Proceedings of the International Workshop on Rough Sets in Knowledge Discovery (RSKD 2003), associated with the European Joint Conferences on Theory and Practice of Software 2003, Warsaw, Poland, April 5–13, 2003, 132–140.

A Comparison of Three Strategies to Rule Induction from Data with Numerical Attributes

Jerzy W. Grzymala-Busse
Department of Electrical Engineering and Computer Science
University of Kansas, Lawrence, KS 66045, USA
E-mail: Jerzy@ku.edu

Abstract. Our main objective was to compare two discretization techniques, both based on cluster analysis, with a new rule induction algorithm called MLEM2, in which discretization is performed simultaneously with rule induction. The MLEM2 algorithm is an extension of the existing LEM2 rule induction algorithm. The LEM2 algorithm works correctly only for symbolic attributes and is a part of the LERS data mining system. For the two strategies, based on cluster analysis, rules were induced by the LEM2 algorithm. Our results show that MLEM2 outperformed both strategies based on cluster analysis, in terms of complexity (size of rule sets) and, more importantly, error rates.

Keywords: Rough set theory, data mining, machine learning, discretization, rule induction.

Introduction

Many real-life data contain numerical attributes, with values being integers or real numbers. Such numerical values cannot be used in rules induced by data mining systems since there is a very small chance that these values may match values of unseen, testing cases. There are two possible approaches to processing data with numerical attributes: either to convert numerical attributes into intervals through the process called discretization before rule induction or conduct both discretization and rule induction at the same time.

The former approach is more frequently used in practice of data mining. An entire spectrum of discretization algorithms was invented [7]. Using this approach discretization is performed as a preprocessing for the main process of rule induction.

The latter approach was used in a few systems, e.g., in C4.5 that induces decision trees at the same time discretizing numerical attributes [17], in the MODLEM algorithm [9, 10, 18], a modification of LEM2, and in the MLEM2 algorithm. The MLEM2 algorithm [8] is another extension of the existing LEM2 rule induction algorithm. Performance of MLEM2 was compared with MODLEM performance in [8]. The LEM2 algorithm is a part of data mining system LERS (Learning from Examples based on Rough Sets) [5, 6]. LERS uses rough set theory [14, 15] to deal with inconsistency in input data.

As follows from our previous results on melanoma diagnosis [1], discretization based on cluster analysis is a sound approach. For example, discretization algorithms based on divisive and agglomerative methods were ranked as the second and third (out of six) with respect to error rate (with minimal entropy being first), however, an expert in the domain ranked rule sets induced by LEM2 from the data discretized by minimal entropy on the fourth position, while the discretization algorithms based on divisive method of cluster analysis was again second [1]. In our previous research we used only one data set describing melanoma.

Our current objective was to compare two discretization techniques, based on cluster analysis, with a new rule induction algorithm called MLEM2, in which discretization is performed simultaneously with rule induction. As follows from our results, MLEM2 outperformed two other strategies, in which discretization techniques based on cluster analysis were used first and then rule induction was conducted by LEM2 algorithm. Note that MLEM2 produces the same rule sets from symbolic attributes as LEM2.

Discretization algorithms based on cluster analysis

The data mining system LERS uses for discretization a number of discretization algorithms, including two methods of cluster analysis: agglomerative (bottom-up) [3] and divisive (top-down) [16]. In agglomerative techniques, initially each case is a single cluster, then they 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. In the sequel, the former method will be called the agglomerative discretization method, the latter will be called the divisive discretization method. In our experiments, both methods used were *polythetic* (*all* numerical attributes were used).

Initially all attributes were categorized into numerical and symbolic. During clustering, symbolic attributes were used only for clustering stopping condition. First, all numerical attributes were normalized (attribute values were divided by the attribute standard deviation, following [4]).

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 [4]:

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

where d_{xy} id 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 □ on the set of all cases. All symbolic attributes define another partition □on the set of all cases. The set of all concepts define yet another partition □ on the set of all cases. The process of forming new clusters was terminated when □·□> □.

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 stopping condition was the same as for the agglomerative discretization method.

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 provided that the union was a labeled again by the same decision value. 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 from an attribute with the most intervals first.

MLEM2

In general, LERS uses two different approaches to rule induction: one is used in machine learning, the other in knowledge acquisition. In machine learning, or more specifically, in learning from examples (cases), the usual task is to learn *discriminant description* [13], i.e., to learn the smallest set of minimal rules, describing the concept. To accomplish this goal, i.e., to learn discriminant description, LERS uses two algorithms: LEM1 and LEM2 (LEM1 and LEM2 stand for Learning from Examples Module, version 1 and 2, respectively) [5].

Let B be a nonempty lower or upper approximation of a concept represented by a decision-value pair (d, w). Set B *depends* on a set T of attribute-value pairs (a, v) if and only if

$$\emptyset \neq [T] = \bigcap_{(a,v) \in T} [(a, v)] \subseteq B.$$

where $[(a, v)]$ denoted the set of all examples such that for attribute a its values are v.

Set T is a *minimal complex* of B if and only if B depends on T and no proper subset T' of T exists such that B depends on T'. Let $\mathbb{T}$ be a nonempty collection of nonempty sets of attribute-value pairs. Then $\mathbb{T}$ is a *local covering of B* if and only if the following conditions are satisfied:

(1) each member T of $\mathbb{T}$ is a minimal complex of B,

(2) $\bigcup_{T \in \mathbb{T}} [T] = B$, and

(3) $\mathbb{T}$ is minimal, i.e., $\mathbb{T}$ has the smallest possible number of members.

The user may select an option of LEM2 with or without taking into account attribute priorities. The procedure LEM2 with attribute priorities is presented below. The option without taking into account priorities differs from the one presented below in the selection of a pair $t \in T(G)$ in the inner loop WHILE. When LEM2 is not to take attribute priorities into account, the first criterion is ignored. In our experiments all attribute priorities were equal to each other. The user may select an option of LEM2 with or without taking into account attribute priorities. The procedure LEM2 with attribute priorities is presented below. The other option differs from the one presented below in the selection of a pair $t \in T(G)$ in the inner loop WHILE. When LEM2 is not to take attribute priorities into account, the first criterion is ignored. In our experiments all attribute priorities were equal to each other.

```
Procedure LEM2
(input: a set B,
output: a single local covering $\mathbb{T}$ of set B);
begin
    G := B;
    $\mathbb{T}$ := Ø;
    while G ≠ Ø
        begin
        T := Ø;
        T(G) := {t | [t] $\cap$ G ≠ Ø};
        while T = Ø or [T] $\not\subseteq$ B
            begin
               select a pair t $\in$ T(G) with the highest attribute priority, if a tie
               occurs, select a pair t $\in$ T(G) such that |[t] $\cap$ G| is maximum;
               if another tie occurs, select a pair t $\in$ T(G) with the smallest
               cardinality of [t]; if a further tie occurs, select first pair;
             T := T $\cup$ {t};
             G := [t] $\cap$ G;
             T(G) := {t | [t] $\cap$ G ≠ Ø};
             T(G) := T(G) – T;
            end {while}
        for each t in T do
            if [T – {t}] $\subseteq$ B then T := T – {t};
        $\mathbb{T}$ := $\mathbb{T}$ $\cup$ {T};
        G := B – $\bigcup_{T \in \mathbb{T}}$ [T];
    end {while};
    for each T in $\mathbb{T}$ do
        if $\bigcup_{S \in \mathbb{T} - \{T\}}$ [S] = B then $\mathbb{T}$ := $\mathbb{T}$ – {T};
end {procedure}.
```

Rules induced from raw, training data are used for classification of unseen, testing data. The classification system of LERS is a modification of the *bucket brigade algorithm* [2, 12]. The decision to which concept a case belongs is made on the basis of three factors: strength, specificity, and support. They are defined as follows: *Strength* is the total number of cases correctly classified by the rule during training. *Specificity* is the total number of attribute-value pairs on the left-hand side of the rule. The matching rules with a larger number of attribute-value pairs are considered more specific. The third factor, *support*, is defined as the sum of scores of all matching rules from the concept. The concept *C* for which the support (i.e., the sum of all products of strength and specificity, for all rules matching the case, is the largest is a winner and the case is classified as being a member of *C*).

MLEM2, a modified version of LEM2, categorizes all attributes into two categories: numerical attributes and symbolic attributes. For numerical attributes MLEM2 computes blocks in a different way than for symbolic attributes. First, it sorts all values of a numerical attribute. Then it computes cutpoints as averages for any two consecutive values of the sorted list. For each cutpoint *x* MLEM2 creates two blocks, the first block contains all cases for which values of the numerical attribute are smaller than *x*, the second block contains remaining cases, i.e., all cases for which values of the numerical attribute are larger than *x*. The search space of MLEM2 is the set of all blocks computed this way, together with blocks defined by symbolic attributes. Starting from that point, rule induction in MLEM2 is conducted the same way as in LEM2.

Experiments

In our experiments we used eight well-known data sets with numerical attributes (Table 1). All of our data sets, except *Bank*, were obtained from the University of California at Irvine Machine Learning Depository. The *Australian Credit Approval* data set was donated by J. R. Quinlan. The data set *Bank* describing bankruptcy was created by E. Altman and M. Heine at the New York

University School of Business in 1968. The data set *Bupa*, describing liver disorders, contain data gathered by BUPA Medical Research Ltd., England. *German* data set, with only numerical attributes, was donated by H. Hoffman from the University of Hamburg (Germany). The data set *Glass*, representing glass types, was created by B. German, Central Research Establishment, Home Office Forensic Science Service, Canada. The *Iris* data set was created by R. A. Fisher and donated by M. Marshall in 1988. The *Pima* data set describes Pima Indian diabetes and was donated by V. Sigillito in 1990. The data set *Segmentation* created in 1990 by the Vision Group, University of Massachusetts, represents image features: brickface, sky, foliage, cement, window, path, and grass.

Table 1. Data sets

	Number	Number of	Number of
	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
Pima	768	8	2
Segmentation	210	19	7

Table 2 presents error rates for all eight data sets. The error rates were computed using ten-fold cross validation, with the exception of *Bank*, where leaving-one-out was used.

Table 2. Error rates

	Agglomerative	Divisive	MLEM2
	Discretization	Discretization	
	Method	Method	

Australian	28.84	31.01	17.83
Bank	7.58	7.58	4.55
Bupa	41.74	42.03	34.78
German	25.64	26.55	27.09
Glass	30.37	31.78	28.5
Iris	8.0	6.67	4.67
Pima	32.03	32.55	29.3
Segmentation	22.86	19.52	19.52

Table 3 presents the cardinalities of rule sets induced by respective methods.

Table 3. Size of rule sets

	Agglomerative Discretization Method	Divisive Discretization Method	MLEM2
Australian	115	109	90
Bank	4	6	3
Bupa	164	162	71
German	205	232	159
Glass	82	76	30
Iris	13	11	8
Pima	287	264	116
Segmentation	38	35	14

Conclusions

Our main objective was to compare three different strategies for rule induction from data with numerical attributes. In the first two strategies, data with numerical attributes are discretized first, using two different discretization algorithms, based on agglomerative and divisive algorithms of cluster analysis. In the third strategy we used our new algorithm, called MLEM2, an extension of the LEM2 algorithm. The LEM2 algorithm is the most frequently used rule induction option of the LERS data mining system. Results of our experiments are presented in Table 2 and Table 3. In order to rank these three methods we used the Wilcoxon matched-pairs signed rank test, two-tailed, with the significance level 5% [11].

The very first observation is that the rule sets induced by MLEM2 are simpler than rule sets induced by the remaining two strategies (the total number of rules, for any data sets used in our experiments, was always smaller for MLEM2).

Results of the Wilcoxon matched-pairs signed rank test are: the error rate for MLEM2 is significantly smaller than the error rated for the remaining two strategies in which discretization was used as a preprocessing. Also, differences in performance for the two strategies based on cluster analysis discretization as preprocessing, for both complexity (the size of rule sets) and error rate are statistically insignificant.

Our final observation is that MLEM2 induces rules from raw data with numerical attributes, without any prior discretization, and that MLEM2 provides the same results as LEM2 for symbolic attributes. Note that MLEM2 can handle also missing attribute values. A comparison of MLEM2 and other approaches to missing attribute values will be reported in the future.

References

[1] Bajcar, S., Grzymala-Busse, J. W., and Hippe. Z. S. A comparison of six discretization algorithms used for prediction of melanoma. Proceedings of the Eleventh International Symposium on Intelligent Information Systems, IIS'2002, Sopot, Poland, June 3–6, 2002, Physica-Verlag, 2003, 3–12.

[2] Booker, L. B. , Goldberg, D. E., and Holland J. F. Classifier systems and genetic algorithms. In *Machine Learning. Paradigms and Methods*. Carbonell, J. G. (Ed.), The MIT Press, Boston, MA, 1990, 235–282.

[3] Chmielewski, M. R. and Grzymala-Busse, J. W. Global discretization of continuous attributes as preprocessing for machine learning. *Int. Journal of Approximate Reasoning* 15, 1996, 319–331.

[4] Everitt, B. (1980). *Cluster Analysis*. London, United Kingdom: Heinmann Educational Books, Second Edition.

[5] Grzymala-Busse, J. W. LERS—A system for learning from examples based on rough sets. In *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory*. Slowinski, R. (ed.), Kluwer Academic Publishers, Dordrecht, Boston, London, 1992, 3–18.

[6] Grzymala-Busse J. W. A new version of the rule induction system LERS. *Fundamenta Informaticae* 31 (1997), 27–39.

[7] Grzymala-Busse, J. W. Discretization of numerical attributes. In *Handbook of Data Mining and Knowledge Discovery*, ed. by W. Klösgen and J. Zytkow, Oxford University Press, 2002, 218–225.

[8] Grzymala-Busse, J. W. MLEM2: A new algorithm for rule induction from imperfect data. Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002, July 1–5, Annecy, France, 243–250.

[9] Grzymala-Busse, J. W. and Stefanowski, J. Discretization of numerical attributes by direct use of the rule induction algorithm LEM2 with interval extension. Proc. of the Sixth Symposium on Intelligent Information Systems (IIS'97), Zakopane, Poland, June 9–13, 1997, 149–158.

[10] Grzymala-Busse, J. W. and Stefanowski, J. Three discretization methods for rule induction. *International Journal of Intelligent Systems* 16 (2001), 29–38.

[11] Hamburg, M.: *Statistical Analysis for Decision Making*. New York, NY: Harcourt Brace Jovanovich, Inc. (1983) 546–550, 721.

[12] Holland, J. H., Holyoak, K. J., and Nisbett, R. E. *Induction. Processes of Inference, Learning, and Discovery.* The MIT Press, Boston, MA, 1986.

[13] Michalski, R. S. A Theory and Methodology of Inductive Learning. In: Michalski, R. S., Carbonell, J. G., Mitchell T. M. (eds.): *Machine Learning. An Artificial Intelligence Approach*, Morgan Kauffman (1983) 83–134.

[14] Pawlak, Z. Rough Sets. *International Journal of Computer and Information Sciences*, 11, 1982, 341–356.

[15] Pawlak, Z. Rough Sets. *Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, Dordrecht, Boston, London, 1991.

[16] Peterson, N.: Discretization using divisive cluster analysis and selected post-processing techniques. Department of Computer Science, University of Kansas, internal report, 1993.

[17] Quinlan, J. R. *C*4.5: *Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[18] Stefanowski, J. On rough set based approaches to induction of decision rules. In Polkowski L., Skowron A. (eds.) *Rough Sets in Data Mining and Knowledge Discovery*. Physica Verlag, Heidelberg New York (1998) 500–529