A Bayesian Discretizer for Real-Valued Attributes

Xindong Wu

Department of Software Development
Monash University
900 Dandenong Road, Melbourne 3145, Australia

xindong@insect.sd.monash.edu.au

Abstract

Discretization of real-valued attributes into nominal intervals has been an important area for symbolic induction systems because many real world classification tasks involve both symbolic and numerical attributes. Among various supervised and unsupervised discretization methods, the information gain based methods have been widely used and cited. This paper designs a new discretization method, called the Bayesian discretizer, and compares its performance with some other methods including the information gain methods implemented in C4.5 and HCV (Version 2.0). Over the 7 tested datasets, the Bayesian discretizer has the best results for 3 in terms of predictive accuracy.

1 Introduction

In the context of rule induction and decision tree construction, dealing with a continuous attribute means discretization of the numerical attribute into a number of intervals. The discretized intervals can be treated in a similar way to nominal values during induction and deduction. The essential aspect of discretization is to find the right places to set up interval borders. In supervised discretization methods, such as the information gain based methods implemented in C4.5 [Quinlan 93] and HCV (Version 2.0) [Wu 95], the class information of examples in a training set is used. In unsupervised (or class-blind) discretization methods, such as equal-width discretization and equal-frequency discretization [Chiu et al. 91], training examples are grouped into

intervals without taking into account the respective classes of the training examples. Discretization can be performed at induction time (such as in C4.5) or before induction takes place (see [Dougherty *et al.* 95] and [Pfahringer 95]).

Among various discretization methods, the information gain based methods have been widely used and cited ([Dougherty *et al.* 95]). C4.5 provides only binary discretization at induction time based on an information gain approach for real-valued attributes. In HCV (Version 2.0), an information gain based method is the default discretization method for processing numerical attributes before induction takes place. In this paper, we present a new discretization method, called the Bayesian discretizer, which has been implemented in HCV (Version 2.0) as a counterpart to the information gain method, and compare its performance with some other methods including the information gain method also implemented in HCV (Version 2.0). The results of C4.5 with its binary discretization method are also included for comparison although this inclusion might not be very relevant because C4.5 and HCV (Version 2.0) have different induction strategies.

2 The Bayesian Discretizer and Other Discretization Methods

2.1 The simplest class-separating method

The simplest discretization method is to place interval borders between each adjacent pair of examples that are not classified into the same class. Suppose the pair of adjacent values on attribute X are x_1 and x_2, $x = (x_1 + x_2)/2$ can be taken as an interval border.

If the continuous attribute in question is very informative, which means that positive and negative examples take different value intervals on the attribute, this method is very efficient and useful. You can find, for example, that Professors and Lecturers at Australian universities have distinctive salary ranges, and the continuous attribute salary is very informative in distinguishing academic positions. However, this method tends to produce too many intervals on those attributes which are not very informative. These intervals can also easily confuse algorithms like HCV because a 0.1^6 difference between a positive example and a negative one on a numerical attribute makes one more interval.

2.2 Bayesian discretizers

According to Bayes formula,

$$P(c_j|x) = \frac{P(x|c_j)P(c_j)}{\sum_{k=1}^{m} P(x|c_k)P(c_k)} \quad (1)$$

where $P(c_j|x)$ is the probability of an example belonging to class c_j if the example takes value x on the continuous attribute in question, and $P(x|c_j)$ is the probability of the example taking value x on the attribute if it is classified in the class c_j.

$P(c_j)$ can be approximated by using one of the following three probability estimation methods, and $P(c_j|x)$ can take the frequency of c_j under x over all the examples in the training set.

- The frequency method. Given that an event has occurred n times out of N attempts, the simplest method for estimating the probability of e, $p(e)$, is to use its relative frequency, n/N.

- Laplace's Law of Succession. If the data set is representative, the *Laplace's Law of Succession* [Niblett & Bratko 87] uses the following formula rather than the relative frequency to estimate the probability of an event e under the same assumption as the frequency method:

$$p(e) = \frac{n+1}{N+2}. \quad (2)$$

The Laplace's Law of Succession is designed as the default method for probability estimation in HCV (Version 2.0).

- m estimate. The m estimate method [Lavrac & Dzeroski 94] generalizes the Laplace's formula to the following form:

$$p(e) = \frac{n + m \times p_a(+)}{N+m} \quad (3)$$

Given $P(c_j)$ and $P(c_j|x)$, we can construct a probability curve,

$$f_j(x) = P(x|c_j)P(c_j) \quad (4)$$

for each class c_j. When the curves for every class have been constructed, interval borders are placed on each of those points where the leading curves are different on its two sides. Between each pair of those points including the two open ends, -∞ and $+\infty$, the leading curve is the same.

We call a discretization implemented by the above method a Bayesian discretizer.

2.3 The information gain heuristic

When the examples in a training set have taken values of $x_1, ..., x_n$ in ascending order on a continuous attribute, we can use the information gain heuristic adopted in ID3 [Quinlan 86] to find a most informative border to split the value domain of the continuous attribute. [Fayyad & Irani 92] has shown that the maximum information gain by the heuristic is always achieved at a cut point (say, the mid-point) between the values taken by two examples of different classes.

We adopt the information gain heuristic in HCV (Version 2.0) in the following way. Each $x = (x_i + x_{i+1})/2$ $(i = 1, ..., n-1)$ is a possible cut point if x_i and x_{i+1} have been taken by examples of different classes in the training set. Use the information gain heuristic to check each of the possible cut points and find the best split point. Run the same process on the left and right halves of the splitting to split them further. The number of intervals produced this way may be very large if the attribute is not very informative. [Catlett 91] has proposed some criteria to stop the recursive splitting which have been adopted in HCV (Version 2.0):

- Stop if the information gain on all cut points is the same,
- Stop if the number of examples to split is less than a certain number (*e.g.* fourteen in HCV (Version 2.0)), and
- Limit the number of intervals to be produced to a certain number (*e.g.* eight in HCV (Version 2.0)).

In C4.5 [Quinlan 93], the information gain approach is revised in the following ways. Firstly, each of the possible cut points is not the midpoint between the two nearest values, but rather the greatest value in the entire training set that does not exceed the midpoint. This ensures that all border values occur in the training data. Each border value in this case is not necessarily the same as the lower of the two neighbouring values since all training examples are examined for the selection. Secondly, C4.5 adopts the information gain ratio rather than the information gain heuristic. Finally, C4.5 does binarization of continuous attributes, which means only one interval border is found for each continuous attribute at each decision node.

2.4 Other methods

In addition to the methods mentioned above, the HCV (Version 2.0) software has implemented a few unsupervised methods, such as the equal distance

division and the k-nearest neighbours discretization below which will also be used in the experiments in Section 3.

- Equal distance discretization. This method divides the value domain of a real-valued attribute between the smallest value (x_1) and the largest (x_n) into a user specified number (say B) of equally long intervals. Interval borders are placed at $b_i = x_1 + i\frac{x_n - x_1}{B}$ (for all $i \in \{1, 2, \ldots, n-1\}$). B is 5 for the experiments in Section 3.

- k-nearest neighbors discretization (`knn`). Given a specific attribute value, the k-nearest neighbors method tries to estimate which class the value most likely belongs to. `knn` places a border between two values x_i and x_{i+1} if the estimate is different for them. The estimate is based on the assumption that the most probable class is the most common class among the k nearest examples. k is specified by the `-k` switch, with 3 as default.

3 Experiment Results

Table 1 shows accuracy results by HCV (Version 2.0) with different discretization techniques on 7 different data sets, all of which contain noise and continuous attributes. These data sets are all available from the University of California at Irvine Repository of Machine Learning Databases. `Bayes` in Table 1 indicates the Bayesian discretizer designed in Section 2.2. `Itp` refers to the information gain heuristic in HCV (Version 2.0) mentioned in Section 2.3. `Split` refers to the class-separating method in Section 2.1. `Eqdist` and `Knn` refer to the equal distance division and the k-nearest neighbors discretization respectively mentioned in Section 2.4. In addition to HCV (Version 2.0) with these different discretization strategies, we have also chosen C4.5 from the ID3-like algorithms to compete with HCV (Version 2.0) in this section. C4.5 is the most recent successor of ID3-like algorithms. It provides facilities to deal with real-valued and nominal attributes, and the discretization method is based on an information gain heuristic (see Section 2.3).

Apart from the discretization strategy mentioned above for HCV (Version 2.0), the results were produced by using the default parameters of HCV (Version 2.0) and C4.5. The results shown for C4.5 are the pruned ones. The best result for each problem is highlighted with **boldface** font in the table.

Of the 7 different problems, HCV with the Bayesian discretizer (`Bayes`) and C4.5 both get the best results for 3, but `Bayes` performs better than C4.5 on 3 data sets and worse on only 2. Surperisingly, all the unsupervised discretization methods have performed pretty well on these problems. `Split`

Table 1: Accuracy Comparison with Continuous Domains

Domain	Bayes	Itp	C4.5	Split	Eqdist	Knn
Bupa	58.5%	57.6%	**61.0%**	58.5%	50.8%	58.5%
Cleveland 2	72.5%	78.0%	76.9%	72.5%	**82.4%**	72.5%
Cleveland 5	**56.0%**	**56.0%**	**56.0%**	**56.0%**	52.7%	**56.0%**
Crx	**83.0%**	82.5%	80.0%	**83.0%**	81.5%	**83.0%**
LaborNeg	**82.4%**	76.5%	**82.4%**	**82.4%**	76.5%	**82.4%**
Swiss 5	37.5%	28.1%	31.2%	37.5%	**50.0%**	37.5%
Va 2	71.8%	**78.9%**	70.4%	71.8%	76.1%	71.8%

and `Knn` have the same results as `Bayes` does on the 7 data sets. `Eqdist` gets the best results for 2 data sets.

From these experiments, we have found that it is not fair to say that supervised discretization methods generally perform better than unsupervised ones. Also, the information gain heuristic based methods do not seem to perform better than the Bayesian discretizer designed in this paper. Actually, the experiments in this section support the opposite conclusion: the Bayesian method performs better than the information gain based methods implemented in both HCV (Version 2.0) and C4.5 on more example sets.

4 Conclusions

We have designed a new discretization method, the Bayesian discretizer in this paper, which discretizes real-valued attributes before induction takes place. Experiment results have shown that the performance of the Bayesian discretizer is competitive with the information gain based methods in terms of predictive accuracy.

In addition to the discretization methods mentioned in Section 2, there are quite a number of other discretization methods (such as [Holte 93] and [Pfahringer 95]) available in the literature. Since the information gain based methods have been widely used and cited among various supervised and unsupervised discretization methods, we have compared the Bayesian discretizer mainly with the information gain based methods implemented in HCV (Version 2.0) and C4.5. Future research will take more methods especially the newly developed ones on board, and will also address the issues of induction complexity and rule compactness with different discretization methods.

References

[Catlett 91] J. Catlett, On Changing Continuous Attributes into Ordered Discrete Attributes, *Proceedings of the 1991 European Working Session on Learning*, 1991.

[Chiu et al. 91] D. Chiu, A. Wong and B. Cheung, Information Discovery through Hierarchical Maximum Entropy Discretization and Synthesis, *Knowledge Discovery in Databases*, G. Piatetsky-Shapiro and C.J. Matheus (Eds.), MIT Press, 1991.

[Dougherty *et al.* 95] J. Dougherty, R. Kohavi and M. Sahami, Supervised and Unsupervised Discretization of Continuous Features, *Proceedings of the 12th International Conference on Machine Learning*, 194–202.

[Fayyad & Irani 92] U.M. Fayyad and K.B. Irani, On the Handling of Continuous-Valued Attributes in Decision Tree Generation, *Machine Learning*, **8**(1992), 87–102.

[Holte 93] R.C. Holte, Very Simple Classification Rules Perform Well on Most Commonly Used Datasets, *Machine Learning*, **11**(1993), 63–90.

[Lavrac & Dzeroski 94] N. Lavrac and S. Dzeroski, *Inductive Logic Programming - Techniques and Applications*, Ellis Horwood, 1994.

[Niblett & Bratko 87] T. Niblett and I. Bratko, Learning Decision Rules in Noisy Domains, *Research and Development in Expert Systems*, Vol. 3, M.A. Bramer (Ed.), Cambridge University Press, 1987, 25–34.

[Pfahringer 95] B. Pfahringer, Compression-Based Discretization of Continuous Attributes, *Proceedings of the 12th International Conference on Machine Learning*, 456–463.

[Quinlan 86] J.R. Quinlan, Induction of Decision Trees, *Machine Learning*, **1**(1986), 81–106.

[Quinlan 93] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.

[Wu 95] X. Wu, *Knowledge Acquisition from Databases*, Ablex, USA, 1995.