Discretization of Target Attributes for Subgroup Discovery

Katherine Moreland[1] and Klaus Truemper[2]

[1]The MITRE Corporation, McLean, VA 22102, U.S.A.
[2]Department of Computer Science, University of Texas at Dallas
Richardson, TX 75083, U.S.A.

Abstract. We describe an algorithm called TargetCluster for the discretization of continuous targets in subgroup discovery. The algorithm identifies patterns in the target data and uses them to select the discretization cutpoints. The algorithm has been implemented in a subgroup discovery method. Tests show that the discretization method likely leads to improved insight.

Key words: Subgroup Discovery, Logic, Classification, Feature Selection

1 Introduction

The task of *subgroup discovery* is defined as follows. Given a data set that is representative of a particular population, find all statistically *interesting* subgroups of the data set for a given *target attribute* of interest. Target attributes may be binary, nominal, or continuous. Many subgroup discovery algorithms can handle only binary target attributes, and continuous target attributes must be discretized prior to application of these algorithms. In this paper, we present a new algorithm called TargetCluster for that task as well as a new subgroup discovery method.

There are three main goals of target discretization. First, clusters should be densely populated since then they are likely to represent similar cases. Second, clusters should be clearly distinct since two clusters located close together may actually correspond to a similar target group. Finally, isolated points that do not convincingly fall into a cluster should be effectively skipped since they are unlikely part of an interesting target group.

Let's begin by evaluating how well-suited current approaches for unsupervised discretization meet these goals. The approaches can be classified as being one of three types: 1) equal-width-intervals, 2) equal-frequency-intervals, or 3) clustering. The first two cases simply divide the attribute into a series of equal width or frequency intervals. Consider the following case for a target attribute.

In Figure 1, both the equal-width and equal-frequency methods identify clusters with lower density that are located very close to neighboring clusters. Existing clustering approaches do not satisfy the third goal of target discretization

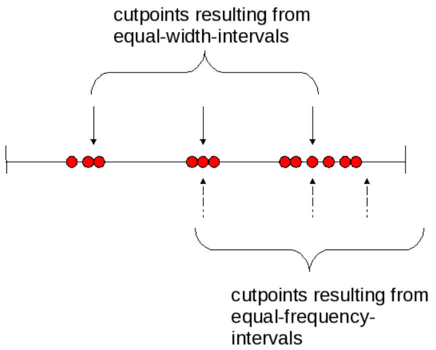

Fig. 1. Example target discretization using equal-width- and equal-frequency-intervals

since they assign all points to clusters with the exception of outliers. We conclude that current unsupervised discretization methods are not well suited for target discretization.

We propose a new method for target discretization that simultaneously achieves the above three goals of target discretization. The algorithm uses a dynamic programming approach. The next section provides details.

2 Solution Algorithm for Cluster Problem

Let the values of the target attribute to be discretized be $v_1, v_2, \dots, v_N$ in ascending order. We separate clusters of these values using intervals containing roughly $\log_2 N$ consecutive values. We say "roughly" since the actual number of values can be somewhat less. We cover details later, when we also motivate the selected size of intervals. Two intervals are *disjoint* if they do not contain a common value. A collection of intervals is *acceptable* if the intervals are pairwise disjoint. Suppose we have a collection of acceptable intervals. If we color the values v_i occurring in the intervals red and the remaining values green, then there are green subsequences separated by red subsequences. The values in each green subsequence define a *cluster*. A collection of intervals is *feasible* if it is acceptable and if each cluster contains at least two values v_i. We want a valuation of feasible collections of intervals. To this end, we define the *width* w_k of the kth cluster, say containing values v_i, v_{i+1}, ..., v_j, to be

$$w_k = v_j - v_i \tag{1}$$

The *score* of a feasible collection c is

$$s_c = \max_k w_k \tag{2}$$

If the score is small, then all clusters have small width, an attractive situation. Let C be the set of feasible c. An *optimal* collection $c^* \in C$ has minimum score among the $c \in C$, and thus is the most attractive one. Therefore, the *score* of c^* is defined via

$$s_{c^*} = \min_c s_c \tag{3}$$

We determine an optimal collection c^* by dynamic programming as follows. For $m \geq 0$ and $1 \leq n \leq N$, we compute optimal solutions for the restricted case of m intervals and for the sequence of values $v_1, v_2, \ldots, v_n$. We denote the optimal score under these restrictions by s_n^m. The stages of the dynamic programming recursion are indexed by m, where $m = 0$ means that no interval is used at all. In a slight change of notation, denote the width of a cluster starting at v_l and terminating at v_r by $w_{l,r}$ Since each cluster of a feasible c contains at least two values v_i, the scores s_n^0 for the stage $m = 0$ are given by

$$s_n^0 = \begin{cases} \infty, & \text{if } n = 1 \\ w_{1,n}, & \text{otherwise} \end{cases} \tag{4}$$

For statement of the recursive formula for any stage $m \geq 1$, let a be any acceptable interval for the sequence $v_1, v_2, \ldots, v_n$. Referring again to the coloring of the values, this time based on the values in a being red and other values being green, define the largest v-value that is smaller than all red values to have index a_l, and define the smallest v-value that is larger than all red values to have index a_r. Differently stated, the green values v_{a_l} and v_{a_r} bracket the red values of the interval a. With these definitions, the recursive equation for stage $m \geq 1$ can be stated as follows.

$$s_n^m = \min_a \max\{s_{a_l}^{m-1}, w_{a_r,n}\} \tag{5}$$

We still must discuss the choice of the size of the intervals. Suppose the interval is chosen to be very small. Then random changes of values may move a value from one cluster across the interval into another cluster, an unattractive situation. We want to choose the interval large enough so that such movement becomes unlikely. In [2], this situation is investigated for the classification of records into two classes. For the case of N records, the conclusion of the reference is that the interval should have roughly $\log_2 N$ data points. In the present context, that size can lead to odd choices, given that values may occur in multiplicities. Using some experimentation, the following modification leads to reasonable results. First, a tolerance ϵ is defined as

$$\epsilon = (v_{a_r} - v_{a_l})/10 \tag{6}$$

Take any value $q \leq N - 1$ such that $p = q - \log_2 N \geq 2$. Declare the values v_i satisfying $v_p + \epsilon \leq v_i \leq v_q - \epsilon$ to be the points of an interval associated with index q. By having q take on all possible values, we obtain all intervals of interest for the optimization problem. There is one more refinement of the method based on experimentation. We do not want to obtain clusters that are not dense, where density of a cluster with values $v_i, v_{i+1}, \ldots, v_j$ is defined to be

$$d_{i,j} = w_{i,j}/(j - i + 1) \tag{7}$$

Specifically, we do not want to have clusters that have less density than the cluster containing the entire sequence $v_1, v_2, \ldots, v_N$. Thus, we declare the width of any cluster whose density $d_{i,j}$ is less than $d_{1,N}$ to be equal to ∞. We omit the simple modification of the above optimization formulas (4) and (5) that accommodate the change.

3 Implementation

We created a subgroup discovery algorithm using the EXARP learning algorithm of [7], which in turn utilizes the Lsquare algorithm of [3, 4, 8]. Each attribute is of one of three possible types: (1) target attribute only, (2) explanation attribute only, or (3) both target and explanation attribute. The goal is to explain the target attributes using the explanation attributes. The explanations correspond to the potentially interesting subgroups for each target. If an attribute is labeled both a target and explanation attribute, it may be explained since it is a target, and it may be part of explanations for other targets.

Subgroup discovery proceeds as follows. The data set is evenly divided into a training and testing set. We set the testing data aside, and for the moment use only the training data. Each target is discretized as described in Section 2. For a given target, this yields j cutpoints with j uncertain intervals, for $j = 1,\ 2,\ \ldots$ For each such j, we use the intervals to partition the data into two sets, A_{train} and B_{train}, in two ways. In the first case, for a given interval, records with a target value below the interval make up the A_{train} set while records with target value above the interval comprise the B_{train} set. If a record's target value falls within the interval, the record is considered uncertain and is not used in computation of explanations. The second case enumerates all pairs of consecutive intervals, k_1 and k_2. For each pair, records with a target value less than interval k_1 or larger than interval k_2 make up the A_{train} set while records with target value larger than interval k_1 and less than interval k_2 make up the B_{train} set. Analogous to the first case, records with a target value that falls within interval k_1 or k_2 are uncertain and not used to compute the explanations. The testing data is partitioned into an A and B set in the same fashion. The EXARP method of [7] is called to compute two DNF logic formulas that separate the A_{train} and B_{train} sets using only the explanation attributes. One formula evaluates to *True* for all records of the A_{train} set while the other evaluates to *True* for records of the B_{train} set. Each clause of the resulting formulas describes a potentially interesting target subgroup.

Using the discretized testing data, we calculate the significance of subgroups somewhat differently from current methods [1, 5, 6]. Suppose a formula evaluates to *True* on the A records and let T be a clause of the formula. We define the subgroup S associated with clause T to be

$$S = \{x \in A \mid \text{clause } T \text{ evaluates to } \mathit{True}\} \tag{8}$$

Let n be the number of records in S. A random process is postulated that constructs records of the A set using the clause T. We skip the details, but mention the following. Let p be the probability that the random process produces at least n records that satisfy the clause. Define q to be the fraction of set B

for which the clause correctly has the value *False*. Then the clause significance is given by

$$s = (1 - p + q)/2 \tag{9}$$

To reduce the number of significant explanations in the output to the user, logically equivalent explanations are grouped together in *equivalence classes*, as follows. Each attribute appearing in a clause is one of four possible types: (1) the attribute has low values (e.g., $x < 2.5$), (2) high values (e.g., $x > 7.5$), (3) low or high values (e.g., $x < 2.5 \;||\; x > 7.5$), or (4) neither low nor high values (e.g., $2.5 < x < 7.5$). Two explanations are logically equivalent if they meet the following criteria: (1) The explanations describe subgroups for the same target attribute, (2) the explanations contain the same attributes, and (3) each attribute must be of the same type in both explanations. The explanation in the equivalence class with the highest significance is chosen as the representative for the class. The representative explanations constitute the interesting subgroups presented to the user.

4 Computational Results

For testing of the TargetCluster algorithm, we use the Abalone, Heart Disease, and Wine sets from the UC Irvine Machine Learning Repository as well as two data sets supplied by H. Bickel, Technical University Munich, Germany, and S. Kümmel, Charité Berlin, Germany. The data set from Technical University Munich concerns Dementia and is called SKT; it contains binary and continuous attributes. The data set from Charité Berlin is called Cervical Cancer and contains only continuous attributes. Thus, we have 5 data sets called Abalone, Cervical Cancer, Heart Disease, SKT, and Wine. Table 1 summarizes the data sets.

Table 1. Summary of Data Sets

	No. of	No. of		Attribute Type	
Data Set	Rec's	Attr's	Target Only	Explanation Only	Both
Abalone	4177	9	number of rings	measurements, gender	none
Cervical Cancer	109	11	none	none	serum levels
Heart Disease	303	14	none	age	patient measurements
SKT	183	20	9 test scores, overall score	patient measurements	none
Wine	178	14	none	none	color intensity, hue, intensity levels

We compare the number and significance of subgroups obtained via TargetCluster discretization with the results found via a combined equal-width-intervals and equal-frequency-intervals method (EWF). In both cases, the discretization data are input to the subgroup discovery method described in Section 3.

The graphs of Figures 2 - 6 illustrate the number of subgroups identified when EWF and TargetCluster are used and different significance thresholds are applied. Except for the Heart Disease case, TargetCluster leads to more subgroups with high significance than EWF.

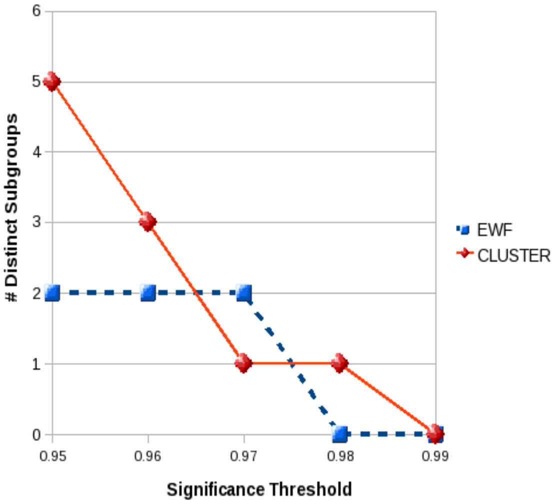

Fig. 2. Abalone Data Set

Table 2 shows the significance value of the highest ranking subgroup produced via each of the discretization methods. In all cases, TargetCluster discretization leads to the discovery of at least one additional subgroup with significance ≥ 0.98. For instance, using the Cervical Cancer data set, 3 subgroups with significance ≥ 0.98 are identified with TargetCluster that are not identified using EWF.

On average, TargetCluster results in subgroups with higher significance than EWF. For the SKT data set, EWF does not result in any subgroup with significance higher than 0.93. However, the TargetCluster method leads to a subgroup with significance of 0.98. On average, TargetCluster results in 2.6 additional subgroups per data set not produced when EWF is used.

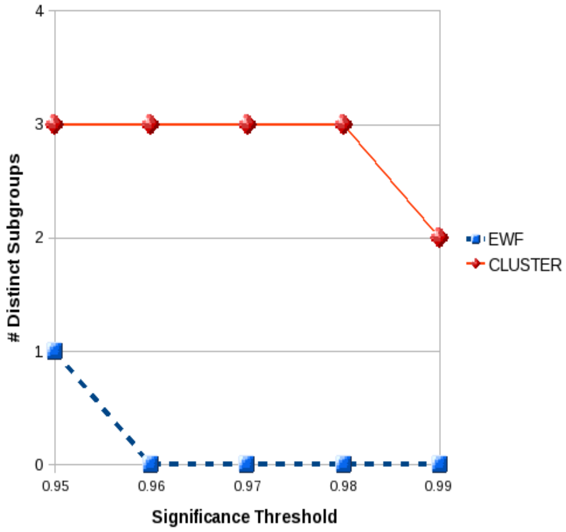

Fig. 3. Cervical Cancer Data Set

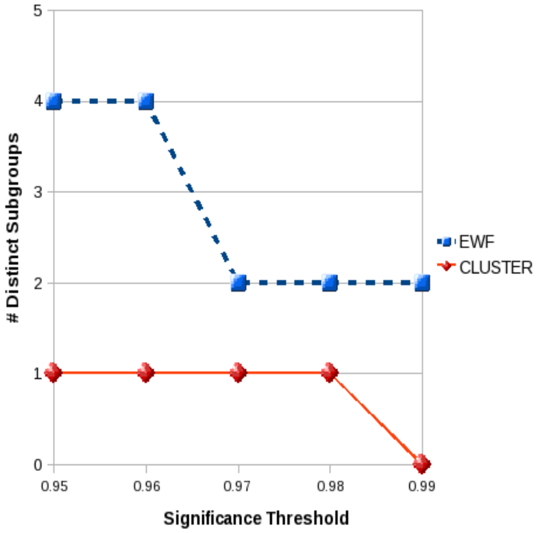

Fig. 4. Heart Disease Data Set

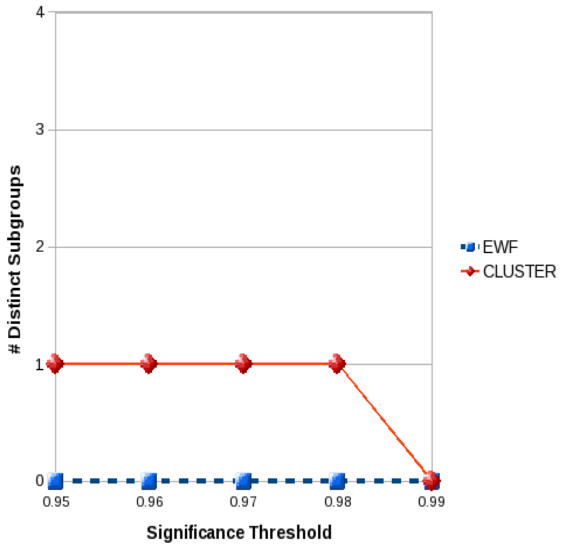

Fig. 5. SKT Data Set

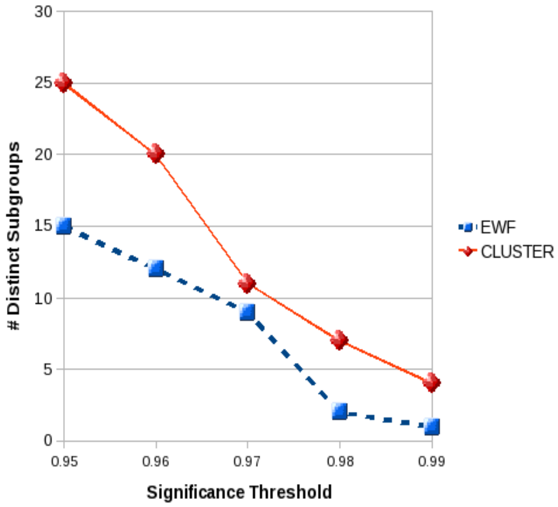

Fig. 6. Wine Data Set

Table 2. TargetCluster vs EWF

	Significance of Highest Ranking Subgroup		Number of Subgroups Identified Significance ≥ 0.98			
Data Set	EWF	Target-Cluster	EWF	Target-Cluster	BOTH	TOTAL
Abalone	0.97	0.98	0	1	0	1
Cervical Cancer	0.95	0.99	0	3	0	3
Heart Disease	0.99	0.98	2	1	0	3
SKT	0.93	0.98	0	1	0	1
Wine	0.99	0.99	2	7	2	11
Average	0.966	0.984	0.8	2.6	0.4	3.8

Based on these findings, the TargetCluster method likely is a useful additional discretization tool for subgroup discovery.

5 Summary

This paper describes the TargetCluster algorithm for discretization of continuous targets in subgroup discovery. The algorithm is based on a clustering approach on the real line. TargetCluster has been combined with a subgroup discovery method. In tests, the use of TargetCluster resulted in a number of additional interesting subgroups when compared with standard unsupervised discretization methods.

References

1. M. Atzmueller, F. Puppe, and H.-P. Buscher. Exploiting background knowledge for knowledge-intensive subgroup discovery. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, 2005.
2. S. Bartnikowski, M. Granberry, J. Mugan, and K. Truemper. Transformation of rational and set data to logic data. In *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques.* Springer, 2006.
3. G. Felici, F. Sun, and K. Truemper. Learning logic formulas and related error distributions. In *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques.* Springer, 2006.
4. G. Felici and K. Truemper. A MINSAT approach for learning in logic domain. *INFORMS Journal of Computing*, 14:20–36, 2002.
5. W. Klösgen. Subgroup discovery. In *Handbook of Data Mining and Knowledge Discovery.* Morgan Kaufmann, 2002.
6. N. Lavrač, P. Flach, B. Kavsek, and L. Todorovski. Adapting classification rule induction to subgroup discovery. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, 2002.
7. K. Riehl and K. Truemper. Construction of deterministic, consisten, and stable explanations from numerical data and prior domain knowledge. Working paper, 2008.
8. K. Truemper. *Design of Logic-based Intelligent Systems.* Wiley, 2004.