A Comparison of Six Approaches to Discretization—A Rough Set Perspective

Piotr Blajdo[1], Jerzy W. Grzymala-Busse[2], Zdzislaw S. Hippe[1], Maksymilian Knap[1], Teresa Mroczek[3], and Lukasz Piatek[3]

[1] Department of Expert Systems and Artificial Intelligence, University of Information Technology and Management, 35-225 Rzeszow, Poland
{pblajdo,zhippe,mknap}@wsiz.rzeszow.pl
[2] Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA
and
Institute of Computer Science, Polish Academy of Sciences, 01-237 Warsaw, Poland
jerzy@ku.edu
[3] Department of Distributed Systems, University of Information Technology and Management, 35-225 Rzeszow, Poland
{tmroczek,lpiatek}@wsiz.rzeszow.pl

Abstract. We present results of extensive experiments performed on nine data sets with numerical attributes using six promising discretization methods. For every method and every data set 30 experiments of ten-fold cross validation were conducted and then means and sample standard deviations were computed. Our results show that for a specific data set it is essential to choose an appropriate discretization method since performance of discretization methods differ significantly. However, in general, among all of these discretization methods there is no statistically significant worst or best method. Thus, in practice, for a given data set the best discretization method should be selected individually.

Keywords: Rough sets, Discretization, Cluster analysis, Merging intervals, Ten-fold cross validation, Test on the difference between means, F-test.

1 Introduction

Many real-life data contain numerical attributes whose values are integers or real numbers. Mining such data sets requires special techniques, taking into account that input data sets are numerical. Most frequently, numerical attributes are converted into symbolic ones during a special process, called discretization [9], before the main process of knowledge acquisition. In some data mining systems both processes, discretization and knowledge acquisition, are executed at the same time. Examples of such systems are C4.5 [14], CART [1], and MLEM2 [10].

Our objective was to compare, for the first time, the most promising discretization techniques [4,9,11,15] through extensive experiments on real-life data sets

G. Wang et al. (Eds.): RSKT 2008, LNAI 5009, pp. 31–38, 2008.

and using ten-fold cross validation, standard statistical test on the difference between means and F-test. The oldest method, among our six approaches to discretization, is based on conditional entropy and was presented in [7]. The remaining five approaches are implemented in the data mining system LERS (Learning from Examples based on Rough Sets) [8]. One of them is based on a hierarchical method of cluster analysis, two on a divisive method of cluster analysis, and remaining two on merging intervals. Our results show that there is no best or worst method. Additionally, for a specific data set, difference in performance between different discretization techniques is significant and the best discretization method should be selected individually, trying as many techniques as possible.

2 Discretization Methods

Knowledge acquisition, such as rule induction or decision tree generation, from data with numerical attributes requires converting numerical values of an attribute into intervals. The process of converting numerical values into intervals, called *discretization*, is usually done as a preprocessing, before the main process of knowledge acquisition. In some data mining systems, such as C4.5, CART, and MLEM2, both processes: discretization and knowledge acquisition are conducted simultaneously. In this paper we will discuss discretization as a preprocessing.

For a numerical attribute a with an interval $[a, b]$ as a range, a partition of the range into n intervals

$$\{[a_0, a_1), [a_1, a_2), ..., [a_{n-2}, a_{n-1}), [a_{n-1}, a_n]\},$$

where $a_0 = a$, $a_n = b$, and $a_i < a_{i+1}$ for $i = 0, 1, ..., n-1$, defines discretization of a. The numbers a_1, a_2,..., a_{n-1} are called *cut-points*.

Discretization methods in which attributes are processed one at a time are called *local* [4,9] (or *static* [5]). On the other hand, if all attributes are considered in selection of the best cut-point, the method is called *global* [4,9] (or *dynamic* [5]). Additionally, if information about the expert's classification of cases is taken into account during the process of discretization, the method is called *supervised* [5].

Many discretization methods [9] are used in data mining. In this paper we will use three approaches to discretization based on cluster analysis, two additional methods that will use similar principles, and, for comparison, a well-known discretization method based on minimal conditional entropy. All of these methods are global and supervised.

The simplest discretization methods are local methods called Equal Interval Width and Equal Frequency per Interval [4,9]. Another local discretization method [7] is called a Minimal Class Entropy. The conditional entropy, defined by a cut-point q that splits the set U of all cases into two sets, S_1 and S_2 is defined as follows

$$E(q, U) = \frac{|S_1|}{|U|} E(S_1) + \frac{|S_2|}{|U|} E(S_2),$$

where $E(S)$ is the entropy of S and $|X|$ denotes the cardinality of the set X. The cut-point q for which the conditional entropy $E(q, U)$ has the smallest value is selected as the best cut-point. If k intervals are required, the procedure is applied recursively $k-1$ times. Let q_1 and q_2 be the best cut-points for sets S_1 and S_2, respectively. If $E(q_1, S_1) > E(q_2, S_2)$ we select q_1 as the next cut-point, if not, we select q_2.

2.1 Globalization of Local Discretization Methods

We will present an approach to convert local discretization methods to global [4]. First, we discretize all attributes, one at a time, selecting the best cut-point for all attributes. If the level of consistency is sufficient, the process is completed. If not, we further discretize, selecting an attribute a for which the following expression has the largest value

$$M_a = \frac{\sum_{B \in \{a\}^*} \frac{|B|}{|U|} E(B)}{|\{a\}^*|}.$$

In all six discretization methods discussed in this paper, the stopping condition was the level of consistency [4], based on rough set theory introduced by Z. Pawlak in [12]. Let U denote the set of all cases of the data set. Let P denote a nonempty subset of the set of all variables, i.e., attributes and a decision. Obviously, set P defines an equivalence relation $\wp$ on U, where two cases x and y from U belong to the same equivalence class of $\wp$ if and only if both x and y are characterized by the same values of each variable from P. The set of all equivalence classes of $\wp$, i.e., a partition on U, will be denoted by P^*.

Equivalence classes of $\wp$ are called *elementary sets* of P. Any finite union of elementary sets of P is called a *definable set* in P. Let X be any subset of U. In general, X is not a definable set in P. However, set X may be approximated by two definable sets in P, the first one is called a *lower approximation of X in P*, denoted by $\underline{P}X$ and defined as follows

$$\bigcup \{Y \in P^* | Y \subseteq X\}.$$

The second set is called an *upper approximation of X in P*, denoted by $\overline{P}X$ and defined as follows

$$\bigcup \{Y \in P^* | Y \cap X \neq \emptyset\}.$$

The lower approximation of X in P is the greatest definable set in P, contained in X. The upper approximation of X in P is the least definable set in P containing X. A *rough set of X* is the family of all subsets of U having the same lower and the same upper approximations of X.

A *level of consistency* [4], denoted L_c, is defined as follows

$$L_c = \frac{\sum_{X \in \{d\}^*} |\underline{A}X|}{|U|}.$$

Practically, the requested level of consistency for discretization is 100%, i.e., we want the discretized data set to be *consistent*.

2.2 Discretization Based on Cluster Analysis and Interval Merging

The data mining system LERS uses two methods of cluster analysis, agglomerative (bottom-up) [4] and divisive (top-down) [13], for discretization. In agglomerative techniques, initially each case is a single cluster, then clusters are fused together, forming larger and larger clusters. In divisive techniques, initially all cases are grouped in one cluster, then this cluster is gradually divided into smaller and smaller clusters. In both methods, during the first step of discretization, *cluster formation*, cases that exhibit the most similarity are fused into clusters. Once this process is completed, clusters are projected on all attributes to determine initial intervals on the domains of the numerical attributes. During the second step (*merging*) adjacent intervals are merged together.

Initially all attributes were categorized into numerical and symbolic. During clustering, symbolic attributes were used only for clustering stopping condition. All numerical attributes were normalized [6] (attribute values were divided by the attribute standard deviation).

In agglomerative discretization method initial clusters were single cases. Then the distance matrix of all Euclidean distances between pairs of cases was computed. The closest two cases, a and b, compose a new cluster $\{a, b\}$. The distance from $\{a, b\}$ to any remaining case c was computed using the Median Cluster Analysis formula [6]:

$$\frac{1}{2}d_{ca} + \frac{1}{2}d_{cb} - \frac{1}{4}d_{ab},$$

where d_{xy} is the Euclidean distance between x and y. The closest two cases compose a new cluster, etc.

At any step of clustering process, the clusters form a partition π on the set of all cases. All symbolic attributes define another partition τ on the set of all cases. The process of forming new clusters was continued as long as $\pi \cdot \tau \leq \{d\}^*$.

In divisive discretization method, initially all cases were placed in one cluster C_1. Next, for every case the average distance from all other cases was computed. The case with the largest average distance was identified, removed from C_1, and placed in a new cluster C_2. For all remaining cases from C_1 a case c with the largest average distance d_1 from all other cases in C_1 was selected and the average distance d_2 from c to all cases in C_2 was computed. If $d_1 - d_2 > 0$, c was removed from C_1 and put to C_2. Then the next case c with the largest average distance in C_1 was chosen and the same procedure was repeated. The process was terminated when $d_1 - d_2 \leq 0$. The partition defined by C_1 and C_2 was checked whether all cases from C_1 were labeled by the same decision value and, similarly, if all cases from C_2 were labeled by the same decision value (though the label for C_1 might be different than the label for C_2). The process of forming new clusters was continued until $\pi \cdot \tau \leq \{d\}^*$.

Final clusters were projected into all numerical attributes, defining this way a set of intervals. The next step of discretization was merging these intervals to reduce the number of intervals and, at the same time, preserve consistency. Merging of intervals begins from *safe merging*, where, for each attribute, neighboring intervals labeled by the same decision value are replaced by their union.

Table 1. Data sets

Data set	Number of		
	cases	attributes	concepts
Australian	690	14	2
Bank	66	5	2
Bupa	345	6	2
German	1000	24	2
Glass	214	9	6
Iris	150	4	3
Segmentation	210	19	7
Wine	178	13	3
Wisconsin	625	9	9

The next step of merging intervals was based on checking every pair of neighboring intervals whether their merging will result in preserving consistency. If so, intervals are merged permanently. If not, they are marked as un-mergeable. Obviously, the order in which pairs of intervals are selected affects the final outcome. In our experiments we started either from an attribute with the most intervals first or from an attribute with the largest conditional entropy.

3 Experiments

Our experiments were conducted on nine data sets, summarized in Table 1. All of these data sets, with the exception of *bank*, are available on the University of California at Irvine *Machine Learning Repository*. The bank data set is a well-known data set used by E. Altman to predict a bankruptcy of companies.

The following six discretization methods were used in our experiments:

- Cluster analysis divisive method with merging intervals with preference for attributes with most intervals, coded as 00,
- Cluster analysis divisive method with merging intervals with preference for attributes with largest conditional entropy, coded as 01,
- Merging intervals with preference for attributes with most intervals, coded as 10,
- Merging intervals with preference for attributes with largest conditional entropy, coded as 11,
- Globalized minimal class entropy method, coded as 13,
- Cluster analysis hierarchical method, coded as 14.

Every discretization method was applied to every data set, with the level of consistency equal to 100%. For any discretized data set, the ten-fold cross

Table 2. Error Rates—Means

Data set	Methods of discretization					
	00	01	10	11	13	14
Australian	16.01	14.62	16.32	14.81	15.51	15.88
Bank	4.70	4.35	3.69	4.29	4.50	3.03
Bupa	36.81	36.82	36.72	37.38	42.73	35.90
German	31.11	31.32	31.12	31.34	29.99	29.30
Glass	31.21	28.54	28.65	28.78	42.63	31.21
Iris	3.26	3.29	3.29	3.31	8.71	4.02
Segmentation	14.67	15.00	16.51	12.87	49.24	17.05
Wine	7.32	7.27	7.21	7.42	2.40	6.26
Wisconsin	21.03	19.45	20.76	19.06	20.87	20.05

validation experiment for determining an error rate was repeated 30 times, with different re-ordering and partitioning the set U of all cases into 10 subsets, where rule sets were induced using the LEM2 algorithm [3,8]. The mean and standard deviation were computed for every sequence of 30 experiments.

Then we used the standard statistical test about the difference between two means, see, e.g., [2]. With the level of significance at 0.05, the decision: reject H_0 if $Z \geq 1.96$ or $Z \leq -1.96$, where H_0 is the hypothesis that the performance of two respective methods do not differ. For example, for *Australian* data set, the value of Z for methods 00 and 01 is 5.55, hence they do differ—as follows from Table 2, method 01 is better (the error rate is smaller). In general, for *Australian* data set, methods 01 and 11 are significantly better than all remaining methods while methods 01 and 11 do not differ significantly. For remaining methods situation is more complicated: methods 13 and 14 do not differ significantly, but method 13 is obviously worse than methods 01 and 11 and is better than methods 00 and 10. Method 14, though worse than 01 and 11, does not differ significantly from methods 00 and 10. On the other hand, methods 00 and 01 do not differ significantly between each other and method 14. A similar analysis was conducted for every data set, the details are skipped because of the page limit for this paper.

As follows from Tables 2–3, performance of discretization methods varies with the change of the data set. The question is if there exists a universally best or worst method. The appropriate test here is the F-test, based on the analysis of variance. The variance s_n^2 of sample means is equal to 6.07. The mean s_d^2 of all sample variances is equal to 178.0. Thus the test statistics F, where

$$F = \frac{s_2^2}{s_d^2}(n)$$

Table 3. Error Rates—Standard Deviations

Data set	Methods of discretization					
	00	01	10	11	13	14
Australian	0.88	1.05	1.02	0.88	0.88	0.78
Bank	1.28	1.24	1.10	1.95	0.63	0.00
Bupa	1.79	1.41	1.43	1.41	2.06	1.83
German	1.15	0.80	0.81	0.90	0.87	0.84
Glass	1.98	2.20	2.10	2.06	2.36	1.42
Iris	0.20	0.17	0.46	0.33	1.05	0.62
Segmentation	1.42	1.35	1.31	1.06	2.22	1.31
Wine	0.98	1.11	1.09	1.13	0.83	1.26
Wisconsin	0.56	0.43	0.55	0.64	0.74	0.41

and $n = 9$ (the number of data sets), is equal to 0.307. Since F is less than 1, we do not need to look to the F-table to know that these discretization methods do not show a statistically significant variation in performance.

4 Conclusions

Our paper presents results of experiments in which six promising discretization methods were used on nine data sets with numerical attributes. All six methods were global and supervised. Results of all six methods, the discretized input data, were used for rule induction using the same LEM2 rule induction algorithm. The performance of discretization, for every method and every data set, was evaluated using 30 experiments of ten-fold cross validation. As a result, we conclude that

- for a specific data set, difference in performance between different discretization methods is significant,
- there is no universally best or worst discretization method. In different words, difference in performance for our six discretization methods, evaluated on all nine data sets, is not significant.

Thus, for a specific data set the best discretization method should be selected individually.

References

1. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees, Wadsworth & Brooks, Monterey, CA (1984)
2. Chao, L.L.: Introduction to Statistics. Brooks Cole Publishing Co., Monterey (1980)

3. Chan, C.C., Grzymala-Busse, J.W.: On the attribute redundancy and the learning programs ID3, PRISM, and LEM2. Department of Computer Science, University of Kansas, TR-91-14 (1991)
4. Chmielewski, M.R., Grzymala-Busse, J.W.: Global discretization of continuous attributes as preprocessing for machine learning. Int. Journal of Approximate Reasoning 15, 319–331 (1996)
5. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: 12-th International Conference on Machine Learning, pp. 194–202. Morgan Kaufmann, San Francisco (1995)
6. Everitt, B.: Cluster Analysis, 2nd edn. Heinmann Educational Books, London (1980)
7. Fayyad, U.M., Irani, K.B.: On the handling of continuous-valued attributes in decision tree generation. Machine Learning 8, 87–102 (1992)
8. Grzymala-Busse, J.W.: A new version of the rule induction system LERS. Fundamenta Informaticae 31, 27–39 (1997)
9. Grzymala-Busse, J.W.: Discretization of numerical attributes. In: Klösgen, W., Zytkow, J. (eds.) Handbook of Data Mining and Knowledge Discovery, pp. 218–225. Oxford University Press, New York (2002)
10. Grzymala-Busse, J.W.: MLEM2: A new algorithm for rule induction from imperfect data. In: 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems ESIA Annecy, France, pp. 243–250 (2002)
11. Grzymala-Busse, J.W., Stefanowski, J.: Three discretization methods for rule induction. Int. Journal of Intelligent Systems. 16, 29–38 (2001)
12. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
13. Peterson, N.: Discretization using divisive cluster analysis and selected post-processing techniques, University of Kansas, Internal Report. Department of Computer Science (1993)
14. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993)
15. Stefanowski, J.: Handling continuous attributes in discovery of strong decision rules. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 394–401. Springer, Heidelberg (1998)