	cleve	73.62±2.25	76.57±2.60	79.24±2.41	79.23±2.48	77.58±3.31
5	crx	86.09±0.98	84.78±1.82	84.78±1.94	85.51±1.93	84.64±1.64
6	diabetes	70.84±1.67	73.44±1.07	76.04±0.85	72.40±1.72	72.01±1.07
7	german	72.30±1.37	71.10±0.37	74.00±1.62	70.10±0.94	70.10±0.48
8	glass	65.89±2.38	59.82±3.21	69.62±1.95	59.31±2.07	59.83±2.04
9	glass2	74.20±3.72	80.42±3.55	76.67±1.63	71.29±5.10	74.32±3.80
10	heart	77.04±2.84	78.52±1.72	81.11±3.77	82.59±3.39	80.74±0.94
11	hepatitis	78.06±2.77	80.00±2.37	75.48±1.94	79.35±4.28	80.00±2.37
12	horse-colic	84.78±1.31	85.33±1.23	85.60±1.25	85.60±1.24	85.33±1.23
13	hypothyroid	99.20±0.27	97.30±0.49	99.00±0.30	98.00±0.43	96.30±0.58
14	iris	94.67±1.33	96.00±1.25	94.00±1.25	94.00±1.25	96.00±1.25
15	sick-euthyroid	97.70±0.46	94.10±0.72	97.30±0.49	97.40±0.49	95.70±0.62
16	vehicle	69.86±1.84	68.45±2.19	69.62±1.57	66.80±3.39	68.33±2.12
	Average	82.25	82.19	83.26	81.81	81.84

Table 3: Accuracies using C4.5 with different discretization methods. Continuous denotes running C4.5 on the undiscretized data; Bin-log ℓ and Ten Bins use equal-width binning with the respective number of intervals; Entropy refers to a global variant of the discretization method proposed by Fayyad and Irani.

	Dataset	Naive-Bayes				
		Continuous	Bin-log ℓ	Entropy	1RD	Ten Bins
1	anneal	64.48±1.47	95.99±0.59	97.66±0.37	95.44±1.02	96.22±0.64
2	australian	77.10±1.58	85.65±0.84	86.09±1.06	84.06±1.02	85.07±0.75
3	breast	96.14±0.74	97.14±0.50	97.14±0.50	97.14±0.60	97.28±0.52
4	cleve	84.19±2.01	83.86±3.10	82.87±3.11	81.86±1.84	82.21±2.63
5	crx	78.26±1.15	84.78±1.17	86.96±1.15	85.22±1.25	85.07±1.35
6	diabetes	75.00±1.77	74.87±1.39	74.48±0.89	72.14±1.52	75.00±1.74
7	german	72.60±2.65	75.60±0.87	73.30±1.38	71.80±1.29	74.40±1.19
8	glass	47.19±0.71	70.13±2.39	71.52±1.93	69.19±3.18	62.66±3.11
9	glass2	59.45±2.83	76.04±3.06	79.17±1.71	82.86±1.46	77.88±2.52
10	heart	84.07±2.24	82.22±2.72	81.48±3.26	81.85±2.44	82.96±2.77
11	hepatitis	84.52±3.29	83.87±4.08	84.52±4.61	83.87±4.67	85.81±4.16
12	horse-colic	80.14±2.45	79.60±2.52	80.96±2.50	80.13±3.17	80.14±2.09
13	hypothyroid	97.82±0.44	97.54±0.47	98.58±0.36	98.29±0.40	97.25±0.50
14	iris	95.33±1.33	96.00±1.25	94.00±1.25	93.33±1.05	95.33±1.70
15	sick-euthyroid	84.64±1.11	88.44±0.98	95.64±0.62	94.98±0.67	91.09±0.87
16	vehicle	44.21±1.58	60.76±1.75	59.22±1.56	62.18±1.88	60.29±2.32
	Average	76.57	83.28	83.97	83.40	83.00

Table 4: Accuracies using Naive-Bayes with different discretization methods

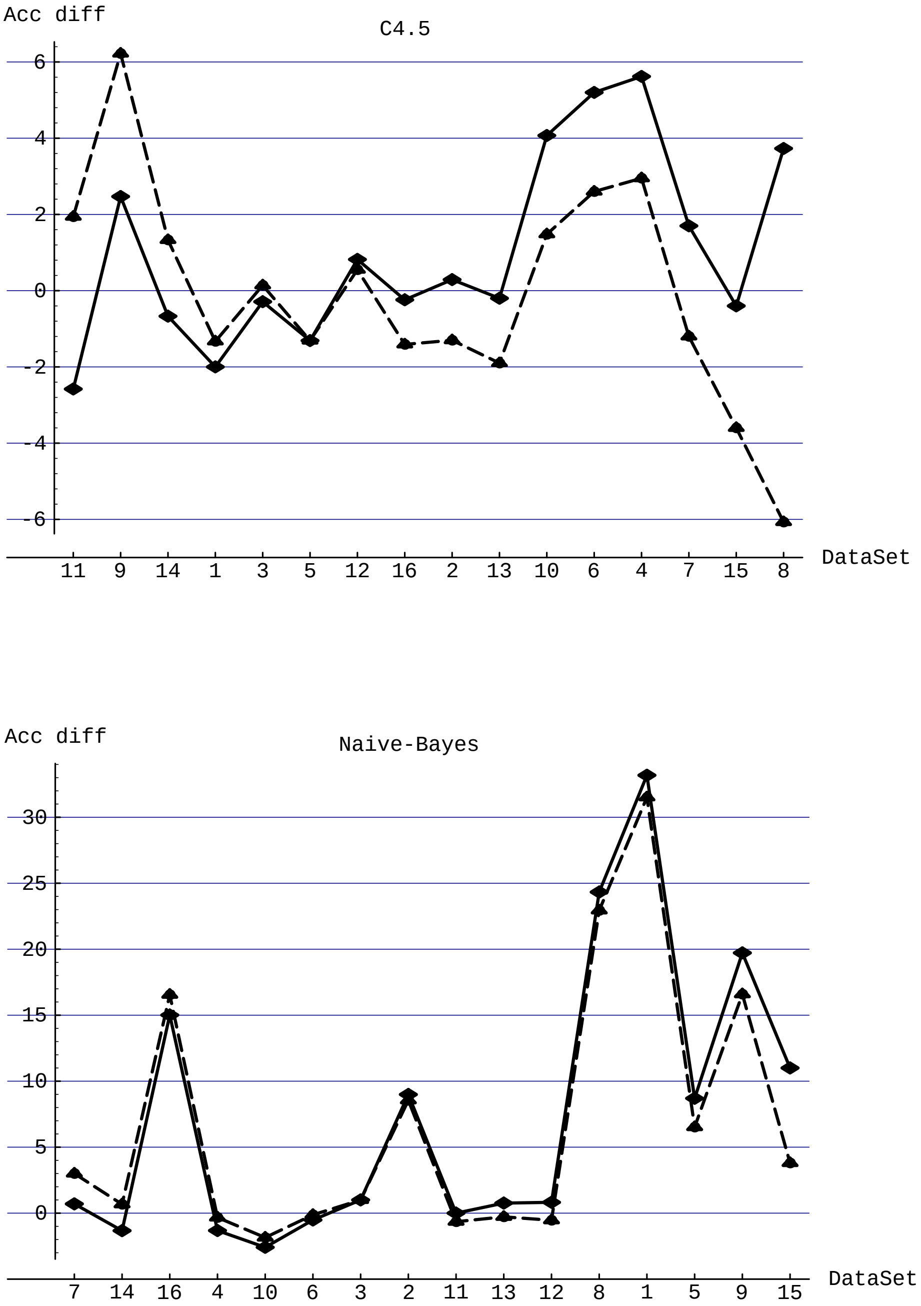

Figure 1: Comparison of entropy (solid) and $\log \ell$-binning (dashed). Graphs indicate accuracy difference from undiscretized C4.5/Naive-Bayes. The 0% line indicates the performance of C4.5/Naive-Bayes without prior discretization. The datasets were arranged by increasing differences between the discretization methods.

degradation in accuracy when a global discretization method is used, we conjecture that the C4.5 induction algorithm is not taking full advantage of possible local discretization that could be performed on the data or that such local discretization cannot help the induction process for the datasets we tested.

One possible advantage to global discretization as opposed to local methods is that it provides regularization because it is less prone to variance in estimation from small fragmented data.

At the 95% confidence level, the Naive-Bayes with entropy-discretization is better than C4.5 on five datasets and worse on two. The average performance (assuming the datasets are coming from some real-world distribution) of the entropy-discretized Naive-Bayes is 83.97% compared to C4.5 at 82.25% and the original Naive-Bayes at 76.57%.

The supervised learning methods are slightly better than the unsupervised methods, although even simple binning tends to significantly increase performance of the Naive-Bayesian classifier that assumes a Gaussian distribution for continuous attributes.

6 Summary

We presented an empirical comparison of discretization for continuous attributes and showed that discretization prior to induction can sometimes significantly improve the accuracy of an induction algorithm. The global entropy-based discretization method seems to be the best choice of the discretization methods tested here.

We found that the entropy-discretized Naive-Bayes improved so much, that its average performance slightly surpassed that of C4.5. C4.5's performance did not degrade if data were discretized in advance using the entropy discretization method, and in two cases even improved significantly.

None of the methods tested was dynamic, *i.e.*, each feature was discretized independent of other features and of the algorithm's performance. We plan to pursue wrapper methods (John, Kohavi & Pfleger 1994) that search through the space of k values, indicating the number of intervals per attribute. Another variant that could be explored is local versus global discretization based on Fayyad & Irani's method.

Acknowledgments The work in this paper was done using the $\mathcal{MLC}$++ library, partly funded by ONR grant N00014-94-1-0448 and NSF grants IRI-9116399 and IRI-9411306. We thank Jason Catlett, Usama Fayyad, George John, and Bernhard Pfahringer for their useful comments. The third author is supported by a Fred Gellert Foundation ARCS scholarship.

References

Catlett, J. (1991*a*), Megainduction: machine learning on very large databases, PhD thesis, Univeristy of Sydney.

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

Chan, C.-C., Batur, C. & Srinivasan, A. (1991), Determination of quantization intervals in rule based model for dynamic systems, *in* "Proceedings of the IEEE Conference on Systems, Man, and Cybernetics", Charlottesvile, Virginia, pp. 1719–1723.

Chiu, D. K. Y., Cheung, B. & Wong, A. K. C. (1990), "Information synthesis based on hierarchical entropy discretization", *Journal of Experimental and Theoretical Artificial Intelligence* **2**, 117–129.

Chmielewski, M. R. & Grzymala-Busse, J. W. (1994), Global discretization of continuous attributes as preprocessing for machine learning, *in* "Third International Workshop on Rough Sets and Soft Computing", pp. 294–301.

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 Artificial Intelligence", Morgan Kaufmann, pp. 1022–1027.

Fulton, T., Kasif, S. & Salzberg, S. (1994), An efficient algorithm for finding multi-way splits for decision trees, Unpublished paper.

Holte, R. C. (1993), "Very simple classification rules perform well on most commonly used datasets", *Machine Learning* **11**, 63–90.

Iba, W. & Langley, P. (1992), Induction of one-level decision trees, *in* "Proceedings of the Ninth International Conference on Machine Learning", Morgan Kaufmann, pp. 233–240.

John, G., Kohavi, R. & Pfleger, K. (1994), Irrelevant features and the subset selection problem, *in* "Machine Learning: Proceedings of the Eleventh International Conference", Morgan Kaufmann, pp. 121–129. Available by anonymous ftp from: `starry.Stanford.EDU:pub/ronnyk/ml94.ps`.

Kerber, R. (1992), Chimerge: Discretization of numeric attributes, *in* "Proceedings of the Tenth National Conference on Artificial Intelligence", MIT Press, pp. 123–128.

Kohavi, R. (1994), Bottom-up induction of oblivious, read-once decision graphs : strengths and limitations, *in* "Twelfth National Conference on Artificial Intelligence", pp. 613–618. Available by anonymous ftp from `Starry.Stanford.EDU:pub/ronnyk/aaai94.ps`.

Kohavi, R. (1995), A study of cross-validation and bootstrap for accuracy estimation and model selection, *in* "Proceedings of the 14th International Joint Conference on Artificial Intelligence". Available by anonymous ftp from `starry.Stanford.EDU:pub/ronnyk/accEst.ps`.

Kohavi, R., John, G., Long, R., Manley, D. & Pfleger, K. (1994), MLC++: A machine learning library in C++, *in* "Tools with Artificial Intelligence", IEEE Computer Society Press, pp. 740–743. Available by anonymous ftp from: `starry.Stanford.EDU:pub/ronnyk/mlc/toolsmlc.ps`.

Kohonen, T. (1989), *Self-Organization and Associative Memory*, Berlin, Germany: Springer-Verlag.

Langley, P., Iba, W. & Thompson, K. (1992), An analysis of bayesian classifiers, *in* "Proceedings of the tenth national conference on artificial intelligence", AAAI Press and MIT Press, pp. 223–228.

Maass, W. (1994), Efficient agnostic pac-learning with simple hypotheses, *in* "Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory", pp. 67–75.

Michalski, R. S. (1978), A planar geometric model for representing multidimensional discrete spaces and multiple-valued logic functions, Technical Report UIUCDCS-R-78-897, University of Illinois at Urbaba-Champaign.

Michalski, R. S. & Stepp, R. E. (1983), Learning from observations: Conceptual clustering, *in* T. M. M. R. S. Michalski, J. G. Carbonell, ed., "Machine Learning: An Artificial Intelligence Approach", Tioga, Palo Alto.

Murphy, P. M. & Aha, D. W. (1994), UCI repository of machine learning databases, For information contact ml-repository@ics.uci.edu.

Pagallo, G. & Haussler, D. (1990), "Boolean feature discovery in empirical learning", *Machine Learning* **5**, 71–99.

Pfahringer, B. (1995), Compression-based discretization of continuous attributes, *in* A. Prieditis & S. Russell, eds, "Proceedings of the Twelfth International Conference on Machine Learning", Morgan Kaufmann.

Quinlan, J. R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann, Los Altos, California.

Richeldi, M. & Rossotto, M. (1995), Class-driven statistical discretization of continuous attributes (extended abstract), *in* N. Lavrac & S. Wrobel, eds, "Machine Learning: ECML-95 (Proc. European Conf. on Machine Learning, 1995)", Lecture Notes in Artificial Intelligence 914, Springer Verlag, Berlin, Heidelberg, New York, pp. 335 – 338.

Rissanen, J. (1986), "Stochastic complexity and modeling", *Ann. Statist* **14**, 1080–1100.

Spector, P. (1994), *An Introduction to S and S-PLUS*, Duxbury Press.

Ting, K. M. (1994), Discretization of continuous-valued attributes and instance-based learning, Technical Report 491, University of Sydney.

Van de Merckt, T. (1993), Decision trees in numerical attribute spaces, *in* "Proceedings of the 13th International Joint Conference on Artificial Intelligence", pp. 1016–1021.

Weiss, S. M. & Kulikowski, C. A. (1991), *Computer Systems that Learn*, Morgan Kaufmann, San Mateo, CA.

Weiss, S. M., Galen, R. S. & Tadepalli, P. V. (1990), "Maximizing the predicative value of production rules", *Artificial Intelligence* **45**, 47–71.

Wong, A. K. C. & Chiu, D. K. Y. (1987), Synthesizing statistical knowledge from incomplete mixed-mode data, *in* "IEEE Transaction on Pattern Analysis and Machine Intelligence 9", pp. 796–805.