Class-Driven Statistical Discretization of Continuous Attributes (Extended Abstract)

M. Richeldi and M. Rossotto

CSELT (Centro Studi e Laboratori Telecomunicazioni) - Torino, ITALY

Abstract. Discretization is a pre-processing step of the learning task which offers cognitive benefits as well as computational ones. This paper describes StatDisc, a statistical algorithm that supports supervised learning by performing class-driven discretization. StatDisc provides a concise summarization of continuous attributes by investigating the data composition, i.e., by discovering intervals of the numeric attribute values wherein examples feature distribution of classes homogeneous and strongly contrasting with the distribution of other intervals. Experimental results from a variety of domains confirm that discretizing real attributes causes little loss of learning accuracy while offering large reduction in learning time.

1. Introduction

Discretization is a pre-processing step of the learning task. It is performed by dividing the values of a continuous attribute into a small number of intervals, where each interval is mapped to a discrete symbol. The discretization process constructs an abstraction space over the continuous attributes. Learning in this new abstraction space has several advantages. First, it allows for effective feature construction. Second, dependence analysis between continuous and nominal attributes can be performed. Finally, discretization results in substantial speed-up for the inductive process, i.e., it cuts down the computational cost of the learning task [1].

Simple methods for discretizing a continuous attribute have appeared in the statistical literature. [2] compares three techniques: the *equal-width discretization*, the *equal-frequency discretization*, and the *maximum marginal entropy discretization*. methods. They all require the user to specify the number of intervals into which each attribute will be partitioned. These methods are easy to implement but performs poorly in many situations. The main reason for their failure is their inherent "class-blindness", i.e., they ignore the class of examples when partitioning the training set. A classification algorithm will be no longer able to separate examples of different classes that have been grouped into the same interval. Conversely, a discretization that takes into account class distribution of examples, termed a *class-driven discretization technique*, will retain examples of different classes into different intervals and produce the right partitions.

2. Class-driven discretization techniques

Class-driven methods achieve a good discretization by approximating the class distribution of the continuous attribute. True class distributions are estimated by relative class frequencies. Training examples are first sorted according to their value of the attribute being discretized. Each training example is regarded as a single-

element interval. Then, an association measure is applied continuosly to ascertain whether adjacent intervals feature dissimilar relative class frequencies. If they do not, intervals are merged. As a result, the method yields high intra-interval uniform and high inter-interval different partitions.
Different association measures can be used to compare relative class frequencies of adjacent intervals. Catlett's D2 [1] exploits information gain ratio and results are quite reasonable. A more effective approach has been introduced by Kerber in [3]. Kerber proposes the statistical measure χ^2 to evaluate the similarity of relative class frequencies of adjacent intervals. The χ^2 statistic is used to compare the class distribution of each single interval (observed distribution) with the class distribution that would be expected if the pair of intervals were independent of the class attribute (hypotethical, or expected distribution). If the hypothesis of independence is confirmed, the intervals should be merged, since their relative class frequencies are very similar. Conversely, the difference in the class distribution of the two intervals is statistically significant, and the intervals should remain separated.
Chi-Merge is robust with respect to class-blind methods but suffers a major shortcoming: it is inclined to produce more intervals when there are more examples. When the observed distribution is fixed, the χ^2 value augments proportionally to the increase in the number of examples. Consequently, the algorithm produces a very high number of intervals for medium or large data sets (see experimental results relating to Segment dataset in section 4). Further limitations are: First, it examines pair of adjacent intervals only, ignoring other surrounding intervals. Thus, it is possible that the formation of large, uniform intervals is prevented by this restricted local analysis. Second, it produces a fixed partition of the attribute values. Lastly, the algorithm can separate adjacent examples of the same class by assigning them to two different intervals. This is highly undesirable: a poor class separation obtained by the discretization algorithm makes the learning process harder.

3. The StatDisc Algorithm

StatDisc (*Stat*istical *Disc*retization) overcomes most of the undesirable properties of the techniques described in section 2. It consists of three phases: initialization; interval hierarchy creation; selection of the best discretization.
1. The initialization phase. StatDisc is initialized by sorting the training examples according to their value for the attribute being discretized. The initial discretization is obtained by grouping adjacent examples labelled by the same class value in the same interval. Thus, no time is wasted to merge examples which are known to belong to the same concept. Conversely, ChiMerge always applies the merging process on all examples, even if it is not necessary.
2. *The creation of the interval hierarchy.* StatDisc's merge process creates a hierarchy of intervals. Each level of the hierarchy represents a discretization of the continuous attribute. The construction of the tree proceeds bottom-up. Intervals that were created in the initialization phase are associated to leaf nodes. StatDisc repeatedly selects a set of adjacent intervals to merge by using the *merge criterion* that will be described later on. A new non-leaf node is added to the tree whenever a merge occurs. This node, associated to the new interval, is connected to the nodes that represent the intervals that have been merged. The root of the tree is associated

to the number line. All the training examples belong to this interval. When the merging process is over, the interval hierarchy is explored and a discretization automatically selected according to the characteristic of the data (selection of the best discretization phase). However, one can decide to select a different discretization from the one suggested by the algorithm. The hierarchy provides an insight into the data composition.

Changing the scope of the merging process. The user can select the maximum number N of adjacent intervals that are examined to perform a merge step. By enlarging the scope of the merging process, we moderately slow down the algorithm but strongly decrease the likelihood of missing to form large, uniform intervals.

Statistics for measuring interval similarities. We surveyed most of the statistics in the two-way tables [5] to find out an effective and statistically sound criterion for merging adjacent intervals. χ^2, Fisher's exact test, and measures related to χ^2, e.g. Cramer's *V* test, the contingency coefficient *P*, and Φ (Phi), can be used to compare relative class frequencies of adjacent intervals. The considerations below drew us to the conclusion that Φ is the association measure most appropriated for the merge task. First, χ^2 yields different results when comparing the same distributions for different sample sizes. Second, Φ overcomes the shortcomings of χ^2, as it is truly independent of the cardinality of intervals. Third, it can be shown that Cramer's *V* is equivalent to $\sqrt{2}\Phi$ and *P* is equivalent to $\sqrt{\Phi^2/(1+\Phi^2)}$ when used to test similarities of class fequencies. Lastly, Fisher's exact test is sometimes moderately more precise of Φ but it is very much computationally expensive.

The merge criterion. The merging process contains two phases, repeated continously. In the first one, StatDisc computes the Φ statistic for any N-uple of adjacent intervals. In the second phase, it merges the N-uples with the lowest Φ value. Merging continues until all N-uples of intervals have a Φ value greater than $\Phi_{\nu,\alpha}(\eta)$. $\Phi_{\nu,\alpha}(\eta)$ denotes the value of the Φ distribution at the desired level of significance α and degrees of freedom ν for a sample of size η. A two-steps heuristic is then applied to force the merging process to continue until a one-interval partition is obtained.

3. Selection of the best discretization. StatDisc seeks the largest partition that was obtained before decreasing the significance level α. If this search fails, it returns the partition which, on average, contains the largest adjacent intervals whose relative class frequencies are the most dissimilar.

4. Results of experiments

To test StatDisc on classification accuracy, we ran the following experiment on six domains with real-valued attributes. The domains were obtained from LIACC, University of Port [6]. They are: Australian (690 cases); Diabetes (768 cases); Hypertyroid (3772 cases); Segment (2310 cases); Vehicle (846 cases); Glass (214 cases). As StatDisc is not a classification algorithm itself, it was used to create intervals for C4.5 [4]. We averaged the results of three 10-fold cross validation test trials for each domain. We ran C4.5 on raw continuous data, then discretized the

dataset using equal-width discretization (EW), equal-frequency discretization (EF), D2, Chi-Merge (CM) where possible, StatDisc (SD), and ran C4.5 on discretized data. Equal-width, equal-frequency, and D2 were forced to create 5- and 10-interval discretizations. Next tables show the results of the comparison experiments. Cell entries report mean error rate on the test set in percentage.

Dataset	raw data	EF (5)	EF (10)	EW (5)	EW (10)	D2 (5)	D2 (10)	CM [α]	SD
Australian	15.5	14.9	14.0	15.8	14.9	14.8	14.5	15.2 [0.01]	13.6
Diabetes	27.9	27.2	28.9	24.8	25.6	24.1	25.3	23.8 [0.005]	23.3
Hypertyroid	0.5	6.7	6.7	7.1	6.6	0.7	0.7	0.4 [0.001]	0.4
Segment	4.0	6.3	5.0	7.2	5.0	6.9	4.6	---	3.7
Vehicle	28.8	31.7	28.3	33.1	29.7	28.7	29.2	28.9 [0.005]	26.8
Glass	32.9	36.3	31.6	38.1	34.2	29.8	27.8	26.8 [0.05]	25.2

Table 1. Results of experiments.

Some comparisons with class-blinded methods showed that their performance is quite unpredictable. Furthermore, their performance is nearly always inferior to the one of class-driven techniques. All the domains were improved by discretization. Our results confirm experiments reported in [1]. Two are the possible explanations. First, discretized trees are smaller than trees on original data. Second, discretization can accurately approximate class distribution from the full data set.

Chi-Merge could not produce a discretization with less than 50 intervals in the Segment domain. It could hardly discretize Diabetes and Hypertyroid domains as well. Partitions with less than 20 intervals could be obtained only by decreasing the level of significance [α] to the limit. However, the performance of Chi-Merge on these two last datasets was close to the one of StatDisc. This result was expected, since both algorithms base on chi-squared related statistics.

Two-sided t-test at the 5% level confirmed that StatDisc showed a significant improvement in performance in regard to the other algorithms. StatDisc seems to provide a more concise and effective summarization of continuous attributes.

Comparison of learning time showed that C4.5 took half the time to process discretized dataset. We can safely conclude that discretization can speedup the learning task with very little, if any, loss of accuracy.

5. References

[1] J. Catlett. "On Changing Continuous Attributes into Ordered Discrete Attributes," *Proc. of the EWSL-91*, 164-177, 1991.

[2] D. Chiu, A. Wong, et al. "Information Discovery through Hierarchical Maximum Entropy Discretization and Synthesis," *Knowledge Discovery in Databases*, The AAAI Press, 1991.

[3] R. Kerber. "ChiMerge: Discretization of Numeric Attributes," *Proc. of the AAAI-92*, 1992.

[4] J. Quinlan. "C4.5: Programs for Machine Learning," Morgan Kaufmann Publ., 1993.

[5] M. Kendall and A. Stuart. "The Advanced Theory of Statistics," Griffin London, 1973.

[6] P. Brazdil, J. Gama, and B. Henery. "Characterizing the Applicability of Classification Algorithms Using Meta-Level Learning," *ECML-94*, Springer Verlag, 1994.