Discretization of Continuous Attributes for Learning Classification Rules

Aijun An and Nick Cercone

Department of Computer Science, University of Waterloo
Waterloo, Ontario N2L 3G1 Canada

Abstract. We present a comparison of three entropy-based discretization methods in a context of learning classification rules. We compare the binary recursive discretization with a stopping criterion based on the Minimum Description Length Principle (MDLP)[3], a non-recursive method which simply chooses a number of cut-points with the highest entropy gains, and a non-recursive method that selects cut-points according to both information entropy and distribution of potential cut-points over the instance space. Our empirical results show that the third method gives the best predictive performance among the three methods tested.

1 Introduction

Recent work on entropy-based discretization of continuous attributes has produced positive results [2, 6] . One promising method is Fayyad and Irani's binary recursive discretization with a stopping criterion based on the Minimum Description Length Principle (MDLP) [3]. The MDLP method is reported as a successful method for discretization in the decision tree learning and Naive-Bayes learning environments [2, 6]. However, little research has been done to investigate whether the method works well with other rule induction methods. We report our performance findings of the MDLP discretization in a context of learning classification rules. The learning system we use for experiments is ELEM2 [1], which learns classification rules from a set of training data by selecting the most relevant attribute-value pairs. We first compare the MDLP method with an entropy-based method that simply selects a number of entropy-lowest cut-points. The results show that the MDLP method fails to find sufficient useful cut-points, especially on small data sets. The experiments also discover that the other method tends to select cut-points from a small local area of the entire value space, especially on large data sets. To overcome these problems, we introduce a new entropy-based discretization method that selects cut-points based on both information entropy and distribution of potential cut-points. Our conclusion is that MDLP does not give the best results in most tested datasets. The proposed method performs better than MDLP in the ELEM2 learning environment.

2 The MDLP Discretization Method

Given a set S of instances, an attribute A, and a cut-point T, the class information entropy of the partition induced by T, denoted as $E(A, T; S)$, is defined

N. Zhong and L. Zhou (Eds.): PAKDD'99, LNAI 1574, pp. 509-514, 1999.

as

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

where $Ent(S_i)$ is the class entropy of the subset S_i, defined as

$$Ent(S_i) = -\sum_{j=1}^{k} P(C_j, S_i)log(P(C_j, S_i)),$$

where there are k classes $C_1, \cdots, C_k$ and $P(C_j, S_i)$ is the proportion of examples in S_i that have class C_j. For an attribute A, the MDLP method selects a cut point T_A for which $E(A, T_A; S)$ is minimal among all the boundary points[1]. The training set is then split into two subsets by the cut point. Subsequent cut points are selected by recursively applying the same binary discretization method to each of the newly generated subsets until the following condition is achieved:

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

where N is the number of examples in S, $Gain(A,T;S) = Ent(S) - E(A,T;S)$, $\Delta(A,T;S) = log_2(3^k - 2) - [kEnt(S) - k_1Ent(S_1) - k_2Ent(S_2)]$, and k, k_1 and k_2 are the number of classes represented in the sets S, S_1 and S_2, respectively. Empirical results, presented in [3], show that the MDLP stopping criterion leads to construction of better decision trees. Dougherty *et al.* [2] also show that a global variant of the MDLP method significantly improved a Naive-Bayes classifier and it also performs best among several discretization methods in the context of C4.5 decision tree learning.

3 Experiments with MDLP Discretization and ELEM2

We conducted experiments with two versions of ELEM2. Both versions employ the entropy-based discretization method, but with different stopping criteria. One version uses the global variant of the MDLP discretization method, i.e., it discretizes continuous attributes using the recursive entropy-based method with the MDLP stopping criterion applied before rule induction begins. The other version uses the same entropy criterion for selecting cut-points before rule induction, but it simply chooses a maximal number of m entropy-lowest cut-points without recursive application of the method. m is set to be $max\{2, k * log_2 l\}$ where l is the number of distinct observed values for the attribute being discretized and k is the number of classes. We refer to this method as Max-m. Both versions first sort the examples according to their values of the attribute and then evaluate only the boundary points in their search for cut-points.

We first conduct the experiments on an artificial data set. Each example in the data set has two continuous attributes and a symbolic attribute. The

[1] Fayyad and Irani proved that the value T_A that minimizes the class entropy $E(A, T_A; S)$ must always be a value between two examples of different classes in the sequence of sorted examples. These kinds of values are called boundary points.

Training Set Size	Predictive accuracy		No. of cut-points		No. of rules		No. of Boundary Points
	MDLP	Max-m	MDLP	Max-m	MDLP	Max-m	
47	56.71%	95.20%	0	14	3	6	58
188	90.41%	100%	2	21	4	6	96
470	100%	100%	5	22	6	6	97
1877	100%	100%	29	22	6	6	97
4692	100%	100%	73	22	6	6	97

Table 1. Results on the Artificial Domain.

two continuous attributes, named $A1$ and $A2$, have value ranges of $[0, 90]$ and $[0, 5]$, respectively. The symbolic attribute, *color*, takes one of the four values: *red, blue, yellow* and *green*. An example belongs to class "1", if the following condition holds: $(30 < A1 \leq 60) \wedge (1.5 < A2 \leq 3.5) \wedge (color = blue\ or\ green)$; otherwise, it belongs to class "0". The data set has a total of 9384 examples. We randomly chose 6 training sets from these examples. The sizes of the training sets range from 47 examples (0.5%) to 4692 examples (50%). We run the two versions of ELEM2 on each of the 6 training sets to generate a set of decision rules. The rules are then tested on the original data set of 9384 examples. Table 1 depicts, for all the training sets, the predictive accuracy, the total number of cut-points selected for both continuous attributes, the total number of rules generated for both classes, and the number of boundary points for both continuous attributes. The results indicate that, when the number of training examples is small, the MDLP method stops too early and fails to find enough useful cut-points, which causes ELEM2 to generate rules that have poor predictive performance on the testing set. When the size of the training set increases, MDLP generates more cut-points and its predictive performance improves. For the middle-sized training set (470 examples), MDLP works perfectly because it finds only 5 cut-points from 97 boundary points, which include all of the four right cut-points that the learning system needs to generate correct rules. However, when the training set becomes larger, the number of cut-points MDLP finds increases greatly. In the last training set (4692 examples), it selects 73 cut-points out of 97 potential points, which slows down the learning system. In contrast, the Max-m method is more stable. The number of cut-points it produces ranges from 14 to 22 and its predictive performance is better than MDLP when the training set is small. We also run the two versions of ELEM2 on a number of actual data sets obtained from the UCI repository [4], each of which has at least one continuous attribute. Table 2 reports the ten-fold evaluation results on 6 of these data sets.

4 Discussion

The empirical results presented above indicate that MDLP is not superior to Max-m in most of tested data sets. One possible reason is that, when the training set is small, the examples are not sufficient to make the MDLP criterion valid and meaningful so that the criterion causes the discretization process to stop too

Data Set	Number of Examples	Predictive accuracy		Average no. of rules	
		MDLP	Max-m	MDLP	Max-m
bupa	345	57.65%	66.93%	4	65
german	1000	68.30%	68.50%	107	100
glass	214	63.14%	68.23%	31	30
heart	270	81.85%	82.59%	48	30
iris	150	95.33%	96.67%	8	7
segment	2310	95.76%	90.65%	67	99

Table 2. Results on the Actual Data Sets.

early before producing useful cut-points. Another possible reason is that, even if the recursive MDLP method is applied to the entire instance space to find the first cut-point, it is applied "locally" in finding subsequent cut-points due to the recursive nature of the method. Local regions represent smaller samples of the instance space and the estimation based on small samples using the MDLP criterion may not be reliable.

Now that MDLP does not seem to be a good discretization method for ELEM2, is Max-m a reliable method? A close examination of the cut-points produced by Max-m for the *segment* data set uncovers that, for several attributes, the selected cut-points concentrate on a small local area of the entire value space. For example, for an attribute that ranges from 0 to 43.33, Max-m picks up 64 cut-points all of which fall between 0.44 and 4, even if there are many boundary cut-points lying out of this small area. This problem is caused by the way the Max-m method selects cut-points. Max-m first selects the cut-point that has the lowest entropy value and then selects as the next point the point with the second lowest entropy, and so on. This strategy may result in a large number of cut-points being selected near the first cut-point because their entropy values are closer to the entropy value of the first cut-point than the entropy values of the cut-points located far from the first cut-point. The cut-points located on a small area around the first cut-point offer very little additional discriminating power because the difference between them and the first cut-point involves only a few examples. In addition, since only the first m cut-pints are selected by Max-m, selecting too many cut-points in a small area may prohibit the algorithm from choosing the promising points in other regions.

5 A Revised Max-m Discretization Method

To overcome the weakness of the Max-m method, we propose a new entropy-based discretization method by revising Max-m. The new method avoids selecting cut-points within only one or two small areas. The new method chooses cut-points according to both information entropy and the distribution of boundary points over the instance space. The method is referred to as EDA-DB (Entropy-based Discretization According to Distribution of Boundary points). Similar to Max-m, EDA-DB selects a maximal number of m cut-points, where m is defined

as in the Max-m method. However, rather than taking the first m entropy-lowest cut-points, EDA-DB divides the value range of the attribute into intervals and selects in each interval m_i number of cut-points based on the entropy calculated over the entire instance space. m_i is determined by estimating the probability distribution of the boundary points over the instance space. The EDA-DB discretization algorithm is described as follows. Let l be the number of distinct observed values for a continuous attribute A, b be the total number of boundary points for A, and k be the number of classes in the data set. To discretize A,

1. Calculate m as $max\{2, k * log_2(l)\}$.
2. Estimate the probability distribution of boundary points:
   (a) Divide the value range of A into d intervals, where $d = max\{1, log(l)\}$.
   (b) Calculate the number b_i of boundary points in each interval iv_i, where $i = 1, 2, \cdots, d$ and $\sum_{i=1}^{d} b_i = b$.
   (c) Estimate the probability of boundary points in each interval iv_i $(i = 1, 2, \cdots, d)$ as $p_i = \frac{b_i}{b}$
3. Calculate the quota q_i of cut-points for each interval iv_i $(i = 1, 2, \cdots, d)$ according to m and the distribution of boundary points as follows: $q_i = p_i * m$
4. Rank the boundary points in each interval iv_i $(i = 1, 2, \cdots, d)$ by increasing order of the class information entropy of the partition induced by the boundary point. The entropy for each point is calculated globally over the entire instance space.
5. For each interval iv_i $(i = 1, 2, \cdots, d)$, select the first q_i points in the above ordered sequence. A total of m cut-points are selected.

6 Experiments with EDA-DB

We conducted experiments with EDA-DB coupled with ELEM2. We first conducted ten-fold evaluation on the *segment* data set to see whether EDA-DB improves over Max-m on this data set which has a large number of boundary points for several attributes. The result is that the predictive accuracy is increased to 95.11% and the average number of rules drops to 69. Figure 1 shows the ten-fold evaluation results on 14 UCI data sets. In the figure, the solid line represents the difference between EDA-DB's predictive accuracy and Max-m's, and the dashed line represents the accuracy difference between EDA-DB and MDLP. The results indicate that EDA-DB outperforms both Max-m and MDLP on most of the tested data sets.

7 Conclusions

We have presented an empirical comparison of three entropy-based discretization methods in a context of learning decision rules. We found that the MDLP method stops two early when the number of training examples is small and thus it fails to detect sufficient cut-points on small data sets. Our empirical results also indicate that Max-m and EDA-DB are better discretization methods for ELEM2 on most of the tested data sets. We conjecture that the recursive nature of the MDLP method may cause most of the cut-points to be selected based on small

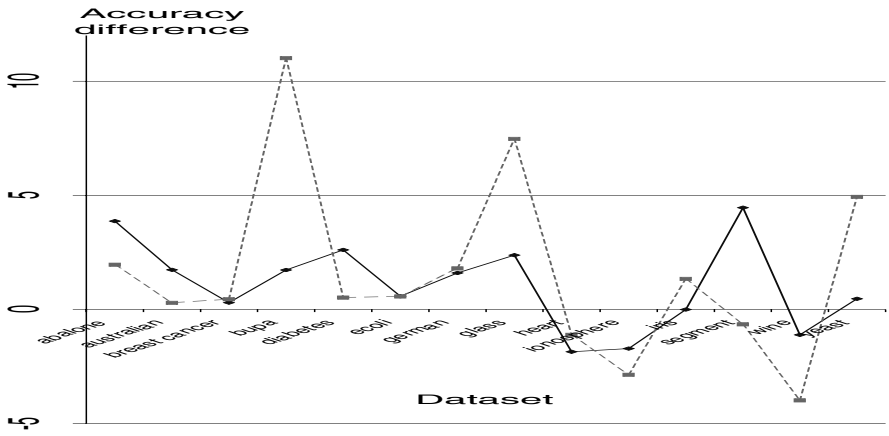

Fig. 1. Ten-fold Evaluation Results on Actual Data Sets.

samples of the instance space, which leads to generation of unreliable cut-points. The experiment with Max-m on the *segment* data set reveals that the strategy of simply selecting the first m entropy-lowest cut-points does not work well on large data sets with large number of boundary points. The reason for this is that entropy-lowest cut-points tend to concentrate on a small region of the instance space, which leads to the algorithm failing to pick up useful cut-points in other regions. Our proposed EDA-DB method alleviates the Max-m's problem by considering the distribution of boundary points over the instance space. Our test of EDA-DB on the *segment* data set shows that EDA-DB improves over Max-m on both the predictive accuracy and the number of rules generated. The experiments with EDA-DB on other tested data sets also confirm that EDA-DB is a better method than both Max-m and MDLP.

References

1. An, A. and Cercone, N. 1998. ELEM2: A Learning System for More Accurate Classifications. *Lecture Notes in Artificial Intelligence 1418.*
2. Dougherty, J., Kohavi, R. and Sahami, M. 1995. Supervised and Unsupervised Discretization of Continuous Features. *Proceedings of the Twelfth International Conference on Machine Learning.* Morgan Kaufmann Publishers, San Francisco, CA.
3. Fayyad, U.M. and Irani, K.B. 1993. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. *IJCAI-93.* pp. 1022-1027.
4. Murphy, P.M. and Aha, D.W. 1994. *UCI Repository of Machine Learning Databases.* URL: http://www.ics.uci.edu/AI/ML/MLDBRepository.html.
5. Quinlan, J.R. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers. San Mateo, CA.
6. Rabaseda-Loudcher, S., Sebban, M. and Rakotomalala, R. 1995. Discretization of Continuous Attributes: a Survey of Methods. *Proceedings of the 2nd Annual Joint Conference on Information Sciences*, pp.164-166.