International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:1, 2009

A Decision Boundary based Discretization Technique using Resampling

Taimur Qureshi Djamel A Zighed

Abstract—**Many supervised induction algorithms require discrete data, even while real data often comes in a discrete and continuous formats. Quality discretization of continuous attributes is an important problem that has effects on speed, accuracy and understandability of the induction models. Usually, discretization and other types of statistical processes are applied to subsets of the population as the entire population is practically inaccessible. For this reason we argue that the discretization performed on a sample of the population is only an estimate of the entire population. Most of the existing discretization methods, partition the attribute range into two or several intervals using a single or a set of cut points. In this paper, we introduce a technique by using resampling (such as bootstrap) to generate a set of candidate discretization points and thus, improving the discretization quality by providing a better estimation towards the entire population. 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 soft decision trees.**

Index Terms—**Bootstrap, discretization, resampling, soft decision trees.**

I. Introduction

DISCRETIZATION is a general purpose preprocessing method that can be used for data exploration or data preparation in data mining. While they are critical in the case of decision tree methods, discretization methods can also be used for bayesian networks, rule-set algorithms or logistic regression. However, discretization methods have mainly been evaluated using decision trees such as CART [2] or C4.5 [3]. Many real-world classification algorithms are hard to solve unless the continuous attributes are discretized and Kusiak [13] emphasized that the choice of the discretization technique has important consequences on the induction model used. In the Top-Down Induction of Decision Trees family, the algorithms for discretization are based mostly on binarization within a subset of training data [11]. A simple unsupervised discretization procedure divides the range of a continuous variable into equal-width intervals or equal-frequency intervals. While, supervised methods use information quality or statistical quality based measures to determine the interval boundary points. These algorithms reduce the number of attributed values maintaining the relationship between the class and attribute values.

Y.Yang et al [18] and Liu et al. [14] classified discretization methods from into different viewpoints: supervised vs. unsupervised, static vs. dynamic, global vs. local, top-down vs. bottom-up, Parameterized vs. Unparameterized, Disjoint vs. Non-disjoint, Univariate vs. Multivariate, primary vs. composite and Split vs. Merge. Out of these, Top-down methods as FUSBIN and MDLPC [6], [7] start with one interval and split intervals in the process of discretization. While, bottom-up methods like FUSINTER[6] and Chi-Merge [5] 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 to find a way to produce better discretization points. Previously, various studies have been done to estimate the discretization points from learning samples taken from the population. Because of inaccessibility of entire populations, we usually try to estimate statistical processes such as discretization from samples rather than the population. Significantly, in [1], a set of learning samples are used to approximate the best discretization points of the whole population. They argue that the learning sample is just an approximation of the whole population, so the optimal solution built on a single sample set is not necessarily the global one. Taking this point into consideration, in this paper we try to provide a better estimate toward the entire population.

Our interpretation of the above problem leads us to use a resampling approach [4] to determine better distribution estimates of the data sample. Thus, we focus on obtaining discretization points from the data sample which has a higher probability to be the 'better estimate' in terms of distribution of the whole population and so, the resulting discretization to be a better estimate as well. By doing so, we attempt to improve on the predication accuracy of discretization and thus, treating the discretization problem in the statistical area with new results. We use ordinary bootstrap [9] as a method for resampling in our approach which tries to improve on the above mentioned problem. We argue that the recent increase in processing power of computers has allowed us to use extensive resampling analysis in order to find better estimates of the larger population.

In this paper we focus on supervised discretization. In our approach, we create a histogram density function of an attribute X_i by repeatedly resampling the data sample and obtaining a collective distribution of class frequencies (in percentages), which is obtained at each value of the attribute X_i from the entire resampled data. As a result we get an ensemble distribution function for all the classes. By doing so we try to obtain a better estimate of the class distribution of the data in relation to the entire population. Next, we apply a

smoothing procedure to this distribution function and then find the resulting decision boundaries which, we consider as our discretization points. This work has also enabled us to extend this concept a step further towards soft or fuzzy discretization [16] which, is out of scope of this paper.

In section 2, we lay out the framework for discretization and define the various terminologies. We also discuss the various data sets used in our experiments. In 3, we briefly describe various existing top-down and bottom-up discretization strategies used in our paper. Then, in section 4, we give a detailed illustration of our proposed scheme and then in 5, we show our results by applying the explained techniques to 10 benchmarking data sets. We also compare different discretization strategies to ours and at the end we conclude with observations, deductions and proposals for future work.

II. Terminologies and Definitions

A. Framework and Formulation

Let $X(.)$ be an attribute value on the real set $\Re$. 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 $C(.)$. The purpose of the method is to build a model which can predict class $C(.)$ relative to $X(.)$.

B. Resampling

We use an Ordinary Bootstrap method [**?**] for resampling in which the learning and test sets overlap. In this, a prediction rule is built on a bootstrap sample and tested on the original sample, averaging the misclassification rates across all bootstrap replications gives the ordinary bootstrap estimate.

1) Smoothing: We use simple moving average (SMA) as a smoothing technique which is the unweighted mean of the previous n data points.

2) Decision Boundary: The decision boundary DB [18] is the intersection of any two class based distribution functions which, (in our case) are built from resampling the sample data and building (then smoothing) a histogram based on class frequencies. This is illustrated in fig 1.

C. Quality of Discretization

We use three factors to analyze the quality of the obtained discretization.

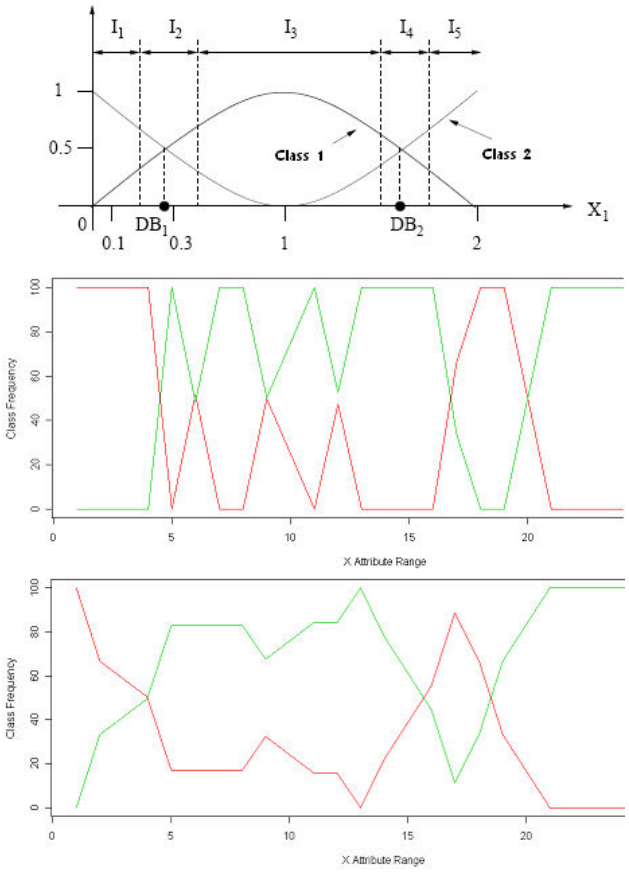

Fig. 1. (a) Decision boundaries for class 1 and 2 (Top). (b) Two class frequency distribution before smoothing (Middle). (c) Two class frequency distribution after smoothing (Bottom)

1) Prediction Accuracy: The goal of the discretization is to make the class $C(.)$ predictable by an attribute $X(.)$. To measure this prediction rate we define a notion of prediction accuracy of the achieved discretization as follows:

The discretization of the attribute X_j from a sample Ω_s, provides k intervals denoted $I_i^j; i = 1, .., k$. For each ω taken from the test sample Ω_t we denote I_i^j the interval to which it belongs after discretization of the sample Ω_s. The point ω will be labeled $C(\omega) = c^*$ if the majority of the points in the Ω_t that are in I_i^j have the class c^*. This corresponds to a bayesian decision rule with a matrix of symmetrical costs and prior probabilities of the classes estimated by the proportion of the individuals belonging to each class in the Ω_t. We measure the quality of the discretization by the rate of good predictions:

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

We denote by τ_j the good prediction rate resulting from the discretization of X_j obtained by applying a method on the sample Ω_s.

2) Complexity: In measuring the complexity of a discretization we take into account the number of intervals I_{number} obtained in the resulting discretization. Large number of intervals increase the complexity of the induction algorithm

that use the discretized input and also because a large number of intervals are likely to increase the discretization bias and variance. Higher discretization variance effects the quality of discretization even if the prediction accuracy is high. This property is explained by Yang et al [18].

3) Robustness: We introduce a concept of robustness. This is equal to the prediction accuracy in terms of the training sample divided by the predication accuracy of the whole population (which is known in our experiments). This measures the degree of accurate estimation of the population from a small training sample.

D. Data Sets

In this article, we used 10 data sets for comparisons and analysis taken from U.C. Irvine repository [19] shown in figure 2.

Dataset	Continuous Attributes	Nominal Attributes	Size	Class Values	Majority Class
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
Heart	10	3	270	2	55.56
Hepatitis	6	13	155	2	79.35
Hypothyroid	7	18	3163	2	95.23
Iris	4	0	150	3	33.33
Pima	8	0	768	2	65.10
Waveform	21	0	5000	3	33.92

Fig. 2. Data sets and their summary.

III. Evaluated Discretization Techniques

We have evaluated 7 different discretization methods. Out of these 3 methods use topdown technique such as (MDLPC,Fusbin,BalancedGain) and 3 use bottomup (Fusinter,Chimerge,MODL), while 1 is based on an optimal algorithm (Fisher). A brief description of these methods is given below, while we describe 2 of these methods in detail.

The MDLPC method is a greedy top-down split method, whose evaluation criterion is based on the Minimum Description Length Principle [**?**]. At each step of the algorithm, the MDLPC evaluates two hypotheses (to cut or not to cut the interval) and chooses the hypothesis whose total encoding cost (model plus exceptions) is the lowest. The BalancedGain method exploits a criterion similar to the GainRatio criterion [15]: it divides the entropy-based Information Gain criterion by the log of the parity of the partition in order to penalize excessive multisplits. Fusbin is a topdown method whereas, the Fusinter method is a greedy bottom-up method. Both exploit an uncertainty measure sensitive to the sample size. Its criterion employs a quadratic entropy term to evaluate the information in the intervals and is regularized by a second term in inverse proportion of the interval frequencies. The ChiMerge [5] method is a greedy bottom-up merge method that locally exploits the chi-square criterion to decide whether two adjacent intervals are similar enough to be merged. MODL [**?**] algorithm is based on a bayesian approach [18]. It defines a criterion which is minimal for the bayes optimal discretization. Fisher's algorithm is a dynamic programming algorithm that tries to find the optimal partition given by W.Fisher, presented in detail by Zighed et al [1].

IV. Proposed Scheme using Resampling

A. Assumptions

(a) In each interval, the distribution of the class values is defined by the frequencies of the class values in this interval. (b) The attributes the conditionally independent of each other given the class. (c) Thus, we discretize one attribute at a time (in two dimensions).

B. Our Approach

Earlier, we argued that the learning sample is just an approximation of the whole population, so the optimal discretization solution built on a single sample set is not necessarily the global one. Thus, our approach is based on finding a better discretization estimate toward the entire population in terms of discretization quality (as discussed above), using a sample selected randomly from that population and then resampling it. We use a ordinary bootstrap based resampling technique to achieve this objective. Our technique is explained below:

C. Resampled Data Distribution (RDD) based Technique using Decision Boundaries

Our proposed discretization technique known as RDD comprises of the following steps:

- We take a random data sample Ω_{rs} from the entire population.
- From this data sample Ω_{rs}, we generate a large data sample Ω_{bs} by repeated resampling of Ω_{rs}, $n = 1000$ times by using ordinary bootstrap.
- Next, we create an ensemble histogram density function of the attribute X_i (to be discretized). This is achieved by merging the plots of all the class frequency histograms (percentage frequencies) as shown in fig 1b.
- Then we apply a smoothing procedure with $n = 3$ to each class frequency curve as illustrated in fig 1c.
- Finally, our discretization points are the decision boundaries which are the intersection of the curves of any two or more classes. This is illustrated in fig 1a.

By building a collective histogram frequency distribution of all the classes, we try to obtain a better estimate of the class distribution of data relative to the entire population. Thus, this can be termed as a prior distribution or a probabilty density function and its application can extend also in *naive bayesian classifiers* [18].

V. Results - Analysis and Comparisons

We obtained discretization points from our approach RDD (as explained above) and we compare this solution to the other 6 discretization methods in the same way as above.

The MODL, MDLPC and BalancedGain methods have an automatic stopping rule and do not require any parameter setting. For the Fusinter criterion, we use the regularization parameters recommended in [1]. For the ChiMerge method, the significance level is set to 0.95 for the chi-square test threshold.

In order to find the add-value of our resampling based discretization techniques and compare them with the above mentioned top-down and bottom-up strategies, we measure the quality of discretization and comparing the strategies considering 10 different datasets.

For our main analysis we used 10 data sets taken from U.C. Irvine repository (figure 2) [19] having 85 continuous attributes denoted as $(X_1(\omega), ..., X_85(\omega))$ and a label $C(\omega)$. We used random learning samples Ω_{rs} consisting of 5 percent of the size of original data sets. We measured the geometric mean of number of intervals $\mu_{I_{number}}$, accuracy rate, and robustness from discretization from 10 random samples of 5 percent of the size of the data set from all the 85 variables using all the 7 methods discussed. Then we calculated RDD as described above. The resulting measures and their results are explained in the following subsections along with figures 3 to 10.

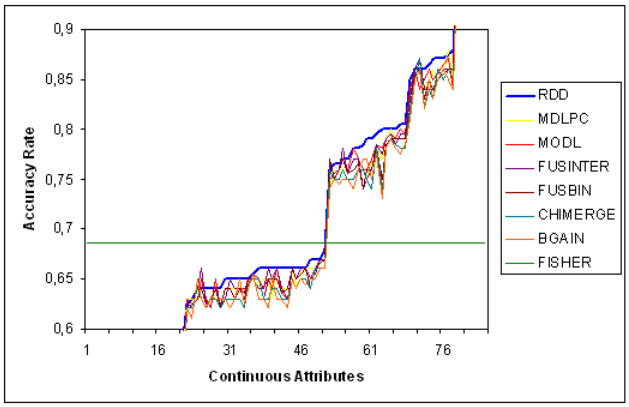

Fig. 3. Comparison of accuracy between the methods.

A. Predication Accuracy

Figure 3 shows the cumulative accuracy rates of all the discretization methods plotted against the 85 continuous attributes taken from 10 data sets of fig 2. It shows the curve of RDD, being above all the curves of the other methods including MDLPC from which our methods are originally built (by resampling and obtaining selected discretization points) as explained earlier.

As we have plotted the curves using all the attributes so the graphical representation might not be very clear, thus we plot the accuracy measurements as repartition functions of the relative differences of accuracy in fig 4 and then we summarize this comparison in the table of fig 9.

In order to analyze the relative differences of accuracy for the 85 attributes in more details, we collect all the geometric

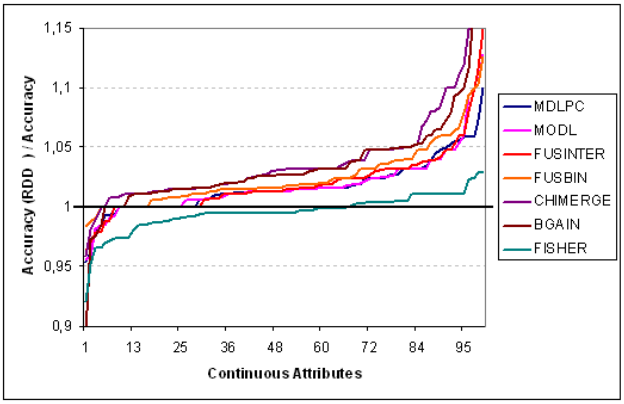

Fig. 4. Repartition function of the relative differences of accuracy between RDD and other methods.

mean ratios per attribute in ascending order. Figure 4 shows the repartition functions of the relative differences of accuracy between RDD method and all the other discretization methods. Each point in this repartition function is the summary of 100 discretization experiments performed on the same attribute. Such repartition functions represent a convenient tool for the fine grain analysis of the differences between methods, in complement with the multi-criteria analysis (that we perform later) carried out on the coarse dataset geometric means. A flat curve reflects two methods that do not differentiate on any of the experiments. A symmetric curve correspond to methods that globally perform equally well, but with differences among the experiences. An unbalanced curve reveals a situation of dominance of one method over the other, with insights on the intensity of the domination and on the size of the region of dominance.

Fig 4 shows the repartition function of all the discretization methods discussed above in comparison with RDD. Our approach dominates Chimerge and Balancedgain in about 90 percent of the attributes with up to 20 percent better accuracy. Its performance is the same with MDLPC, MODL and FUSINTER for about 20 percent of the attributes and dominates them in the rest of 70 percent with up to 15 percent better accuracy. The most important point to note that in this case Fishers optimal algorithm has somewhat of the flat curve showing the closeness of performances in comparison to RDD. Although, it is slightly dominated by Fisher in about 30 percent of the attributes.

B. Complexity - Number of Intervals

In this section we compare the number of intervals generated by each method. Large number of intervals contribute to the complexity of the induction algorithm and add discretization bias and variance as discussed before. Fig 5 shows an asymmetric repartition function of the relative differences of number of intervals generated by each method in comparison to our RDD method. Fig 5 shows that RDD dominates Chimerge and Fusbin in 75 to 90 percent of the attributes which is the curve below the 100 percent mark. The reason

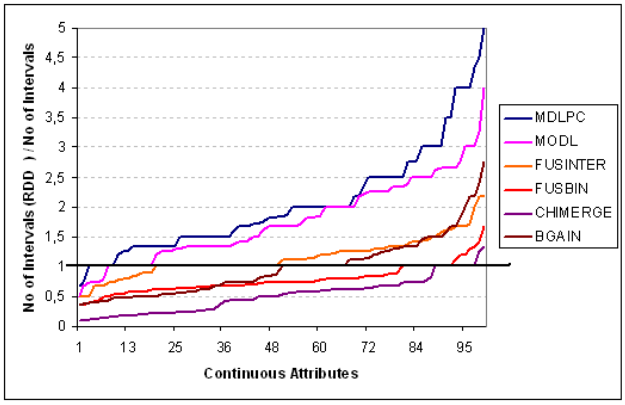

Fig. 5. Repartition function of the relative differences of number of intervals between RDD and other methods.

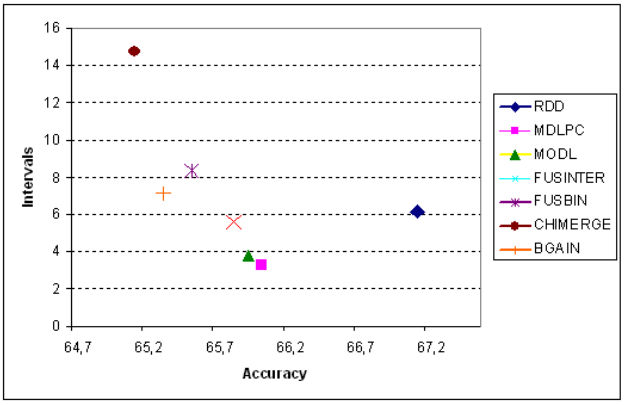

Fig. 7. Bi-criteria evaluation of the methods for the accuracy and number of intervals, using datasets geometric means.

for this is that smaller number of intervals is better than a large number. It demonstrates a flat curve for Fusinter and Balancedgain for about 50 percent of the attributes while, is dominated by MDLPC and MODL for about 80 percent of the attributes.

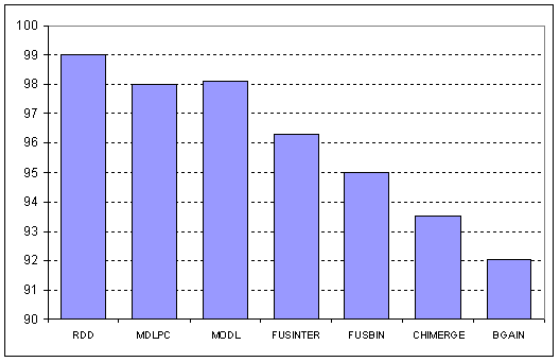

Fig. 6. Repartition function of the relative differences of robustness between RDD and other methods.

C. Bi-criteria Evaluations

In multi-criteria analysis, a solution dominates (or is non-inferior to) another one if it is better for all criteria. A solution that cannot be dominated is Pareto optimal: any improvement of one of the criteria causes a deterioration on another criterion. The Pareto surface is the set of all the Pareto optimal solutions.

In order to analyze both the accuracy and robustness results, we report the dataset geometric means on a two-criteria plan in Figure 7, with the accuracy on the x-coordinate and the robustness on the y-coordinate. Similarly, we report the accuracy and the number of intervals in Figure 8. Each point in these figures represents the summary of all the experiments. The multi-criteria figures are thus reliable and informative: they allow us to clearly differentiate the behavior of almost all the methods.

Accuracy is certainly the most important parameter to distinguish a discretization method so we have grouped it in both the analysis of fig 7 and 8. Fig 7 clearly shows that RDD clearly outperforms all the other methods in accuracy and robustness. MDLPC and MODL are very close to each other with Fusinter not far behind. Balancedgain has the worst robustness but chimerge has the worst accuracy.

In fig 8, RDD again outperforms in terms of accuracy but in terms of the number of intervals is only better than Chimerge, Balancedgain and Fusbin. It is outperformed in this regard by Fusinter, MDLPC and MODL. Chimerge shows a relatively bad result here as well.

D. Comparison Summary

t*	RDD1	RDD2	MDLPC	MODL	FUSINTER	FUSBIN	CHIMERGE	BGAIN	FISHER
RDD	X	1,521111	8,151561	7,382576	7,4419605	10,19841	11,16223618	9,149569	-9,70607
MDLPC			X	-0,76899	-0,709601	2,046845	3,010675182	0,998008	-17,8576
MODL				X	0,0593845	2,81583	3,779660182	1,766993	-17,0887
FUSINTER					X	2,756446	3,720275682	1,707609	-17,148
FUSBIN						X	0,963826182	-1,04884	-19,9045
CHIMERGE							X	-2,01267	-20,8683
BGAIN								X	-18,8556
FISHER									X

Fig. 8. Comparison of the critical area between all the methods.

Instead of comparing the mean accuracy of all the methods, we measure the critical area t^* of the difference of each method. The methods are compared two by two according to the following statistical procedure. Let u and v be two methods to compare. We form the difference Γ_{uv} between the rates of well ordered elements of the methods u and v. This difference is a random attribute which is roughly normal with parameters (μ, σ). We conclude that u is better than v if μ is significantly superior to 0.

We have n = 85 * 10 observations. The estimated mean value μ and mean standard deviation σ are:

$$\mu_{uv} = \frac{1}{850} \sum_{j=1,s=1}^{11,21} \gamma_{js}^{uv} \text{ ; where, } \gamma_{js}^{uv} = \gamma_{js}^{u} - \gamma_{js}^{v}$$

$$\sigma_{uv} = \sqrt{\frac{1}{850} \sum_{j=1,s=1}^{11,21} \gamma_{js}^{uv} - \mu_{uv}}$$

The critical area is:

$$t^* = \frac{\mu_{uv}}{\sigma_{uv}/\sqrt{n}} > t_{1-\alpha}$$

with $t_{1-\alpha}$ the critical value at the rate α of a Student's law with $(n-1)$ degrees of freedom. Since, n is large, we have for $\alpha = 0.05$, $t_{1-\alpha/2} = 1.96$. The computed t^* results are reported in figure 9. Positive values of t^* indicate that the method in the row is better than the method in the column. Aside from Chi-Merge method whose results are relatively bad, all the other methods have relatively smaller differences. However, RDD significantly, report much better results and draw much closer to Fishers optimal results.

E. Time Complexity

In terms of time complexity among these methods MDLPC seemed to be the best with a much lesser time complexity. FUSBIN and FUSINTER also have a smaller time complexity in comparison to Fisher's optimal algorithm which is the most computationally intensive. The time complexity of RDD depends mainly on the number of bootstrap samples generated, but the more bootstrap samples, the better the discretization quality. For maximum of 500 bootstrap samples RDD performs the best, but if the bootstrap samples are increased the performance suffers. This is a trade-off between time complexity and quality. But with vast improvements in computing speeds, we argue that quality could be a much valuable commodity.

VI. 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. Our Resampling based approach tends to give a better data distribution estimate in terms of achieving better discretization quality. Applying our technique although suffers a little in terms of the number of intervals, but improves robustness and prediction accuracies and thus, aiming to arrive nearer to a global optimal solution (Fisher's optimal algorithm). Except for Chi-Merge and Balancedgain, the other methods provide small variations in terms of prediction rates. MDLPC performs the best in terms of number of intervals and time complexity. Also RDD is less computation intensive than Fisher and bottom-up methods.

This work has led us to apply this technique in the context of fuzzy or soft discretization [16] for decision trees, which has enabled us to significantly lower the misclassification rates. 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. We shall also try to examine the potential of resampling as prior distributions in naive bayesian classifiers [18]. But, at the same time carrying out this approach needs to answer some other questions such as the optimal number of bootstrap samples and time complexity.

References

[1] D.A.Zighed,S.Rabasda,R.Rakotomalala. Discretization Methods in Supervised Learning. Encyclopedia of Computer Science and Technology, vol40,pp35-50,1998.

[2] L.Breiman, J.H.Friedman, R.A.Olshen, C.J.Stone. Classification and Regression Trees. Wadsworth International, San Francisco, 1984.

[3] J.Quinlan.C4.5:Programs for Machine Learning. M.Kaufmann,SanMateo,CA,1993.

[4] 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, IPMU'92(1992).

[5] R.Kerber. Discretization of Numeric Attributes. Proceedings of the Tenth National Conference on Artificial Intelligence, MIT Press, Cambridge, MA, 1992, pp.123-128.

[6] D.A.Zighed, R.Rakotomalala and S.Rabasda. Discretization Method for Continuous Attributes in Induction Graphs. Proceeding of the 13th European Meetings on Cybernetics and System Research, 1996, pp.997-1002.

[7] 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

[8] D.A.Zighed and R.Rickotomalala. A Method for Non Arborescent Induction Graphs. Technical Report, Laboratory ERIC, University of Lyon 2 , 1996.

[9] 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.

[10] J. Catlett. On changing continuous attributes into ordered discrete attributes. In Proceedings of the European Working Session on Learning, pages 164-178., 1991.

[11] J. Y. Ching, A. K. C. Wong, and K. C. C. Chan. Class-dependent discretization for inductive learning from continuous and mixed mode data. IEEE Trans. on Pattern Analysis and Machine Intelligence, 17(7):641-651, 1995.

[12] T. Elomaa and J. Rousu. General and efficient multisplitting of numerical attributes. Machine Learing, 36(3):201-244, 1999.

[13] A. Kusiak. Feature transformation methods in data mining. IEEE Trans. on Electronics Packaging Manufacturing, 24(3):214-221, 2001.

[14] H. Liu, F. Hussain, C. L. Tan, and M. Dash. Discretization: An enabling technique. Data Mining and Knowledge Discovery, 6(4):393-423, 2002.

[15] J. R. Quinlan. Improved use of continuous attributes in c4.5. Journal of Artificial Intelligence Research, 4:77-90, 1996.

[16] Y. Peng and P. Flach. Soft Discretization to Enhance the Continuous Decision Tree Induction. Integrating Aspects of Data Mining, Decision Support and Meta-Learning, Christophe Giraud-Carrier, Nada Lavrac and Steve Moyle, editors, pages 109–118, ECML/PKDD'01 workshop notes, September 2001.

[17] Efron B, Tibshirani R. An Introduction to the Bootstrap. Chapman and Hall, 1998.

[18] Y. Yang and G. I. Webb. Discretization for naive-bayes learning: managing discretization bias and variance. Technical Report 2003/131, School of Computer Science and Software Engineering, Monash University, 2003.

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