Discretization of Continuous Features by Resampling

Taimur Qureshi*, D.A.Zighed*

*University of Lyon 2 - Lab ERIC
5, Avenue Pierre Mendès France, 69676 Bron Cedex - France
taimur.qureshi, abdelkader.zighed@univ-lyon2.fr

Résumé. Les arbres de décision sont largement utilisés pour générer des classificateurs à partir d'un ensemble de données. Le processus de construction est une partitionnement récursif de l'ensemble d'apprentissage. Dans ce contexte, les attributs continus son discrétisés. Il s'agit alors, pour chaque variable à discrétiser de trouver l'ensemble des points de coupure. Dans ce papier nous montrons que la recherche des ces points de coupure par une méthode de ré-échantillonnage, comme le BOOTSTRAP conduit à des meilleurs résultats. Nous avons testé cette approche avec les méthodes principales de discrétisation comme MDLPC, FUSBIN, FUSINTER, CONTRAST, Chi-Merge et les résultats sont systématiquement meilleurs en utilisant le bootstrap. Nous exposons ces principaux résultats et ouvrons de nouvelles pistes pour la construction d'arbres de décision.

1 Introduction

In the process of knowledge discovery from a raw data set, we first preprocess the data to remove noise and handle missing data fields. Then data transformation, such as the reduction of the number of variables and the *discretization of attributes* defined on a continuous domain, is often performed, which is later provided to a data mining algorithm. One of the most important and complex issues in data mining is related to the transformation process such as discretization which consists of converting numerical data into symbolic or discrete form. Kusiak [9] emphasized that the quality of knowledge discovery from a data set can be enhanced by discretization because many of the knowledge discovery techniques are very sensitive to size of data in terms of complexity. Thus, the choice of discretization technique has important consequences on the induction model used such as CART [2].

In addition, numerical value ranges are not easy enough for evaluation functions to handle in a nominal domain ; for example, the original versions of the popular machine learning algorithms ID3 could be used only for categorical data and Quinlan [11] had to transform continuous ones into discrete values in his C4.5 decision tree learner. Many real-world classification algorithms are hard to solve unless the continuous attributes are discretized. It is hard to determine the intervals for a discretization of numerical attributes that has an infinite number of candidates. A simple discretization procedure divides the range of a continuous variable into equal-width intervals or equal-frequency intervals. Fayyad et al. [6] suggested a class dependent algorithm which reduce the number of attributed values maintaining the relationship between the class and attribute values. Liu et al. [10] classified discretization methods from

five different viewpoints : supervised vs. unsupervised, static vs dynamic, global vs local, top-down vs bottom-up, and direct vs incremental. Unsupervised methods do not make use of class information in the discretization process while supervised methods utilize it. If no class information is available, unsupervised discretization is the only method possible. Dynamic methods perform discretization of continuous values during classification process, while static methods preprocess discretization before classification process. Local methods use the local region of the instance space while global methods use the entire space. Top-down methods as FUSBIN, MDLPC and CONTRAST [5-7] start with one interval and split intervals in the process of discretization and are based mostly on binarization within a subset of training data. While, bottom-up methods like FUSINTER [5] and Chi-Merge [4] split completely all the continuous values of the attribute and merge intervals in the process of discretization. In this article, we focus on these two types of strategies in determining better discretization points and providing comparisons in terms of quality and prediction rates [1].

Our goal is find a way to produce better discretization points. Previously, various studies have been done to estimate the discretization points from samples. Significantly, in [1], a set of learning samples are used to approximate the best discretization points of the whole population, but also argue that the learning sample is an approximation of the whole population, so the optimal solution built on a single sample set is not necessarily the global one. This interpretation leads us to use a resampling approach [3] to determine better distributions of the discretization points, where each point has a probability to be the exact discretization point towards the whole population. By doing so, we attempt to improve on the quality of discretization and better estimation of the discretization points of the entire population, thus, treating the discretization problem in the statistical area with new results. In this paper, we show that by performing resampling using bootstrap [8] we determine a better estimate of discretization point distribution over the entire population, which is shown improving the prediction rate of the achieved discretization. Moreover, we further improve on the quality and mean prediction rate obtained from resampling by applying a discretization point selection protocol. This protocol selects the cut points according to some criteria (e.g. entropy) from the resampling bootstrap frequency point distribution obtained from resampling n times and improves further on the prediction rate. Furthermore, we compare the prediction rates of different top-down and bottom-up strategies by using resampling. In section 2, we lay out the framework for discretization and define the data sets used in our calculations. In 3, we give an illustration of our work and results by applying the methodology to an example data set and then to a much more detailed, Breiman's wave dataset [2]. We also compare several top-down and bottom-up strategy based criteria as in [1], such as Chi-merge based on χ^2 Statistical Law, FUSBIN and FUSINTER based on the uncertainty principle, MDLPC based on information gain and CONTRAST that takes into account the homogeneity of the classes and also the point density. In the end we conclude with observations, deductions and proposals for future work.

2 Definitions and Notations

Framework and Formulation : Let $X(.)$ be an attribute value on the real line $\Re$. For each example ω of a learning set Ω, $X(\omega)$ is the value taken by the attribute $X(.)$ at ω. The attribute $C(.)$ is called the endogenous variable or class and is usually symbolic and if an example belongs to a class c, we have $C(\omega) = c$. We also suppose that $C(\omega)$ is known for all ω of

the learning sample set Ω. Thus, we try to build a model, denoted by Φ, such that ideally we have : $C(.) = \Phi(X_1(.), ..., X_p(.)))$. The discretization of $X(.)$ consists in splitting the domain D_x of continuous attribute $X(.)$, into k intervals $I_j, j = 1, ..., k$, with $k \geq 1$. We denote $I_j = [d_{j-1}, d_j]$ with the $d'_j s$ called the discretization points which, are determined by taking into account the particular attribute $C(.)$.

Prediction Rate : We measure the quality of discretization by taking into account the prediction rate, which is calculated as follows :

$$\tau_j = \frac{card\{\omega \in \Omega_t / \hat{C}(\omega) = C(\omega)\}}{card\{\Omega_t\}}$$

We denote by τ_{js} the good prediction rate resulting from the discretization of X_j obtained by applying the method q on the sample Ω_s or τ_{jt} by applying on the test sample Ω_t.

Data Set : In this article, we use two different data sets. First, we use a small data set of 110 individuals corresponding to a two-class problem shown in figure 1. The second large data set used for comparisons and results is the Breiman's waveform dataset [2] having 4590 individuals and 21 attributes $X(.)$, that correspond to a three class problem.

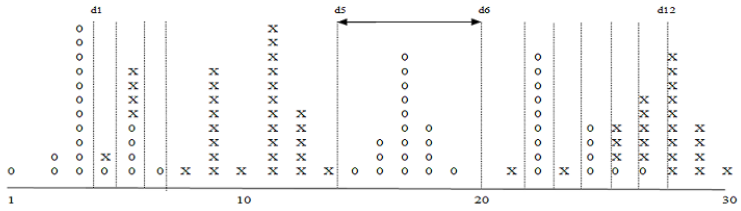

FIG. 1 – *Runs R_i and boundary points d_j for a sample of 2 classes "x" and "o".*

3 Results and Comparisons

3.1 Illustration using Example Data of Figure 1

Consider a data set of figure 1 of 110 individuals having two classes. We perform FUSBIN discretization with $\lambda = 0.91$ on each random and bootstrap sample of size 30 and generate 500 samples. Figure 2 gives us the discretization point distribution from 500 bootstrap and random samples. We see that the discretization achieved from bootstrap is seem to be a little more generalized and well defined over the four small intervals ; 4.5 to 6, 6.5 to 9, 12.5 to 14.5 and 22.5 to 27. While, in random sampling, the point distribution does seem to be poorly defined in a large region of values from 18 to 27. We further argue that this difference increases as the data set becomes larger which we shall see in with Breiman's data set. We also calculated the mean prediction rate i.e. $\mu_v = \frac{1}{100} \sum_{j=1}^{100} \tau_j^v$ by estimating the mean values of each of the above 500 samples. We found the rates of bootstrap and random sampling as 22 and 21.1 respectively showing that the bootstrap samples performed better, with this difference further increasing with added complexity and size of the population as shown in the next subsection.

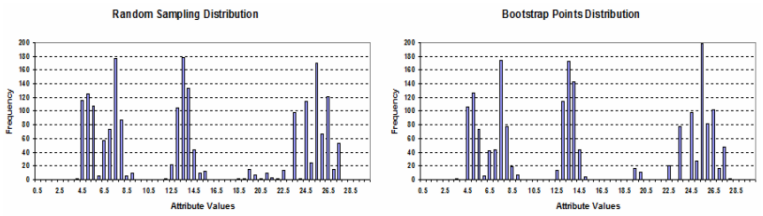

FIG. 2 – *Discretization point distribution from 500 random and bootstrap samples.*

Next, we improve the quality and prediction rate by introducing a notion of discretization point selection protocol. This protocol selects the discretization points from a given point frequency distribution, having higher probability of occurrence, and splits on those points if a certain criterion (e.g. entropy) is met. To illustrate, from figure 2 we see that the highest probable point is 25.5 ; so we take that point and split the population if a certain criterion (FUSBIN entropy) is met. We continue our process on the obtained splits in a top-down manner, until the criterion allows further splitting or all the points from the frequency distribution have been already chosen. We applied this protocol on both the bootstrap and random samples and it selected 6 out of 30 and 8 out of 36 discretization points from both the frequency point distributions respectively. We calculated the prediction rate as 22 for bootstrap and 19.5 for random sampling, demonstrating the better quality of discretization achieved by selection from bootstrap. We further argue that sampling gives us a lot of variation in prediction rates i.e. for bootstrap samples the prediction rate varies from 17 to 26 and thus, it is difficult to obtain a generalized estimate of the discretization points of original population. Here, our protocol achieves well defined discretization points and thus, give better estimate of the original discretization points.

3.2 Analysis and Results using Breiman's Waveform Data

For this section we use the Breiman's waves data set. We generated 100 bootstrap and random samples Ω_b and Ω_s ; s = 1,...,100 of 300 points each and Ω_t a test sample of 4590 points. For any ω taken from the sample, we have a vector of 21 components denoted as $(X_1(\omega), ..., X_21(\omega))$ and a label $C(\omega)$. We repeated the process described above with the waveform data set. We took each variable from the data set and generated bootstrap and random samples as above. Then, we performed FUSBIN on both the 100 bootstrap and random samples and obtained mean prediction rates of 196 and 180 respectively, showing a better performance with bootstrap sampling. Then, we applied our discretization point selection protocol on the point distribution obtained and selected the best points (using FUSBIN criterion) from both sampling methods. We found a prediction rate of 309 for points obtained from bootstrap distribution and a lesser value of 271 for random sampling showing a significant amount of improvement in the prediction rate by using resampling (or bootstrapping).

Finally we compare FUSINTER, FUSBIN, CONTRAST, MDLPC and Chi-Merge by resampling. This is done in two by two according to the following procedure ; Let u and v be the two methods to compare. First, we obtain discretization points from 100 bootstrap samples and create a frequency point distribution for each variable. Then, using our selection protocol, we

Diff in Mean P-Rate	MDLPC	ChiMerge	CONTRAST	FUSBIN	FUSINTER
MDLPC	X	52.7	10.6	6.9	7.3
ChiMerge		X	-42.1	-45	-45.4
CONTRAST			X	-3.7	-3.3
FUSBIN				X	0.4
FUSINTER					X

TAB. 1 – *Computed Results : Difference in Mean Prediction-Rate* μ_{uv}

select discretization points from those point distribution frequencies, by applying the criterion of the respective method (from which the initial discretization points were obtained). We then compute prediction rates $\tau_j t$ of the selected discretization points from each method in relation to the whole test sample Ω_t. We form the difference Γ_{uv} of the two prediction rates obtained and conclude that u is better than v if Γ_{uv} is significantly superior to 0. Table 1 presents the comparison in terms of the difference of the means μ_{uv} of prediction rates from all the variables. Positive values of μ_{uv} indicate that the method in the row is better than the method in the column. Aside from Chi-Merge method whose results are relatively poor, all the other methods have relatively smaller differences. However, among those methods MDLPC seemed to be the best with a much lesser time complexity. FUSBIN and FUSINTER also had a smaller time complexity in comparison to CONTRAST which had a quadratic complexity which had to be taken into account when the number of examples becomes too high.

4 Conclusion

The learning sample is an approximation of the whole population, so the optimal discretization built on a single sample set is not necessarily the global optimal one. Resampling gives a better estimate of the discretization point distribution in terms of achieving a well-defined distribution. Applying our discretization point selection protocol on the frequency distribution achieved by resampling, significantly improves the quality of discretization and prediction rate and thus, nearing to a global optimal solution. Moreover, the same protocol when applied to the frequency point distribution of random samples, achieved much lesser improvements in the prediction rate as compared to bootstrap. We applied our protocol (after resampling) to various methods. Except for Chi-Merge, all the other methods provide small variations in terms of prediction rates. MDLPC performs the best and FUSBIN achieves the best time complexity, which is a key point when dealing with a lot of examples. As future work, we shall apply this discretization approach in the context of decision trees, to see whether it improves the global performance or not. But, at the same time carrying out this approach needs to answer some other questions such as time complexity. This may lead also to apply the potential discretization points in the context of fuzzy or soft discretization [12] in decision trees.

Références

1. D.A.Zighed,S.Rabaséda,R.Rakotomalala. Discretization Methods in Supervised Learning. Encyclopedia of Computer Science and Technology, vol40, pp 35-50, 1998.

2. L.Breiman, J.H.Friedman, R.A.Olshen, C.J.Stone. Classification and Regression Trees. Wadsworth International, San Francisco, 1984.
3. L.Wehenkel. An Information Quality Based Decision Tree Pruning Method. Proceedings of the 4th International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems, IPMUŠ92(1992).
4. R.Kerber. Discretization of Numeric Attributes. Proceedings of the Tenth National Conference on Artificial Intelligence, MIT Press, Cambridge, MA, 1992, pp.123-128.
5. D.A.Zighed, R.Rakotomalala and S.Rabaséda. Discretization Method for Continuous Attributes in Induction Graphs. Proceeding of the 13th European Meetings on Cybernetics and System Research, 1996, pp.997-1002.
6. U.M.Fayyad, K.Irani. Multi-interval Discretization of Continuous-Valued Attributes for Classification Learning. Proceedings of the 13th International Joint Conference on Artificial Intelligence, Morgan Kaufmann,San Mateo,CA,1993,pp1022-1027
7. T.Van de Merckt. Decision Trees in Numerical Attribute Spaces. Proceedings of the 13th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Mateo, CA, 1993, pp 1013-1021.
8. Mooney, C Z Duval, R D (1993). Bootstrapping. A Nonparametric Approach to Statistical Inference. Sage University Paper series on Quantitative Applications in the Social Sciences, 07-095. Newbury Park, CA : Sage.
9. A. Kusiak. Feature transformation methods in data mining. IEEE Trans. on Electronics Packaging Manufacturing, 24(3) :214-221, 2001.
10. H. Liu, F. Hussain, C. L. Tan, and M. Dash. Discretization : An enabling technique. Data Mining and Knowledge Discovery, 6(4) :393-423, 2002.
11. J. R. Quinlan. Improved use of continuous attributes in c4.5. Journal of Artificial Intelligence Research, 4 :77-90, 1996.
12. Y. Peng and P. Flach. Soft Discretization to Enhance the Continuous Decision Tree Induction. Integrating Aspects of Data Mining, Decision Support and Meta-Learning, pages 109-118, ECML / PKDD'01 workshop notes, September 2001.

Summary

Decision tree induction has been widely used to generate classifiers from training data through a process of recursively splitting the data space. In the case of training on continuous-valued data, the associated attributes must be discretized in advance or during the learning process. We generate discretization points by performing resampling on the original data set and then produce a selection of discretization points by using our resampling selection protocol. We also generate discretization points using ordinary random sampling and we calculate the prediction rate of the discretization points obtained using both sampling and resampling techniques. This process is repeated using the different discretization strategies mentioned above. Thus, the goal of this paper is to observe whether the resampling technique can lead to better discretization points, which opens up a new paradigm to construction of decision trees.