Optimal multiple intervals discretization of continuous attributes for supervised learning

D.A. Zighed R. Rakotomalala F. Feschet
ERIC Laboratory - University of Lyon 2
5, av Pierre Mendès-France
69676 BRON CEDEX FRANCE
e-mail: {zighed,rakotoma,ffeschet}@univ-lyon2.fr

Abstract

In this paper, we propose an extension of Fischer's algorithm to compute the optimal discretization of a continuous variable in the context of supervised learning. Our algorithm is extremely performant since its only depends on the number of runs and not directly on the number of points of the sample data set. We propose an empirical comparison between the optimal algorithm and two hill climbing heuristics.

Introduction

Rule induction from examples, such as the well known induction trees (Breiman *et al.* 1984), usually use categorial variables. Hence, to manipulate continuous variables, it is necessary to transform them to be compatible with the learning strategy. The processus of splitting the continuous domain of an attribute into a set of disjoints intervals is called discretization. In this paper, we focus on supervised learning where we take into account a class $Y(.)$ to predict.

Lechevallier (Lechevallier 1990) has described an approach, based on Fischer's works (Fischer 1958), to determine the optimal partition in K intervals among all the ordered partitions in $O(n^2)$. Thus, we can consider the discretization problem to be algorithmically solved since we can fastly compute the optimal discretization with Lechevallier's algorithm. However, the found solution is generaly specific to a finite learning set, so that another sample set on the same problem can lead to a different optimal discretization. Hence, in the context of supervised learning, the quality of the discretization must be measured by the quality of the prediction it implies on a test set. In this case, we wonder whether or not the optimal discretization performs really better than hill-climbing heuristics such as Fusinter (Zighed, Rakotomalala & Rabaseda 1996) or MDLPC (Fayyad & Irani 1993) whose complexities are lower.

In the next section, we present a formulation of the problem of discretization, then we describe an extension of Lechevallier's algorithm to find the optimal discretisation and we insist on the use of runs instead of the points of the sample data set. After, we introduce two hill-climbing strategies. Finaly, we present experiments and empirical studies of the performances of the various presented stategies.

Discretization

Formulation

Let be D_X the domain of definition of a continuous attribute $X(.)$. The discretization of $X(.)$ consist in splitting D_X into k intervals I_j, $j = 1, \ldots, k$ with $k \geq 1$. We note $I_j = [d_{j-1}, d_j[$ with d_j's be the discretization points.

Border points

Let be $X(\Omega) = \{x_1, \ldots, x_j, x_{j+1}, \ldots, x_a\}$ the ordered set of the values of $X(.)$ over the set Ω, $x_1 < \ldots < x_a$. Let us denote by Ω_j the set of the examples whose image by $X(.)$ is x_j. Assume d_j is situated between x_j and x_{j+1}, such that $d_j = \rho \times x_j + (1 - \rho) \times x_{j+1}$ $(0 \leq \rho \leq 1)$. d_j is called a border point if and only if the classes of the elements of Ω_j are not all the same than those of the elements of Ω_{j+1}.

U is the set of border points and we have $u = Card(U)$. Fayyad and Irani (Fayyad & Irani 1993) have proved that the discretization points d_j can only be border points. Thus U is the set of possible points for discretization. Finding the optimal discretization is then equivalent to extract the subset U^* $(U^* \subseteq U)$ which induces an optimal split for the used criterion.

Runs

A run is a set of points placed between two border points. A run is represented by a vector which describe, for each class, its number of observations (the number of points of the run which belong to this class). We

[1] Copyright © 1997, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

can then represent the sample set by an array T as the following:

$$T = (R_1|R_2|\dots|R_r)$$

It is clear that the number of runs is equal to the number of border points plus one: $r = u + 1$.

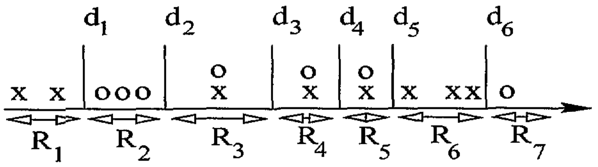

Figure 1: The runs R_i and the border points d_j for a sample set composed of two classes "x" and "o". The sequences can be homogeneous: (R_1, R_2, R_6, R_7) or not (R_3, R_4, R_5). In the first case, all points have the same value for $Y(.)$ and in the last case, all points have the same value for $X(.)$.

Optimal discretization

Measuring the quality of a partition

The problem consists in finding the split from which we can predict the class $Y(.)$ at best. Every subset $U^i \subseteq U$ of border points leads to a partition perfectly described by an array T^i whose structure is similar to T.

We necessarily have to compare partitions containing different numbers of intervals. The quality measure, we shall use, must take into account the increase in complexity induces by an excessive partitionning. There are several ways to introduce complexity bias to avoid excessive partitionning. We can quote measures based on Minimum description length principle (Fayyad & Irani 1993), measures of the type χ^2 (Tschuprow, Cramer), or measures using informational gain taking into account sample size (Zighed, Rakotomalala & Rabaseda 1996). These measures can possibly be guided by resubstitution error rate (Liu & Setiono 1995).

From now on, we use Zighed's measure denoted $\varphi(.)$. We have carried out several experiments which conclude that these different measures have the same behaviour. Our goal is to find among all the arrays T^* which one verifies: $\varphi(T^*) = \min_i[\varphi(T^i)]$.

An algorithm for finding optimal discretization

Finding optimal discretization in k classes with a set of n points could be done by testing all the possible partitions. In this case, the algorithm has a very high complexity $O(n^{k-1})$. But, Lechevallier (Lechevallier 1990) has proposed an algorithm based on Fischer's works (Fischer 1958) for finding optimal discretization, whose complexity is $O(n^2)$.

This algorithm use two fundamental hypothesis which are:

- **ordering elements property** : over the set $X(\Omega) = \{x_1, ..., x_n\}$ a partition in k intervals is ordered if and only if for every two elements x_i and x_j of $X(\Omega)$ which belong to the same interval I_l, every element situated between x_i and x_j belongs to the same interval;

- **Additivity of the quality measure:** if a partition $(\{x_1, ..., x_i\}, I_2, ..., I_k)$ in k intervals is optimal, then the partition $(I_2, ..., I_k)$ is an optimal partition in $k-1$ intervals of the set $\{x_{i+1}, ..., x_n\}$.

The first property is not restrictive since $X(\Omega) \subseteq \mathbb{R}$ the element are necessarily ordered. But, the second property requires the additivity of the choozen measure. It has been proved by Lechevallier (Lechevallier 1990) for the measure based on a χ^2, and by Zighed & al. (Zighed, Rakotomalala & Rabaseda 1996) for the previous $\varphi(.)$ measure.

An extension of Fischer's algorithm

Fischer's algorithm is a dynamic programming procedure. The main idea is to find some relations between the optimal partition in k intervals of the inital data set and the optimal partitions in $k-1$ intervals of subsets of the data set. It uses the order to restrict the number of possible partitions. The additivity of the φ measure is then used to obtain a recurrent equation between optimal partitions. *We present here an extention of Fischer's and Lechevallier's algorithms by considering the partitionning of a set of runs instead of points. This is a consequence of the work of* (Fayyad & Irani 1993) *who have proved than a run can never be split in an optimal discretization.*

Let us consider a set of runs $\{R_i, 1 \leq i \leq r\}$. We search an ordered partition which is optimal for the φ measure. We denote by $\mathcal{P}_k^1$ this partition with k the number of intervals and 1 the first run taken into account. We then have:

$$\mathcal{P}_k^1 = \{\{R_1, \dots, R_{j_1}\}, \dots, \{R_{j_{k-1}+1}, \dots, R_r\}\}$$

Since φ is additive, the value of φ on the previous partition is given by $\sum_{i=1}^{k} \varphi\left(\{R_{j_{i-1}+1}, \dots, R_{j_i}\}\right)$ where $j_0 = 0$ and $j_k = r$ for the sake of simplicity. This additivity of φ implies that

$$\mathcal{P}_{k-1}^{j_1+1} = \{\{R_{j_1+1}, \dots, R_{j_2}\}, \dots, \{R_{j_{k-1}+1}, \dots, R_r\}\}$$

is an optimal partition in $k-1$ intervals of the set of runs $\{R_i, j_1+1 \leq i \leq r\}$. Hence, there are connections between the optimal partition in k intervals and those in $k-1$ intervals. The problem is then to find the first cutting point j_1. This point is one of the integers interval $[1, r-k+1]$. The optimal partition $\mathcal{P}_k^1$ can then be obtained through a minimization process:

$$\varphi\left(\mathcal{P}_k^1\right) = \min_{1 \leq j_1 \leq r-k+1} \left\{\varphi\left(\{R_1, \ldots, R_{j_1}\}\right) + \varphi\left(\mathcal{P}_{k-1}^{j_1+1}\right)\right\}$$

The previous relation introduce a relation of recurrence between $\mathcal{P}_k^1$ and $\mathcal{P}_{k-1}^{j_1+1}$. If we can compute the various partitions $\mathcal{P}_{k-1}^{j_1+1}$, then by using the minimization procedure we deduce $\mathcal{P}_k^1$. To compute the partitions $\mathcal{P}_{k-1}^{j_1+1}$, it is possible to use again the relation of recurrence.

We obtain the following algorithm:

1. Computing the partitions $\mathcal{P}_1^l$ for $1 \leq l \leq r$

2. For all p, $2 \leq p \leq k$, compute the partitions $\mathcal{P}_p^q$ for each q of the interval $[1, r-p+1]$
   - Compute the summations $\varphi\left(\{R_q, \ldots, R_o\}\right) + \varphi\left(\mathcal{P}_{p-1}^o\right)$ for $q \leq o \leq r$
   - $\varphi\left(\mathcal{P}_p^q\right)$ is the minimum value of the previous ones
   - At step k of this algorithm, the optimal partition $\mathcal{P}_k^1$ is determined when $q = 1$.

3. The partition $\mathcal{P}^*$ which is optimal among all the previous optimal partition, is given by: $\varphi\left(\mathcal{P}^*\right) = \min_{j=1}^{r} \varphi\left(\mathcal{P}_j^1\right)$

Two hill-climbing heuristics

Bottom-Up (BU) and Top-Down (TD) strategies

Beside Fischer's strategy whose complexity is $O(r^2)$, it is possible to use less complexity $[O(r)]$ methods but which are not optimal. They are used in most of the contextual discretization algorithms published in the litterature. These methods are based upon two hill-climbing heuristics:

- the first one, called "top-down", uses the "divide and conquer" principle. It recursively computes a binary partitionning of each previously computed sets until a stopping rule is verified (Catlett 1991). The set U^* is iteratively built by adding discretization points.

- the second one, called "bottom-up" uses an opposite principle. Its starts from an initial partition defined by U, the set of border points. Then, it iteratively tries to aggregate adjacents intervals until the partition optimizes the measure (Zighed, Rakotomalala & Rabaseda 1996) or until no aggregation is reliable (Kerber 1991). In the last case, the set U^* is built by deleting points of the partition.

Theses two strategies run very fast but they have the disadvantage of being irrevocable. Each added point in U^* with the "top-down" strategy cannot be deleted; each deleted point in U^* cannot be reintroduced in the last strategy.

A previous studies (Zighed, Rakotomalala & Rabaseda 1996) have showed that MDLPC (Fayyad & Irani 1993) and Fusinter are very close, so we only use this last algorithm here.

Is an optimal discretization algorithm usefull or useless ?

We are now confronted to a simple choice: on one side we have a very fast algorithm, on the other side an algorithm, with a higher cost, but which provides a global optimization. Is it interesting to use one of these instead of the other one ?

In the context of supervised learning, one of our main goals is to build a model having the minimum error rate in prediction, which could be estimated by applying the model on a sample set not used for learning, called test set. It is generally supposed that a model which optimizes a criterion having good properties, especially the resistance to overfitting on noisy data, will perform better in prediction. Hence, the problem of learning is often reduced to an optimization problem. In this paper, we verify this hypothesis by confronting the hill-climbing heuristics with our improvement of Fischer's algorithm.

Experiments

Comparison method

We compare the Fusinter method with Fischer's strategy using the Breiman's waves dataset (Breiman *et al.* 1984). To do so, we have generated 11 learning samples of 300 points each and a test sample of 5000 points. For any ω taken from the learning sample and the test sample, we dispose of a 21 components vectors noted $(X_1(\omega), \ldots, X_j(\omega), \ldots, X_{21}(\omega))$ and of a label $Y(\omega)$. For each attribute X_j, we determine the best discretization obtained on the learning sample and we consider it like a decision tree with one depth level. Then, we measure the quality of the discretization on the test sample by the accuracy rate.

The two methods (Fusinter, Fischer) are compared using a t-test for dependent samples. Critical value of the test is $t_{0.975} = 1.96$ for a 5% significance level, and we found $t^* = 1.735$. So, we conclude that Fischer's strategy is not significantly better than Fusinter.

Accuracy rate	**Trials**	**Mean**	**Std**
FISCHER	231	0.4825	0.069
FUSINTER	231	0.4807	0.070

Table 1: Comparison Fusinter vs Fischer

Results and discussion

Three main results draw our attention:

- in our experiments, Fusinter almost always found the right number of intervals;
- but nearly never find the optimal partition (29 times over 231 trials);
- this disadvantage does not significantly modify its performance towards those of Fischer's strategy if we consider the error rate in prediction. Indeed, over the 231 files, Fusinter is better than Fischer 73 times and has similar performances 47 times. Using the test procedure described above, the difference is not significant for a 5% risk (table 1).

The doubts of several authors (Breiman *et al.* 1984) on the usefullness of optimization in induction process are confirmed in this paper. Our goal is to obtain the lowest error rate in prediction with the simplest model following Occam's razor principle. Then, it is probably not very interesting to use complex learning strategies. We can get better results (in our experiments, for a 10% risk, we can conclude to the superiority of Fischer's strategy) but they are not significant. Hence, the choice of a method is more dependent on the faculty of understanding the model, on its simplicity or its running time.

Moreover, we wonder whether an optimization procedure, which only uses the contingency tables information, is reliable. In fact, in this case, we neglect the distribution of the samples. Let us consider a sample belonging to a class Y_1 which is surrounded by elements of a class Y_2, then we can suppose that this point has the wrong label or that this point is aberrant. There are several solutions to this problem. It is possible to mix supervised and unsupervised methods (Dougherty, Kohavi & Sahmi 1995) by introducing, for instance, a measure which takes into account the relative distribution of the intervals in $\mathbb{R}$, by using the inertia (de Merckt 1993) or the variance (Lechevallier 1990).

Conclusion

In this paper, we have established that the use of optimal discretization using a partition quality measure has no significant improvement on the error rate in prediction beside a simple hill-climbing heuristic.

Nevertheless, we have to qualify this conclusion. Some works have proved that the loose of informations introduced by discretization can hide the relations between the variables (Celeux & Robert 1993). Thus, it would interesting to complete this study by trying to chacterize the problems and the data (distribution, noise level...) for which it is necessary to use optimal discretization. It would be also interesting to study the behaviour of different induction processes in relation with the various discretization methods.

References

Breiman, L.; Friedman, J.; Olshen, R.; and Stone, C. 1984. *Classification and Regression Trees.* California: Wadsworth International.

Catlett, J. 1991. On changing continuous attributes into ordered discrete attributes. *Artificial Intelligence Journal* 164–178.

Celeux, G., and Robert, G. 1993. Une histoire de discretisation (avec commentaires). *La Revue du Modulad* (11):7–43.

de Merckt, T. V. 1993. Decision trees in numerical attributes spaces. In *Proceedings of the 13th IJCAI.*

Dougherty, J.; Kohavi, R.; and Sahmi, M. 1995. Supervised and unsupervised discretization of continuous features. In Preiditis, A., and Russel, S., eds., *Proceedings of the Twelth International Conference in Machine Learning.*

Fayyad, U., and Irani, K. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th IJCAI*, 1022–1027.

Fischer, W. 1958. On grouping for maximum homogeneity. *Journal of American Statistical Association* (53):789–798.

Kerber, R. 1991. Chimerge discretization of numeric attributes. In *Proceedings of the 10th International Conference on Artificial Intelligence*, 123–128.

Lechevallier, Y. 1990. Recherche d'une partition optimale sous contrainte d'ordre total. Technical report, INRIA.

Liu, H., and Setiono, R. 1995. Discretization of ordinal attributes and feature selection. Technical Report TRB4/95, Department of Sys. and Comp. Sci, National University of Singapore.

Zighed, D.; Rakotomalala, R.; and Rabaseda, S. 1996. A discretization method of continuous attributes in induction graphs. In *Proceedings of the 30th European Meetings on Cybernetic and Systems Research*, 997–1002.