Proceedings of the Seventh Florida Artificial Intelligence Research Symposium, pp. 117-21, 1994

BRACE: A Paradigm For the Discretization of Continuously Valued Data

Dan Ventura
Tony R. Martinez

Computer Science Department, Brigham Young University, Provo, Utah 84602
e-mail: dan@axon.cs.byu.edu, martinez@cs.byu.edu

Abstract

Discretization of continuously valued data is a useful and necessary tool because many learning paradigms assume nominal data. A list of objectives for efficient and effective discretization is presented. A paradigm called BRACE (Boundary Ranking And Classification Evaluation) that attempts to meet the objectives is presented along with an algorithm that follows the paradigm. The paradigm meets many of the objectives, with potential for extension to meet the remainder. Empirical results have been promising. For these reasons BRACE has potential as an effective and efficient method for discretization of continuously valued data. A further advantage of BRACE is that it is general enough to be extended to other types of clustering/unsupervised learning.

1 INTRODUCTION

Many machine learning and neurally inspired algorithms are limited, at least in their pure form, to working with nominal data. Examples include ID3 [8], AQ* [6], COBWEB [2], and ASOCS [5]. For many real-world problems, some provision must be made to support processing of continuously valued data. The main effort of this research is to develop an effective and efficient method for discretization of continuously valued data as a preprocessor for supervised learning systems. Discretizing continuously valued data produces inherent generalization by grouping data into several ranges, representing it in a more general way. It has been shown that for some learning algorithms, efficient discretization as preprocessing resulted in significant speedup [1]. Further, discretization has some psychological plausibility since in many cases humans apparently perform a similar preprocessing step representing temperature, weather, speed, etc., as nominal values.

This paper attempts to support models which naturally work with nominal data by presenting a method of quickly and intelligently discretizing continuously valued data into several useful ranges that can then be handled as nominal data. The method concentrates on finding the natural boundaries between ranges and creates a set of possible classifications using these boundaries. All classifications in the set are evaluated according to a criterion function and the classification that maximizes the criterion function is selected. This extended abstract gives a high-level view of the BRACE paradigm with brief examples and discussion. The full paper fleshes out the description of the paradigm with formal definitions, detailed examples, and samples of empirical results.

2 RELATED WORK

Many methods for general clustering of data suggest themselves for the purpose of discretizing continuously valued data. Methods considered include K-means, Mini-max, Bayes, and K-nearest neighbor [9][3]. These methods solve many general clustering problems. However, each of these methods has been empirically evaluated on real-world applications, and none of them are particularly suited to this specific task of discretizing real data into nominal data. Reasons for this include use of a Euclidean distance metric to determine range assignment, dependence on user-defined parameters, dependence on unknown statistical distributions, tendency of "good" parameter settings to be application dependent, and sensitivity to initial ordering of data.

Several methods specifically for discretization also exist including the equal-width-intervals method, the equal-frequency-intervals method, and ChiMerge [4]. These generally outperform the methods mentioned above since they are designed specifically for the task of discretization. However, these too have inherent weaknesses that limit their usefulness. For example, the two former methods are very simple and arbitrary.

They make no attempt to discover any information inherent in the data. The latter is more robust; however, it performs only pairwise evaluation of ranges and depends on some user-defined parameters.

Study of these methods has resulted in the compilation of a list of desirable attributes for a discretization method:

- Measure of classification "goodness"
- No specific closeness measure
- No parameters
- Globality rather than locality
- Simplicity
- Use of feedback
- Use of *a priori* knowledge
- Higher order correlations
- Fast

3 BRACE: BOUNDARY RANKING AND CLUSTERING EVALUATION

Boundary Ranking And Clustering Evaluation (BRACE) is a paradigm designed to meet the objectives presented above. A high-level description of the paradigm consists of the following 4 steps:

1. Find all natural boundaries in the data;
2. Calculate the *rank* of each boundary;
3. Construct a set of possible classifications;
4. Evaluate all classifications in the set and choose the best one.

Each specific algorithmic instantiation of the BRACE paradigm uses a unique *boundary definition* (BD) for finding boundaries, *boundary ranking function* (BRF) for measuring each boundary and *classification optimization function* (COF) for evaluating classifications. A *classification* is defined as a grouping of the data into two or more discrete intervals. The definition and functions are based on some heuristic or bias and depend on the specific algorithm. One such algorithm is presented in section 4. Other BRACE algorithms introduce other methods of finding boundaries in the data, measuring those boundaries, and evaluating the classifications in the set.

A set *C* of possible classifications is constructed which consists of the most likely 2-interval classification, the most likely 3-interval classification, etc. up to the most likely *B+1*-interval classification, where *B* is the number of boundaries identified. The first classification consists of dividing the data into two intervals by using the boundary with the highest rank as a division. The second classification is obtained by dividing the data into three intervals using the first two highest ranked boundaries. This continues until the data has been split into *B+1* intervals and *C* contains *B* different classifications.

Once a set of possible classifications has been created, one of them must be selected for use in discretizing the data. Each of the classifications in the set are evaluated according to the COF and the one classification that maximizes it is selected. All values placed into the same range by the chosen classification are assigned the same nominal value.

4 VALLEY: AN INSTANTIATION OF BRACE

Define:

H as a histogram of the continuously valued data points. The *range* of the data is defined as the interval on the real numbers from the smallest data point to the largest. This range is broken up into an initial number of *intervals* and the height of each interval is the number of data points that appear in it. It must be noted that this number of intervals may be considered a parameter. However, it is more of a parameter of the physical implementation rather that one of the logical algorithm. Any reasonably large number of intervals (~50) will reveal roughly the same profile for the data. Of course, the greater the number of intervals created, the greater the number of spurious valleys that must be processed.

valley as a local minimum in *H*.

V as the set of all valleys.

rank(valley) as the rank (measure of "goodness") of a given valley.

C as the set of possible classifications to be considered by the COF.

instance as a set of attribute values (Left Hand Side) implying a set of output values (Right Hand Side). For example,

YES 7.5 BLUE --> TRUE

training set as a set of instances. For example,

```
YES  7.5  BLUE    --> TRUE
NO   1.2  RED     --> FALSE
YES  4.5  GREEN   --> FALSE
            •
            •
            •
```

VALLEY as an instantiation of the BRACE paradigm defined by its specific Boundary Definition, Boundary Ranking Function, and Classification Optimization Function:

BD (Boundary Definition) is a *valley* in a histogram. See figure 1.

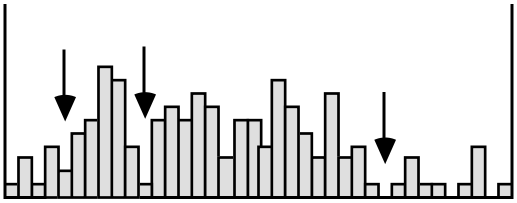

Figure 1. Arrows indicate several valleys

BRF (Boundary Ranking Function) is the area of a valley under the imaginary line between the valley's two bounding local maxima (figure 2.)

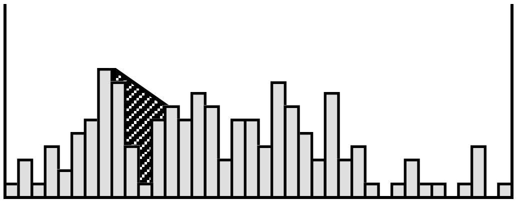

Figure 2. Area of a valley (shaded)

COF (Classification Optimization Function) depends on the data being part of an instance. It is defined as the number of consistent output class predictions minus the number of inconsistent output predictions. These predictions are made considering the discretization of the attribute currently being processed as the entire left-hand side of each instance.

Inserting BD, BRF, and COF into BRACE results in the VALLEY algorithm, which consists of five basic steps:

1. Order the data and build histogram *H.*
2. Find the set *V* of all valleys.
3. For each valley $v \in V$, rank(v) = area of v.
4. Construct set *C* of possible classifications.
5. Evaluate *C* and chose max{COF(c)}, $c \in C$.

The first three steps are relatively straight forward.

Once the valley sizes for all valleys have been calculated, the set *C* of possible classifications is built using the biggest valley, then the 2 biggest valleys and so on until we have split the histogram into $|V|+1$ intervals and obtained $|V|$ classifications in *C* (i.e. using BD and BRF.)

Since the data is being discretized for a *supervised* learning system, knowledge of the output value for each instance in the training set is assumed. It is important to note that this is an assumption made by the VALLEY algorithm and not by the general BRACE paradigm.

Using this knowledge, each classification in *C* is evaluated according to its ability to correctly predict output values for the training set, using only the current attribute under consideration. The number of inconsistent predictions is subtracted from the number of consistent predictions, and the result is the score for that classification. The classification with the highest score is selected (i.e. using COF.)

4.1 An Example

Consider the following *training set* of instances, only the first five of which are actually shown.

YES	7.5	BLUE	--> TRUE
NO	1.2	RED	--> FALSE
YES	4.5	GREEN	--> FALSE
YES	9.2	GREEN	--> TRUE
YES	2.2	RED	--> FALSE

•
•
•

Since the second attribute is continuously valued it must be discretized. Assume that *H* for attribute 2 ranges from 1 to 10 and looks as in figure 3a. Finding all the valleys proceeds as in figure 3 where the numbers above the various valleys indicate their relative rank.

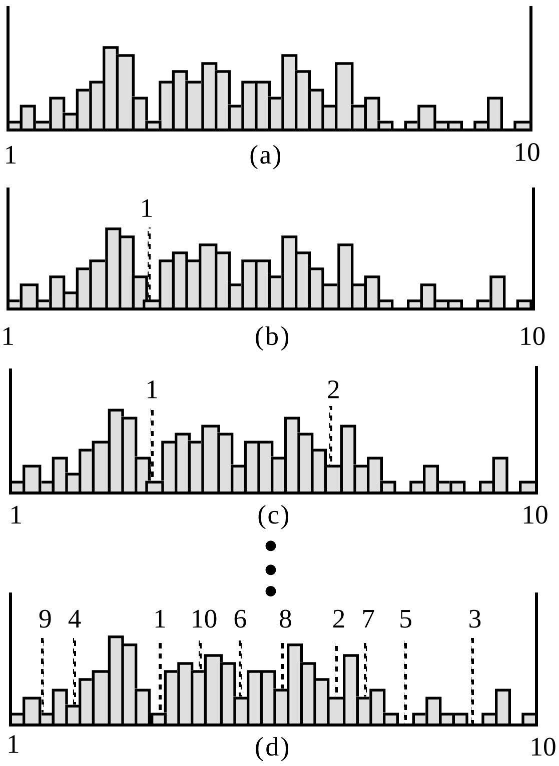

Figure 3. (a) Histogram *H* (b) 1 valley (c) 2 valleys (d) complete set *V*

Recall that the set *C* includes the classification shown in figure 3b, the classification shown in figure 3c, and all the classifications implied between steps c and d in figure 3. Each of these classifications in *C* are evaluated as shown in Table 1. A simplified copy of the training set is "constructed" considering only attribute 2. The table is interpreted as follows: dividing the data into two intervals (as in figure 3b), value 7.5 is in interval 2 and implies TRUE, 1.2 is in interval 1 and implies FALSE, 4.5 is in interval 2 and implies FALSE, 9.2 is in interval 2 and implies TRUE, and 2.2 is in interval 1 and implies FALSE. The score for column one is calculated by finding the number of consistent output predictions minus the number of inconsistent predictions.

2 int.	3 int.	4 int.	5 int.		11 int.
2 --> T	3 --> T	4 --> T	4 --> T		
1 --> F	1 --> F	1 --> F	1 --> F		
2 --> F	2 --> F	3 --> F	3 --> F	• • •	
2 --> T	3 --> T	4 --> T	5 --> T		
1 --> F	1 --> F	2 --> F	2 --> F		
		•			
		•			
		•			
---------	---------	---------	---------		---------
3	5	5	5		5

Table 1. Calculating scores for each classification in *C* using the COF

For the two interval case, interval 1 predicts F twice and interval 2 predicts T twice and F once. The total score for 2 intervals is therefore 4-1=3. The other columns of the table are calculated similarly. For the purpose of illustration, sub scores calculated using only the 5 instances shown are used as the total column score.

Breaking the data into 3 intervals gives the maximum score. Note that every classification thereafter also gives the maximum score. However, in the interest of parsimony and generalization the smallest number of intervals with the maximum score is selected. The data is discretized into three intervals as shown in figure 3c. Values 1.2 and 2.2 are in class 1, 4.5 is in class 2, and 7.5 and 9.2 are in class 3. The modified instance set now looks like:

YES	3	BLUE	--> TRUE
NO	1	RED	--> FALSE
YES	2	GREEN	--> FALSE
YES	3	GREEN	--> TRUE
YES	1	RED	--> FALSE
		•	
		•	
		•	

As a final comment, it should be noted that although the output in the preceding example is binary, VALLEY and BRACE can handle any nominal output. They are not limited to the binary case.

5 EMPIRICAL RESULTS

VALLEY was tested on various data sets from the UC Irvine machine learning database [7]. One example is the hepatitis data set, which has a number of continuously valued variables with widely varying distributions. These six continuously valued variables were discretized with encouraging results (see figure 4.) The next natural step is to compare this method with more traditional ones by feeding the respective discretizations into a supervised learning system and noting effects on the system's ability to learn.

In a study of K-nearest neighbor clustering we found that the K-NN algorithm performed well for discretizing real data--provided that the proper values for K and T (parameters corresponding to size of neighborhood and number of shared neighbors) were used. Unfortunately, there is no method for finding good K and T values other than experimentation. Further, these good values appear to be dependent on the initial data. Sample data from another UCI database on iris flowers was discretized using both K-NN and VALLEY. VALLEY found the same best clustering that K-NN did; however K-NN required repeated attempts with various values for K and T, via interaction with the user, as well as a manual evaluation of classification "goodness." VALLEY found the clustering in one iteration with no user intervention.

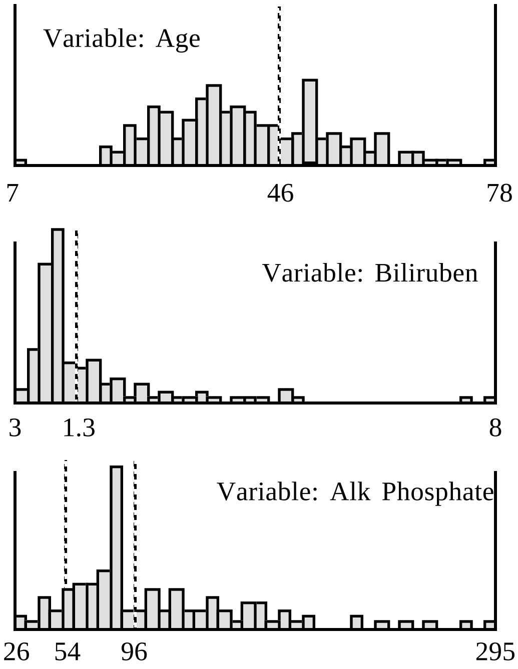

Figure 4. Continuous valued variables from hepatitis data set discretized by VALLEY

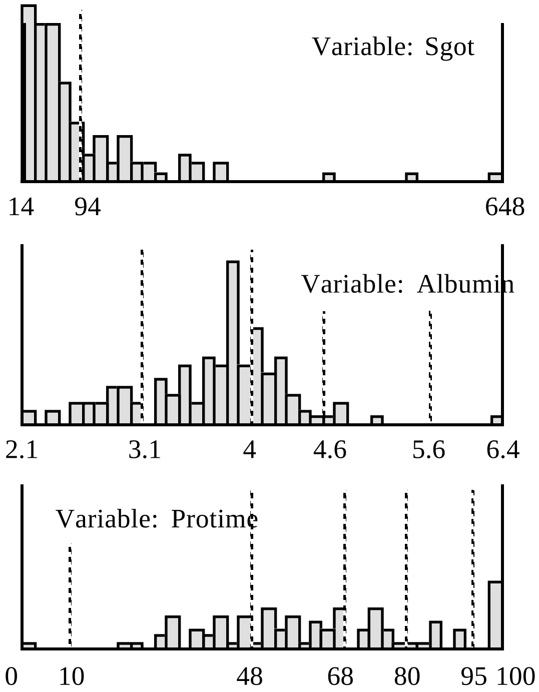

Figure 4. Continued

6 ANALYSIS

Because BRACE is an abstract concept, meaningful analysis of it must be performed upon a concrete instantiation such as VALLEY. Both complexity analysis (O(*nlogn*)) and comparison of advantages vs. disadvantages prove VALLEY to be a promising algorithm for discretization. These encouraging results for VALLEY as an algorithm help validate BRACE as a paradigm.

BRACE and its VALLEY instantiation meet most of the objectives presented. The two notable exceptions are higher order correlations and feedback. A further advantage is that of generality. BRACE may be applied to many different types of clustering and unsupervised learning tasks because of its dynamic Boundary Definition, Boundary Ranking Function, and Classification Optimization Function.

7 FUTURE RESEARCH

Current research focuses on extensions to the basic paradigm and algorithms in order to meet those objectives not met by the paradigm. This includes exploring higher order correlations between attributes, extension to multi-dimensional data, and the use of feedback.

This research was supported in part by grants from Novell, Inc. and WordPerfect Corporation.

References

[1] Catlett, J., "On Changing Continuous Attributes Into Ordered Discrete Attributes", *Lecture Notes in Artificial Intelligence*, Ed. J. Siekmann, Springer-Verlag, Berlin, 1991, pp. 164-78.

[2] Fisher, Douglas H., "Knowledge Acquisition Via Incremental Conceptual Clustering", Readings in Machine Learning, eds. Jude W. Shavlik and Thomas G. Dietterich, Morgan Kaufman Publishers, Inc., San Mateo, California, 1990, pp. 267-283.

[3] Jarvis, R. A. and Patrick, Edward A., "Clustering Using a Similarity Measure Based On Shared Nearest Neighbors", Nearest Neighbor Norms: NN Pattern Classification Techniques, Ed. Belur Dasarathy, IEEE Computer Society Press, Los Alamitos, California, 1991, pp. 388-97.

[4] Kerber, Randy, "ChiMerge: Discretization of Numeric Attributes", *Proceedings of the 10th National Conference on Artificial Intelligence*, 1992, pp. 123-7.

[5] Martinez, Tony R., "Adaptive Self-Organizing Concurrent Systems", *Progress in Neural Networks*, vol. 1, ch. 5, ed. O. Omidvar, Ablex Publishing, 1990, pp. 105-26.

[6] Michalski, Ryszard S., "A Theory and Methodology of Inductive Learning", Readings in Machine Learning, eds. Jude W. Shavlik and Thomas G. Dietterich, Morgan Kaufman Publishers, Inc., San Mateo, California, 1990, pp. 70-95.

[7] Murphy, P. M. and Aha, D. W., *UCI Repository of machine learning databases*, Irvine, CA: University of California, Department of Information and Computer Science, 1992.

[8] Quinlan, J. R., "Induction of Decision Trees", Readings in Machine Learning, eds. Jude W. Shavlik and Thomas G. Dietterich, Morgan Kaufman Publishers, Inc., San Mateo, California, 1990, pp. 57-69.

[9] Tou, J. T. and Gonzalez, R. C., Pattern Recognition Principles, Addison-Wesley Publishing Company, Reading, Massachusetts, 1974.