Multi-Interval Discretization Methods for Decision Tree Learning

Petra Perner and Sascha Trautzsch

Institute of Computer Vision and Applied Computer Sciences e.V.
Arno-Nitzsche-Str. 45, 04277 Leipzig, Germany
e-mail: ibaiperner@aol.com
WWW: http://members.aol.com/ibai

Abstract. Properly addressing the discretization process of continuos valued
features is an important problem during decision tree learning. This paper
describes four multi-interval discretization methods for induction of decision
trees used in dynamic fashion. We compare two known discretization methods
to two new methods proposed in this paper based on a histogram based method
and a neural net based method (LVQ). We compare them according to accuracy
of the resulting decision tree and to compactness of the tree. For our
comparison we used three data bases, IRIS domain, satellite domain and OHS
domain (ovariel hyper stimulation).

1 Introduction

Decision tree learning is a widely used method for pattern recognition and image
interpretation [10][11][13].
Properly addressing the discretization process of continuous-valued features is an
important problem during decision tree learning.
Decision tree learning algorithms like ID3 [8], C4.5 [9] and CART [1] use binary
discretization for continuous-valued features. However, sometimes multi-interval
discretization seems to be better than only binary discretization. It can lead to more
compact and more accurate decision trees whereas the explanation capability of the
decision tree to the user might be better.
In the paper we describe several multi-interval discretization methods and compare
them to binary discretization methods used in C4.5 according to accuracy of the
resulting decision tree and compactness of the tree. We focus our work to dynami-
cally and supervised discretization methods [2].
We use entropy-based multi-interval discretization method introduced by Fayyad and
Irani [3] and ChiMerge method described by Kerber [6]. Two new methods are
introduced: the first one is based on Learning Vector Quantization (LVQ) described
by Kohonen [7] and the second one is based on histogram evaluation.

2 Multi-Interval Discretization Methods

2.1 Entropy-Based Discretization Method (A)

Such algorithms like ID3 and C4.5 use a minimal entropy heuristic for discretization continuous attributes. These methods try to find a binary cut for each attribute. Following a method introduced by Fayyad and Irani [3], the minimal entropy criteria can also be used to find multi-level cuts for each attributes.

The algorithm use the class information entropy of candidate partitions to select binary boundaries for discretization. If there is a given 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:

$$E(A,T,S) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2).$$ (1)

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 be applied recursively to both of the partitions induced by T_{min} until some stopping condition is achieved, thus creating multiple intervals on feature A.

Whereas Fayyad and Irani make use of a Minimal Description Length Principle to determine a stopping criteria for their recursive discretization process, we predefine the number of cuts (2 and 3 cuts) allowed for each attribute.

One of the main problems with this discretization criteria is that it is relatively expensive. It must be evaluated $N-1$ times for each attribute (with N the number of attribute values). Typically, N is very large. Therefore, it would be good to have an algorithm which uses some assumption in order to reduce the computation time. Such an algorithm is described in Section 2.4.

2.2 ChiMerge Discretization Method (B)

The ChiMerge algorithm introduced by Kerber [6] consists of an initialization step and a bottom-up merging process, where intervals are continuously merged until a termination condition is met. Kerber used the ChiMerge method static. In our study we apply ChiMerge dynamically to discretization. The potential cut-points are investigated by testing two adjacent intervals by the χ^2 independence test. The statistical test values is:

$$\chi^2 = \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}}$$ (2)

where m=2 (the intervals being compared), k - number of classes, A_{ij} - number of examples in i-th interval and j-th class, R_i - number of examples in i-th interval

$$R_i = \sum_{j=1}^{k} A_{ij} \; ; \; C_j \text{ - number of examples in j-th class } \; C_j = \sum_{i=1}^{m} A_{ij} \; ;$$

$$\text{N -total number of examples } \; N = \sum_{j=1}^{k} C_j \; ; \; E_{ij} \text{ -expected frequency } \; E_{ij} = \frac{R_i \cdot C_j}{N} \; .$$

Firstly, all boundary points will used for cut-points. In the second step for each pair of adjacent intervals one compute the χ^2-value. The two adjacent intervals with the lowest χ^2-value will merge together. This step is repeated continuously until all χ^2-value exceeding a given threshold. The value for the threshold is determined by selecting a desired significance level and then using a table or formula to obtain the χ^2.

2.3 LVQ-based Discretization Method (C)

Vector quantization is also related to the notion of discretization. We use Learning Vector Quantization (LVQ) [9] for our experiment. LVQ is a supervised learning algorithm. This method attempts to define class regions in the input data space. Firstly, a number of codebook vectors $\mathbf{W}_i$ labeled by a class are placed into the input space. Usually several codebook vectors are assigned to each class.

The learning algorithm is realized as follow: After an initialization of the neural net, each learning sample is presented one or several times to the net. The input vector $\mathbf{X}$ will be compared to all codebook vectors $\mathbf{W}$ in order to find the closest codebook vector $\mathbf{W}_c$. The learning algorithm will try to optimize the similarity between the codebook vectors and the learning samples by shifting the codebook vectors in the direction of the input vector if the sample represents the same class as the closest codebook vector. In case of the codebook vector and the input vector having different classes the codebook vector gets shifted away from the input vector, so that the similarity between these two decrease. All other code book vectors remain unchanged. The following equations represent this idea:

$$\text{for equal classes: } W_c(t+1) = W_c(t) + \alpha(t) \cdot \left[X(t) - W_c(t) \right] \qquad (3)$$

$$\text{for different classes: } W_c(t+1) = W_c(t) - \alpha(t) \cdot \left[X(t) - W_c(t) \right] \qquad (4)$$

$$\text{For all other: } W_j(t+1) = W_j(t) \qquad (5)$$

This behavior of the algorithms we can employ for discretization. A potential cut point might be in the middle of the learned codebook vectors of two different classes. Figure 1 shows this method based on one attribute of the IRIS domain. For our experiment we use the LVQ 2.1 algorithm.

Since this algorithm tries to optimize the misclassification probability we expect to get good results. However, the proper initialization of the codebook vectors and the choice of learning rate $\alpha(t)$ is a crucial problem.
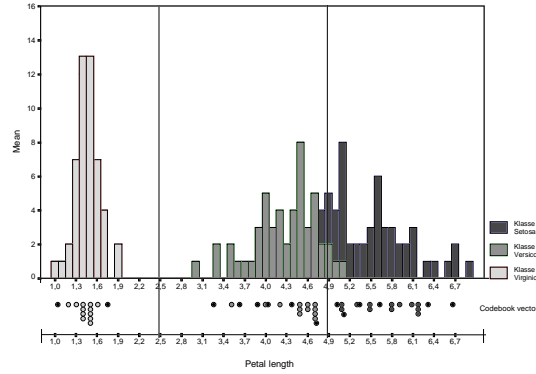
Advances in Pattern Recognition, A. Amin, D. Dori, P. Pudil, and H. Freeman (Eds.), LNCS 1451, Springer Verlag 1998, S. 475-482

Fig. 1. Class Distribution of an Attribute and Codebook Vectors

2.4 Histogram-Based Discretization Method (D)

A histogram-based method has been suggested first by Wu et al. [12]. They used this method in an interactive way during top-down decision tree building. By observing the histogram, the user selects the threshold which partitions the sample set in groups containing only samples of one class. In our experiment we use a histogram-based method in an automatic fashion as follows:

The distribution $p(a\,|\,a \in C_k)P(C_k)$ of one attribute a according to classes C_k is calculated. The curve of the distribution is approximated by a first order polynom and the minimum square error method is used for calculating the coefficients:

$$E = \sum_{i=1}^{n} (a_1 x_i + a_0 - y_i)^2 \qquad \textbf{(6)}$$

$$a_1 = \frac{\sum_{i=1}^{n} x_i \cdot i}{\sum_{i=1}^{n} i^2}$$

The cut points are selected by finding two maxima of different classes situated next to each other.

We used this method in two ways: First, we used the histogram-based discretization method as described before. Second, we used a combined discretization method based on the distribution $p(a\,|\,a \in S_k)P(S_k)$ and the entropy-based minimization criteria. We followed the corollary derived by Fayyad and Irani [3], which says that the entropy-based discretization criteria for finding a binary partition for a continuous attribute will always partition the data on a boundary point in the sequence of the examples ordered by the value of that attribute. A boundary point partitions the examples in two sets, having different classes. Taking into account this fact, we determine potential

boundary points by finding the peaks of the distribution. If we found two peaks belonging to different classes, we used the entropy-based minimization criteria in order to find the exact cut point between these two classes by evaluation each boundary point K with $P_i \leq K \leq P_{i+1}$ between this two peaks.
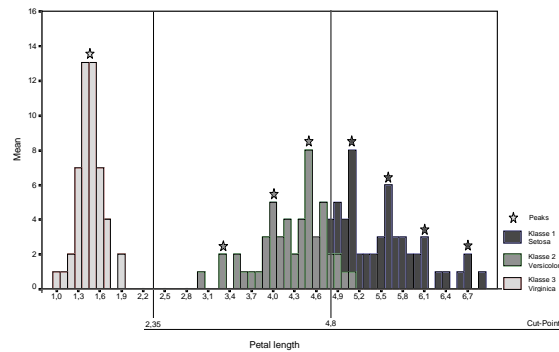

Fig. 2. Examples sorted by attribute values for attribute A and labelled peaks

This method is not as time consuming like the other ones. We wanted to see if this method can be an alternative to the methods described before and if we can find a hybrid version which combines the advantages of the low computation time of the histogram-based method with the entropy minimization heuristic in the context of discretization.

3 Results

In our experimental study, we used discretization methods during decision tree learning process. Therefore, we refer to the discretization methods used in this study as dynamically methods. During a decision tree learning process, the attribute gets discretized according to one of the above described methods first. Then the best attribute is selected according to the attribute selection criteria used in the C4.5 algorithm. Afterwards the splitting of the nodes takes place. This process repeats until no attribute can be selected and the tree building process stops. We compare the different method heuristically to the standard C4.5 method. We choose 3 data sets. Two standard domains: the IRIS domain and satellite images [5]. The satellite domain represented a 1907*1784 pixel TM-satellite image (Technical Mapping) of a 340 m² area of the Colorado. Each pixel (example) represented a 10 m² area and is labeled into 14 different classes of vegetation groups like water, grass, trees and so on. Each example is described by 6 different frequencies (red, green, blue, ...). There are 3.402.088 samples contained in the data set.

The third data set, called OHS domain, is a medical domain and contains a data set of 155 samples describing the over stimulation syndrome of the female menstruation cycle during the IVF-Therapy (In-Vitro-Fertilization) [4]. The Features are taken from

ultra sonic images (like number of follicle, size and so one) and blood values of the patient. It is a two class problem: over stimulation syndrome is possible or not.
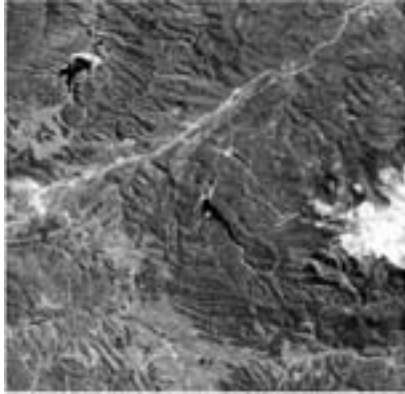

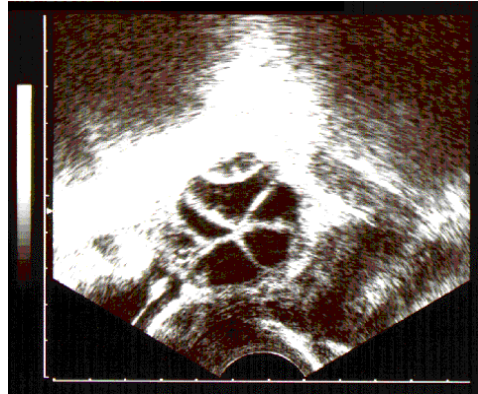**Fig. 3.** Satellite Image   **Fig. 4.** Ultra Sonic Image from Medical Domain

Table 1 shows the accuracy. For evaluation of accuracy we used cross validation method for IRIS- and OHS-domain and the test-and-train method for the satellite domain (train=15000 and test=3387088 examples).

Method	IRIS		OHS		Satellite Images	
	unpruned	pruned	unpruned	pruned	unpruned	pruned
A1	6.67%	4.67%	10.32%	10.97%	10.88%	22.53%
A2	6.67%	5.33%	10.32%	8.39%	11.41%	21.80%
A3	6.67%	5.33%	18.06%	16.13%	12.37%	26.17%
A4	4.00%	4.00%	8.39%	8.39%	10.36%	16.42%
B	7.33%	7.33%	20.65%	17.42%	17.91%	17.93%
C1	4.67%	4.67%	13.55%	13.55%	12.07%	12.95%
C2	5.33%	5.33%	13.48%	15.48%	15.07%	16.95%
C3	6.00%	6.00%	13.55%	13.55%	12.37%	19.83%
C4	4.00%	4.00%	15.48%	15.48%	13.62%	18.92%
C*	2.67%	2.67%	8.00%	8.00%	11.56%	17.72%
D1	7.33%	7.33%	13.55%	12.90%	19.77%	15.91%
D2	6.00%	6.00%	12.90%	12.90%	12.70%	13.50%

Table 1. Error rate for the four discretization method and three domains

A1-3   Entropy-based cut-point strategy with 1, 2, and 3 cut-points
A4   Entropy-based cut-point strategy with minimal description length principle
B   Chi-Merge
C1,2   LVQ 10 codebook vec. without entropy (1), with entropy (2)
C3,4   LVQ 50 codebook vec. without entropy (3), with entropy (4)
C*   LVQ with entropy and minimal description length principle
D1-2   Histogram-based method without (1) and with (2) entropy criteria

4 Discussion

The row A1 shows the accuracy for the standard C4.5 algorithm. The best accuracy for each domain is marked by the dotted pattern. The results show that the entropy-based cut-point strategy with minimal description length principle can outperform the C4.5 method. This method is able to produce decision trees which have cut-points varying in number on each level of the tree. These cut-points are adjusted to the real condition of the data set. The error rate is always better than the error rate obtained with C4.5 and C4.5 with fixed number multi-interval discretization, see Table 1 line A1-A4. Although, we expected the LVQ algorithm to fit best to our problem of local discretization, the algorithm did not show significant better results. Only, when we combined LVQ-based discretization method with the minimal description length principle, this method showed significant better results than the other ones. However, the method is difficult to handle. The initial number of code book vectors has no significant influence to the result. But the learning rate has a drastically influence. The experiments showed that the results strongly depend on the parameter settings of the LVQ algorithm. The ChiMerge method showed good results for static discretization in the experiment of Kerber [6]. It outperforms other static discretization methods like equal-width intervals and D2. But no comparison to C4.5 was given by Kerber. In our experiment we used ChiMerge in a dynamically fashion. The results show the least accuracy for all three domains.

Although the histogram-based method is a very simple method, we obtained reasonable results. In case of the histogram-based method with entropy criteria, it produces sometimes better results than normal C4.5 algorithm. This shows that our rule for setting the cut point in variant D1, which is recently only finding the middle of two peaks, needs to be improved. The approximation error made by the description of the hull curve of the feature distribution is responsible for the slight shift in the position of the cut-points and it also causes that small peaks are not considered for cut-point selection. In case of the IRIS domain it happens that the decision tree building process stop while the other methods built the tree one level deeper with a partition into 47 data set on the one side and 2 data set on the other side.

Generally, in case of small data sets the entropy-based discretization method works while the LVQ-based and histogram-based method fail.

5 Conclusion

In the paper, we addressed the problem of multi-level discretization during decision tree learning process. Four methods for discretization were described: entropy-based method, ChiMerge, LVQ-based method and histogram-based method. Our aim was to find more accurate and not so time consuming methods. Whereas Fayyad´s entropy-based method combined with minimal description length principle showed good results, the ChiMerge method did not. The good results gained with Fayyad´s

minimal description length principle showed the need for automatic determination of the number of cut points for multi-interval discretization method.

The LVQ-based method in combination with the minimal description length principle showed the best results but it is difficult to handle since the accuracy depends on proper settings of the parameter of the LVQ algorithm.

A simple and fast method is the histogram-based method. This method showed promising results. Further work should be done to improve this method by developing more specific rules for finding the right position of the cut-points.

Acknowledgment

References

1. Breiman, L.; Friedman, J.H.; Olsen, R.A. and Stone, C.J.: „Classification and Regression Trees", Monterey, CA: Wadsworth & Brooks, 1984.
2. Dougherty, J.; Kohavi, R. and Sahamin, M.: „Supervised and Unsupervised Discretization of Continuous Features", Machine Learning, 14th IJCAI, pp. 194-202, 1995.
3. Fayyad, U.M. and Irani, K.B.: „Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning", Machine Learning, 13th IJCAI, vol. 2., Chambery, France, Morgan Kaufmann, pp. 1022-1027, 1993.
4. Haake, K.W.; Perner, P.; Trautzsch, S.; List, P. and Alexander, H.: „Inductive machine learning program in 414 stimulated in-vitro-fertilization (IVF) cycles for the estimation and validation of varying parameters", 11th Annual Meeting of Human Reproduction, vol. 10, Abstract Book 2, Hamburg, 1995.
5. Huber, T.P. and Casler, K.E.: „Initial Analysis of Landset TM data for Elk Habitat Mapping", Intern. Journal of Remote Sensing, vol. 44, pp. 907-912, 1990.
6. Kerber, R.: „ChiMerge: Discretization of Numeric Attributes", Learning: Inductive, AAAI 92, pp. 123-128, 1992.
7. Kohonen, T.: „Self-Organizing Maps", Springer Verlag, 1995.
8. Quinlan, J.R.: „Induction of Decision Trees", Machine Learning 1, pp. 81-106, 1986.
9. Quinlan, J.R.: „C4.5: Programs for Machine Learning", Morgan Kaufmann, Los Altos, California, 1993.
10. Perner, P.; Belikova, T. and Yashunskaja, I.: „Knowledge Acquisition by Symbolic Decision Tree Induction for Interpretation of Digital Images in Radiology", In Proc. Advances in Structural and Syntactical Pattern Recognition, Springer Verlag, pp. 208-219, 1996.
11. Li, Y.K. and Fu, K.S.: „Automatic Classification of Cervical Cell using a Binary Tree Classifier", Pattern Recognition, vol. 16, pp. 69-80, 1983.
12. Wu C.; Landgrebe D. and Swain P.: „The decision tree approach to classification", School Elec. Eng., Purdue Univ., W. Lafayette, IN, Rep. RE-EE 75-17, 1975.
13. Harkonen, A.; Mitika, R. and Moring, I.: „Software tool for developring algorithms for surface inspection systems", in Proc. of SPIE 94, vo. 2248, pp. 41-49, 1994.