A Bayesian Approach for Supervised Discretization

M. Boullé
France Telecom R&D, France.

Abstract

In supervised machine learning, some algorithms are restricted to discrete data and thus need to discretize continuous attributes. In this paper, we present a new discretization method called MODL, based on a Bayesian approach. The MODL method relies on a model space of discretizations and on a prior distribution defined on this model space. This allows setting up an evaluation criterion of discretization, which is minimal for the most probable discretization given the data, i.e. the Bayes optimal discretization. We compare this approach with the MDL approach and statistical approaches used in other discretization methods, from a theoretical and experimental point of view. Extensive experiments show that the MODL method builds high quality discretizations.
Keywords: supervised learning, data preparation, discretization, bayesianism.

1 Introduction

While real data often comes in mixed format, discrete and continuous, many induction algorithms rely on discrete attributes and need to discretize continuous attributes, i.e. to slice their domain into a finite number of intervals. More generally, using discretization to preprocess continuous attribute often provides many advantages. Discrete values are generally more understandable than continuous values both for users and experts. Many classification algorithms are more accurate and run faster when discretization is used.

Discretization of continuous attributes is a problem that has been studied extensively in the past [6, 7, 9, 12, 16]. For example, decision tree algorithms exploit a discretization method to handle continuous attributes. C4.5 [13] uses the information gain based on Shannon entropy. CART [5] applies the Gini criterion (a measure of the impurity of the intervals). CHAID [10] relies on a discretization method close to ChiMerge [11].

Most discretization methods are divided into top-down and bottom-up methods. Top-down methods start from the initial interval and recursively split it into smaller intervals. Bottom-up methods start from the set of single value intervals and iteratively merge neighboring intervals. Some of these methods require user parameters to modify the behavior of the discretization criterion or to set up a threshold for the stopping rule. In the discretization problem, a compromise must be found between information quality (homogeneous intervals in regard to the attribute to predict) and statistical quality (sufficient sample size in every interval to ensure generalization). The chi-square-based criteria focus on the statistical point of view whereas the entropy-based criteria focus on the information point of view. Other criteria (such as the Gini criterion used in CART) try to find a trade off between information and statistical properties. The Minimum Description Length Principle Cut (MDLPC) criterion [8] is an original approach that attempts to minimize the total quantity of information both contained in the model and in the exceptions to the model.

In this paper, we present a new discretization method called MODL. This method is founded on a Bayesian approach to find the most probable discretization model given the data. We first define a general family of discretization models, and second propose a prior distribution on this model space. This leads to an evaluation criterion of discretizations, whose minimization conducts to the optimal discretization. We use a greedy bottom-up algorithm to optimize this criterion. The method starts the discretization from the elementary single value intervals. It evaluates all merges between adjacent intervals and selects the best one according to MODL criterion. As the discretization problem has been turned into a minimization problem, the method automatically stops merging intervals as soon as the evaluation of the resulting discretization does not decrease anymore. Extensive experiments show that the MODL method produces high quality discretizations that are both compact and accurate.

The remainder of the paper is organized as follows. Section 2 describes the MODL method. Section 3 proceeds with an extensive experimental evaluation.

2 The MODL discretization method

In this section, we present the MODL approach which results in a Bayes optimal evaluation criterion of discretizations and the greedy heuristic used to find a near-optimal discretization.

2.1 Evaluation of a discretization model

The objective of the discretization process is to induce a list of intervals that split the numerical domain of a continuous explanatory attribute. The data sample consists of a set of instances described by pairs of values: the continuous explanatory value and the class value.

Let n be the number of instances in the data sample, and let J be the number of classes. If we sort the instances according to the continuous values, we obtain a string S of class values. On the basis of these notations, we propose the following formal definition of a discretization model.

Definition: A *standard* discretization model is defined by the following properties:

1. the discretization model relies only on the order of the class values in the string S, without using the values of the explanatory attribute,
2. the discretization model allows to split the string S into a list of substrings (the intervals),
3. in each interval, the distribution of the class values is defined by the frequencies of the class values in this interval.

Such a discretization model is called a SDM model.

Notations:

I: number of intervals

n_i: number of instances in the interval i

n_{ij}: number of instances of class j in the interval i

A SDM model is defined by the parameters $\{ I, \ \{n_i\}_{1 \le i \le I}, \ \{n_{ij}\}_{1 \le i \le I, \ 1 \le j \le J} \ \}$.

This definition is very general, and most discretization methods rely on SDM models. They first sort the samples according to the attribute to discretize (property 1) and try to define a set of intervals by partitioning the string of class values (property 2). The evaluation criterion is always based on the frequencies of class values (property 3).

In the Bayesian approach, the best model is found by maximizing the probability $P(Model/Data)$ of the model given the data. Using the Bayes rule and since the probability $P(Data)$ is constant under varying the model, this is equivalent to maximize $P(Model)P(Data/Model)$.

Once a prior distribution of the models is fixed, the Bayesian approach allows to find the optimal model of the data, provided that the calculation of the probabilities $P(Model)$ and $P(Data/Model)$ is feasible. We define below a prior which is essentially a uniform prior at each stage of the hierarchy of the model parameters. We also introduce a strong hypothesis of independence of the distribution of the class values. This hypothesis is often assumed (at least implicitly) by many discretization methods, that try to merge similar intervals and separate intervals with significantly different distributions of class values. This is the case for example with the ChiMerge discretization method [11], which merges two adjacent intervals if their distributions of class values are statistically similar (using the chi-square test of independence).

Definition: The following distribution prior on SDM models is called the three-stage prior:

1. the number of intervals I is uniformly distributed between 1 and n,

2. for a given number of intervals *I*, every division of the string to discretize into *I* intervals is equiprobable,
3. for a given interval, every distribution of class values in the interval is equiprobable,
4. the distributions of the class values in each interval are independent from each other.

We proved the following theorem in [4], on the basis of the exact calculation of the probabilities in the Bayes formula, using all the hypotheses related to the SDM discretization models and the three-stage prior.

Theorem: A SDM model *M* distributed according to the three-stage prior is Bayes optimal for a given set of instances to discretize if the value of the following criterion is minimal on the set of all SDM models:

$$Value(M) = \log(n) + \log\left(C_{n+I-1}^{I-1}\right) + \sum_{i=1}^{I} \log\left(C_{n_i+J-1}^{J-1}\right) + \sum_{i=1}^{I} \log\left(n_i! / n_{i,1}!n_{i,2}!..n_{i,J}!\right). \quad \mathbf{(1)}$$

The first term of the criterion in eqn 1 stands for the choice of the number of intervals and the second term for the choice of the bounds of the intervals. The third term corresponds to the choice of the class distribution in each interval and the last term encodes the probability of the data given the model.

2.2 Search algorithm

```
Optimized greedy bottom-up merge algorithm:
    -   Initialization
          -   Sort the explanatory values: O(n.log(n))
          -   Create an initial interval for each value: O(n)
          -   Compute the initial discretization value: O(n)
          -   Compute the $\Delta$values related to all the possible merges: O(n)
          -   Sort the possible merges: O(n.log(n))
    -   Optimization of the discretization
        Repeat the following steps: at most n steps
          -   Search for the best possible merge: O(1)
          -   Merge and continue if the best merge decreases the
              discretization value
          -   Compute the $\Delta$values of the two intervals adjacent to the
              merge: O(1)
          -   Update the sorted list of merges: O(log(n))
```

Once the optimality of the evaluation criterion is established, the problem is to design an efficient minimization algorithm. The MODL method uses a greedy bottom-up merge algorithm to perform this optimization. It starts with initial single value intervals and then searches for the best merge between adjacent intervals. This merge is performed if the evaluation of the discretization decreases after the merge, and the process is reiterated until no further merge can decrease the criterion.

A straightforward implementation of the algorithm runs in O(n^3) time. In an initialization step, the explanatory attribute values must be sorted: this requires O(n.log(n)) time. Then, the merge process is repeated at most n times. At each merge step, the MODL criterion is computed for every merge of adjacent intervals (at most n intervals), and the best one is used to evaluate the stopping rule. The MODL criterion is based on eqn 1 and requires at most n computing steps to evaluate all the intervals in the discretization resulting from a merge. All in all, the optimization process thus requires O(n^3) time.

However, the method can be optimized in O(*n.log(n)*) time owing to an algorithm similar to that presented in [3]. The MODL criterion can be partly decomposed on the intervals. It consists of a partition cost (first two terms in eqn 1) and of a sum of interval costs (last two terms in eqn 1). Minimizing the value of the discretization after one merge is the same as maximizing the related variation of value Δvalue. Owing to the additivity of the criterion on the intervals, each Δvalue resulting from the merge between two adjacent intervals can be evaluated using the two local intervals, without scanning all the other intervals. The intervals costs can be kept into memory during the optimization process in order to speed up the evaluation of the merges. The merge Δvalues can also be kept in memory and sorted in a maintained sorted list (such as an AVL binary search tree for example). After a merge is completed, the interval costs and the merge Δvalues need to be updated only for the new interval and its adjacent intervals to prepare the next merge step. All in all, the optimized algorithm requires O(n.log(n)) time.

3 Experiments

In our experimental study, we compare the MODL method with the following supervised and unsupervised discretization algorithms:

- MDLPC [8]
- ChiMerge [11]
- ChiSplit [1]
- Equal Frequency
- Equal Width

The MDLPC discretization method is inspired from the MDL approach [14]. It focuses on the selection of a local model restricted to a single interval, comparing two hypotheses: to cut or not to cut the interval. The "not cut" hypothesis requires the encoding of the distribution of the classes plus the effective classes in the interval given their distribution. The "cut" hypothesis requires the encoding of the position of the cut point in addition to the encoding of the two sub-intervals. The MDL principle is used to select the best hypothesis. This optimal split schema is then applied recursively in a greedy top-down algorithm in order to produce multi-interval discretizations.

The main differences between the MODL and the MDLPC methods are the approach (Bayesian versus MDL), the locality of the evaluation criterion (global to the set of intervals versus local to two intervals) and the optimization heuristic (bottom-up versus top-down). For a close examination of the differences between the MDL approach and the Bayesian approach in model selection, see [15] for example.

The ChiMerge and ChiSplit methods exploit exactly the same evaluation criterion. They just differ in the direction of the algorithm: bottom-up merge heuristic versus top-down split heuristic. The criterion is the chi-square test applied to two adjacent intervals to decide whether their class distribution is statistically similar. The ChiMerge method starts from the set of single value intervals and iteratively merges neighboring intervals while they are statistically similar. The ChiSplit method starts from the initial interval containing all the values and recursively splits it into smaller intervals while the intervals can be cut into two statistically different sub-intervals. In the experiment, the significance level is set to 0.95 for the chi-square test threshold.

The Equal Frequency and Equal Width unsupervised discretization methods are evaluated for comparison purposes. The number of intervals is set to 10.

We gathered 15 datasets from U.C. Irvine repository [2], each dataset has at least one continuous attribute and at least a few tens of instances for each class value in order to perform reliable tenfold cross-validations. Table 1 describes the datasets; the last column corresponds to the relative frequency of the majority class.

Table 1: Datasets.

Data Set	Cont. Attr.	Disc. Attr.	Size	Class Nb.	Maj. Acc.
Adult	7	8	48842	2	76.07
Australian	6	8	690	2	55.51
Breast	10	0	699	2	65.52
Crx	6	9	690	2	55.51
German	24	0	1000	2	70.00
Heart	10	3	270	2	55.56
Hepatitis	6	13	155	2	79.35
Hypothyroid	7	18	3163	2	95.23
Ionosphere	34	0	351	2	64.10
Iris	4	0	150	3	33.33
Pima	8	0	768	2	65.10
SickEuthyroid	7	18	3163	2	90.74
Vehicle	18	0	846	4	25.77
Waveform	21	0	5000	3	33.92
Wine	13	0	178	3	39.89

In order to evaluate the intrinsic performance of the discretization methods and eliminate the bias of the choice of a specific induction algorithm, we use the protocol presented in [3], where each discretization method is considered as an elementary inductive method that predicts the local majority class in each learned interval. The discretizations are evaluated for two criteria: accuracy and interval number.

The discretizations are performed on the 181 continuous attributes of the datasets, using a stratified tenfold cross-validation. We have re-implemented the alternative discretization methods in order to eliminate any variance resulting from different cross-validation splits. In order to determine whether the performances are significantly different between the MODL method and the alternative methods, the t-statistics of the difference of the results is computed. Under the null hypothesis, this value has a Student's distribution with 9 degrees of freedom. The confidence level is set to 5% and a two-tailed test is performed to reject the null hypothesis.

The whole result tables are too large to be printed in this paper. The results are summarized in table 2, which reports the mean of the accuracy and interval number per attribute discretization and the number of significant MODL wins and losses. Except for the ChiMerge method, the supervised discretization methods perform significantly better than the unsupervised methods. The accuracy results allow clustering the methods in three groups. The leading group consisting of the MODL and ChiSplit methods is followed by the intermediate group restricted to the MDLPC method and then by the last group containing the ChiMerge method and the unsupervised methods. The average accuracy difference between the leading group and the MDLPC method is larger than the difference between the MDLPC method and the unsupervised Equal Frequency Method. Although the MODL criterion is Bayes optimal assuming the three-stage prior, its search algorithm exploits a greedy heuristic that may fall in a local optimum. This explains why the ChiSplit method obtains slightly better accuracy results than the MODL method.

Table 2: Discretization results.

	Test Accuracy		Interval Number	
	Mean	MODL wins	Mean	MODL wins
MODL	68.5		3.8	
MDLPC	68.0	14/6	3.2	9/46
ChiMerge	67.5	32/9	66.1	160/0
ChiSplit	68.6	5/11	7.2	148/1
Equal Frequency	67.7	33/16	7.3	164/7
Equal Width	67.1	40/12	8.5	173/4

In order to analyze the differences of accuracy for the 181 attributes with more details, figure 1 shows the repartition function of the differences of accuracy between the MODL methods and the other discretization methods.

On the left of the figure, the MODL method is dominated by the other methods and, on the right, it outperforms the other algorithms. For about 40% of the attributes (between x-coordinates 20 and 60), all the discretization methods obtain equivalent results. Compared to the MDLPC method, the MODL method is between 0 and 3% less accurate in about 10% of the discretizations, but is between 3 and 10% more accurate in about 10% of the discretizations. The average difference of 0.5% is thus significant and reflects potential large differences of accuracy on individual attributes.

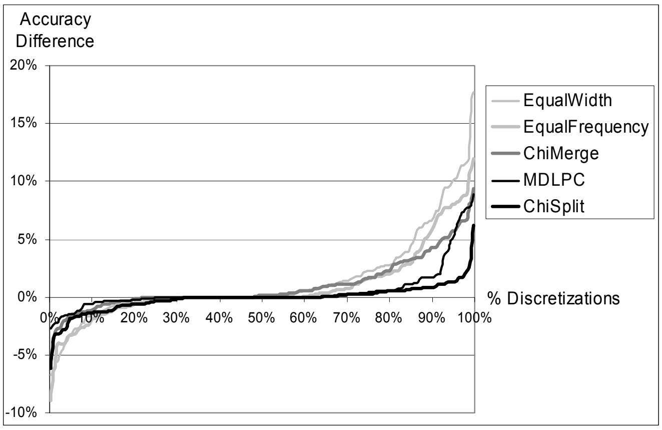

Figure 1. Differences of accuracy of the discretizations.

In order to analyze both the accuracy and interval number results, we reported the mean results on a two-criteria plan in figure 2, with the accuracy on the x-coordinate and the interval number on the y-coordinate.

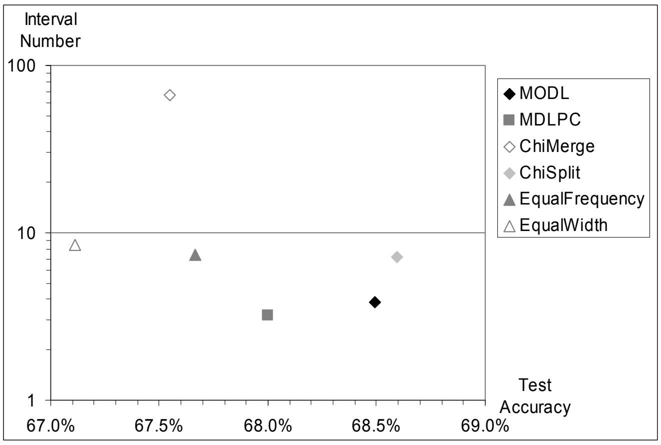

Figure 2. Bi-criteria evaluation of the discretization methods for the test accuracy and the interval number.

In bi-criteria analysis, a solution *dominates* (or is *non-inferior* to) another one if it is better for both criteria. A solution that cannot be dominated is *Pareto optimal*: any improvement on one of the criteria causes deterioration on another criterion. The *Pareto curve* is the set of all the Pareto optimal solutions. The three methods MDLPC, MODL and ChiSplit are Pareto optimal and clearly dominate the other methods. The MDLPC method builds discretizations with small interval numbers, but is outperformed by the two most accurate MODL and ChiSplit methods. The ChiSplit method produces accurate discretizations at

the expense of twice the number of intervals obtained by the MODL and MDLPC methods. The MODL method produces discretizations that are both accurate and compact.
All the supervised discretization methods run in super-linear time. Since the interval splits are searched only on the boundary points (far less numerous on average than the instance values), the discretization time is dominated by the initialization step where the instances are sorted according to the explanatory values. Hence, discretization speed was not a discriminating criterion for the tested methods.

Conclusion

The MODL method is a direct application of Bayesianism for the problem of model selection in the discretization field. For a given space of discretization models and a given prior distribution on this model space, the MODL evaluation criterion allows finding the Bayes optimal discretization. The closest discretization method is the MDLPC method. We have pointed out the differences between the MODL and the MDLPC methods from a theoretical point of view. The main differences come from the model selection approach, the locality of the evaluation criterion and the optimization heuristic. Comparative experiments with several supervised and unsupervised discretization methods show that the MODL method produces accurate and compact discretizations.

References

[1] Bertier, P. & Bouroche, J.M., *Analyse des données multidimensionnelles*. Presses Universitaires de France, 1981.

[2] Blake, C.L. & Merz, C.J., UCI Repository of machine learning databases. Web URL http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[3] Boullé, M., Khiops: a Statistical Discretization Method of Continuous Attributes. *Machine Learning*, 55:1, pp. 53-69, 2004.

[4] Boullé, M., MODL, une méthode quasi-optimale de discrétisation supervisée. Note technique NT/FTR&D/8444. France Telecom R&D, 2004

[5] Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J., *Classification and Regression Trees*. California: Wadsworth International, 1984.

[6] Catlett, J., On Changing Continuous Attributes into ordered discrete Attributes. *Proceedings of the European Working Session on Learning*. Springer-Verlag, pp. 87-102, 1991.

[7] Dougherty, J., Kohavi, R. & Sahami, M., Supervised and Unsupervised Discretization of Continuous Features. *Proceedings of the Twelf International Conference on Machine Learning*. Los Altos, CA: Morgan Kaufmann, pp. 194-202, 1995.

[8] Fayyad, U. & Irani, K., On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8, pp. 87-102, 1992.

[9] Holte, R.C., Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11, pp. 63-90, 1993.

[10] Kass, G.V., An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29:2, pp. 119-127, 1980.

[11] Kerber, R., Chimerge discretization of numeric attributes. *Proceedings of the 10th International Conference on Artificial Intelligence*, pp. 123-128, 1991.

[12] Liu, H., Hussain, F., Tan, C.L. & Dash, M., Discretization: An Enabling Technique. *Data Mining and Knowledge Discovery*, 6:4, pp. 393-423, 2002.

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

[14] Rissanen, J., Modeling by shortest data description. *Automatica*, 14, pp. 465-471, 1978.

[15] Vitanyi, P.M.B. & Li, M., Minimum Description Length Induction, Bayesianism, and Kolmogorov Complexity. *IEEE Trans. Inform. Theory*, IT-46:2, pp. 446-464, 2000.

[16] Zighed, D.A. & Rakotomalala, R., *Graphes d'induction*. HERMES Science Publications, pp. 327-359, 2000.