Mining Numerical Data—A Rough Set Approach

Jerzy W. Grzymala-Busse

[1] Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA
[2] Institute of Computer Science Polish Academy of Sciences, 01-237 Warsaw, Poland
jerzy@ku.edu
http://lightning.eecs.ku.edu/index.html

Abstract. We present an approach to mining numerical data based on rough set theory using calculus of attribute-value blocks. An algorithm implementing these ideas, called MLEM2, induces high quality rules in terms of both simplicity (number of rules and total number of conditions) and accuracy. Additionally, MLEM2 induces rules not only from complete data sets but also from data with missing attribute values, with or without numerical attributes.

1 Introduction

For knowledge acquisition (or data mining) from data with numerical attributes special techniques are applied [13]. Most frequently, an additional step, taken before the main step of rule induction or decision tree generation and called *discretization* is used. In this preliminary step numerical data are converted into symbolic or, more precisely, a domain of the numerical attribute is partitioned into intervals. Many discretization techniques, using principles such as equal interval frequency, equal interval width, minimal class entropy, minimum description length, clustering, etc., were explored, e.g., in [1,2,3,5,6,8,9,10,19,21,22,23,24,27]. Note that discretization used as preprocessing and based on clustering is superior to other preprocessing techniques of this type [8].

Discretization algorithms which operate on the set of all attributes and which do not use information about decision (concept membership) are called *unsupervised*, as opposed to *supervised*, where the decision is taken into account [9]. Methods processing the entire attribute set are called *global*, while methods working on one attribute at a time are called *local* [8]. In all of these methods discretization is a preprocessing step and is undertaken before the main process of knowledge acquisition.

Another possibility is to discretize numerical attributes during the process of knowledge acquisition. Examples of such methods are MLEM2 [14] and MODLEM [20,29,30] for rule induction and C4.5 [28] and CART [4] for decision tree generation. These algorithms deal with original, numerical data and the process of knowledge acquisition and discretization are conducted at the same time. The

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 12–21, 2007.

MLEM2 algorithm produces better rule sets, in terms of both simplicity and accuracy, than clustering methods [15]. However, discretization is an art rather than a science, and for a specific data set it is advantageous to use as many discretization algorithms as possible and then select the best approach.

In this paper we will present the MLEM2 algorithm, one of the most successful approaches to mining numerical data. This algorithm uses rough set theory and calculus of attribute-value pair blocks. A similar approach is represented by MODLEM. Both MLEM2 and MODLEM algorithms are outgrowths of the LEM2 algorithm. However, in MODLEM the most essential part of selecting the best attribute-value pair is conducted using entropy or Laplacian conditions, while in MLEM2 this selection uses the most relevance condition, just like in the original LEM2.

2 MLEM2

The MLEM2 algorithm is a part of the LERS (Learning from Examples based on Rough Sets) data mining system. Rough set theory was initiated by Z. Pawlak [25,26]. 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 the smallest set of minimal rules, describing the concept. To accomplish this goal, LERS uses two algorithms: LEM1 and LEM2 (LEM1 and LEM2 stand for Learning from Examples Module, version 1 and 2, respectively) [7,11,12].

The LEM2 algorithm is based on an idea of an attribute-value pair block. For an attribute-value pair $(a,\ v) = t$, a *block* of t, denoted by $[t]$, is a set of all cases from U such that for attribute a have value v. For a set T of attribute-value pairs, the intersection of blocks for all t from T will be denoted by $[T]$. 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 $t = (a, v)$ if and only if

$$\emptyset \neq [T] = \bigcap_{t \in T} [t] \subseteq B.$$

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 $\mathcal{T}$ be a nonempty collection of nonempty sets of attribute-value pairs. Then $\mathcal{T}$ is a *local covering* of B if and only if the following conditions are satisfied:

- each member T of $\mathcal{T}$ is a minimal complex of B,
- $\bigcap_{t \in \mathcal{T}} [T] = B$, and
- $\mathcal{T}$ is minimal, i.e., $\mathcal{T}$ 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.

Procedure LEM2
(**input**: a set B,
output: a single local covering $\mathcal{T}$ of set B);
begin
    $G := B$;
    $\mathcal{T} := \emptyset$;
    while $G \neq \emptyset$
        begin
        $T := \emptyset$;
        $T(G) := \{t | [t] \cap G \neq \emptyset\}$;
        while $T = \emptyset$ **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;
                $T := T \cup \{t\}$;
                $G := [t] \cap G$;
                $T(G) := \{t | [t] \cap G \neq \emptyset\}$;
                $T(G) := T(G) - T$;
                end {while}
            for each $t \in T$ **do**
                if $[T - \{t\}] \subseteq$ B **then** $T := T - \{t\}$;
            $\mathcal{T} := \mathcal{T} \cup \{T\}$;
            $G := B - \bigcup_{T \in \mathcal{T}} [T]$;
    end {while};
    for each $T \in \mathcal{T}$ **do**
        if $\bigcup_{S \in \mathcal{T} - \{T\}} [S] = B$ **then** $\mathcal{T} := \mathcal{T} - \{T\}$;
end {procedure}.

For a set X, $|X|$ denotes the cardinality of X.

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*. 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.

Let us illustrate the MLEM2 algorithm using the following example from Table 1.

Table 1. An example of the decision table

	Attributes		Decision
Case	Gender	Cholesterol	Stroke
1	man	180	no
2	man	240	yes
3	man	280	yes
4	woman	240	no
5	woman	280	no
6	woman	320	yes

Rows of the decision table represent *cases*, while columns are labeled by *variables*. The set of all cases will be denoted by U. In Table 1, $U = \{1, 2, ..., 6\}$. Independent variables are called *attributes* and a dependent variable is called a *decision* and is denoted by d. The set of all attributes will be denoted by A. In Table 1, $A = \{Gender, Cholesterol\}$. Any decision table defines a function ρ that maps the direct product of U and A into the set of all values. For example, in Table 1, $\rho(1, Gender) = man$. The decision table from Table 1 is *consistent*, i.e., there are no conflicting cases in which all attribute values are identical yet the decision values are different. Subsets of U with the same decision value are called *concepts*. In Table 1 there are two concepts: $\{1, 4, 5\}$ and $\{2, 3, 6\}$.

Table 1 contains one numerical attribute (*Cholesterol*). The sorted list of values of *Cholesterol* is 180, 240, 280, 320. The corresponding cutpoints are: 210, 260, 300.

Since our decision table is consistent, input sets to be applied to MLEM2 are concepts. The search space for MLEM2 is the set of all blocks for all possible attribute-value pairs $(a, v) = t$. For Table 1, the set of all attribute-value pair blocks are

$[(Gender, man)] = \{1, 2, 3\}$,
$[(Gender, woman)] = \{4, 5, 6\}$,
$[(Cholesterol, 180..210)] = \{1\}$,
$[(Cholesterol, 210..320)] = \{2, 3, 4, 5, 6\}$,
$[(Cholesterol, 180..260)] = \{1, 2, 4\}$,
$[(Cholesterol, 260..320)] = \{3, 5, 6\}$,
$[(Cholesterol, 180..300)] = \{1, 2, 3, 4, 5\}$,
$[(Cholesterol, 300..320)] = \{6\}$.

Let us start running MLEM2 for the concept {1, 4, 5}. Thus, initially this concept is equal to B (and to G). The set $T(G)$ is equal to {(Gender, man), (Gender, woman), (Cholesterol, 180..210), (Cholesterol, 210..320), (Cholesterol, 180..260), (Cholesterol, 260..320), (Cholesterol, 180..300)}.

For the attribute-value pair (Cholesterol, 180..300) from $T(G)$ the following value $|[(attribute, value)] \cap G|$ is maximum. Thus we select our first attribute-value pair t = (Cholesterol, 180..300). Since $[(Cholesterol, 180..300)] \not\subseteq B$, we have to perform the next iteration of the inner WHILE loop. This time $T(G)$ = {(Gender, man), (Gender, woman), (Cholesterol, 180..210), (Cholesterol, 210..320), (Cholesterol, 180..260), (Cholesterol, 260..320)}. For three attribute-value pairs from $T(G)$: (Gender, woman), (Cholesterol, 210..320) and (Cholesterol, 180..260) the value of $|[(attribute, value)] \cap G|$ is maximum (and equal to two). The second criterion, the smallest cardinality of [(attribute, value)], indicates (Gender, woman) and (Cholesterol, 180..260) (in both cases that cardinality is equal to three). The last criterion, "first pair", selects (Gender, woman). Moreover, the new T = {(Cholesterol, 180..300), (Gender, woman)} and new G is equal to {4, 5}. Since $[T] = [(Cholesterol, 180..260] \cap [(Gender, woman)] = \{4, 5\} \subseteq B$, the first minimal complex is computed.

Furthermore, we cannot drop any of these two attribute-value pairs, so $\mathcal{T} = \{T\}$, and the new G is equal to $B - \{4, 5\} = \{1\}$.

During the second iteration of the outer WHILE loop, the next minimal complex T is identified as {(Cholesterol, 180..210)}, so $\mathcal{T}$ = {{(Cholesterol, 180..300), (Gender, woman)}, {(Cholesterol, 180..210)}} and $G = \emptyset$.

The remaining rule set, for the concept {2, 3, 6} is induced in a similar manner. Eventually, rules in the LERS format (every rule is equipped with three numbers, the total number of attribute-value pairs on the left-hand side of the rule, the total number of examples correctly classified by the rule during training, and the total number of training cases matching the left-hand side of the rule) are:

2, 2, 2
(Gender, woman) & (Cholesterol, 180..300) -> (Stroke, no)
1, 1, 1
(Cholesterol, 180..210) -> (Stroke, no)

2, 2, 2
(Gender, man) & (Cholesterol, 210..320) -> (Stroke, yes)
1, 1, 1
(Cholesterol, 300..320) -> (Stroke, yes)

3 Numerical and Incomplete Data

Input data for data mining are frequently affected by missing attribute values. In other words, the corresponding function ρ is incompletely specified (partial). A decision table with an incompletely specified function ρ will be called *incompletely specified*, or *incomplete*.

Though four different interpretations of missing attribute values were studied [18]; in this paper, for simplicity, we will consider only two: lost values (the values that were recorded but currently are unavailable) and "do not care" conditions (the original values were irrelevant).

For the rest of the paper we will assume that all decision values are specified, i.e., they are not missing. Also, we will assume that all missing attribute values are denoted either by "?" or by "$*$", lost values will be denoted by "?", "do not care" conditions will be denoted by "$*$". Additionally, we will assume that for each case at least one attribute value is specified.

Incomplete decision tables are described by characteristic relations instead of indiscernibility relations. Also, elementary blocks are replaced by characteristic sets, see, e.g., [16,17,18]. An example of an incomplete table is presented in Table 2.

Table 2. An example of the incomplete decision table

	Attributes		Decision
Case	Gender	Cholesterol	Stroke
1	?	180	no
2	man	*	yes
3	man	280	yes
4	woman	240	no
5	woman	?	no
6	woman	320	yes

For incomplete decision tables the definition of a block of an attribute-value pair must be modified. If for an attribute a there exists a case x such that $\rho(x, a) = ?$, i.e., the corresponding value is lost, then the case x is not included in the block $[(a, v)]$ for any value v of attribute a. If for an attribute a there exists a case x such that the corresponding value is a "do not care" condition, i.e., $\rho(x, a) = *$, then the corresponding case x should be included in blocks

$[(a, v)]$ for all values v of attribute a. This modification of the definition of the block of attribute-value pair is consistent with the interpretation of missing attribute values, lost and "do not care" condition. Numerical attributes should be treated in a little bit different way as symbolic attributes. First, for computing characteristic sets, numerical attributes should be considered as symbolic. For example, for Table 2 the blocks of attribute-value pairs are:

$[(Gender, man)] = \{2, 3\}$,
$[(Gender, woman)] = \{4, 5, 6\}$,
$[(Cholesterol, 180)] = \{1, 2\}$,
$[(Cholesterol, 240)] = \{2, 4\}$,
$[(Cholesterol, 280)] = \{2, 3\}$,
$[(Cholesterol, 320)] = \{2, 6\}$.

The *characteristic set* $K_B(x)$ is the intersection of blocks of attribute-value pairs (a, v) for all attributes a from B for which $\rho(x, a)$ is specified and $\rho(x, a) = v$. The characteristic sets $K_B(x)$ for Table 2 and $B = A$ are:

$K_A(1) = U \cap \{1, 2\} = \{1, 2\}$,
$K_A(2) = \{2, 3\} \cap U = \{2, 3\}$,
$K_A(3) = \{2, 3\} \cap \{2, 3\} = \{2, 3\}$,
$K_A(4) = \{4, 5, 6\} \cap \{2, 4\} = \{4\}$,
$K_A(5) = \{4, 5, 6\} \cap U = \{4, 5, 6\}$,
$K_A(6) = \{4, 5, 6\} \cap \{2, 6\} = \{6\}$.

For incompletely specified decision tables lower and upper approximations may be defined in a few different ways [16,17,18]. We will quote only one type of approximations for incomplete decision tables, called concept approximations. A *concept* B-lower approximation of the concept X is defined as follows:

$$\underline{B}X = \cup\{K_B(x) | x \in X, K_B(x) \subseteq X\}.$$

A concept B-upper approximation of the concept X is defined as follows:

$$\overline{B}X = \cup\{K_B(x) | x \in X, K_B(x) \cap X \neq \emptyset\} = \cup\{K_B(x) | x \in X\}.$$

For Table 2, concept lower and upper approximations are:

$$\underline{A}\{1, 4, 5\} = \{4\},$$

$$\underline{A}\{2, 3, 6\} = \{2, 3, 6\},$$

$$\overline{A}\{1, 4, 5\} = \{1, 2, 4, 5, 6\},$$

$$\overline{A}\{2, 3, 6\} = \{2, 3, 6\}.$$

For inducing rules from data with numerical attributes, blocks of attribute-value pairs are defined differently than in computing characteristic sets. Blocks of attribute-value pairs for numerical attributes are computed in a similar way as for complete data, but for every cutpoint the corresponding blocks are computed taking into account interpretation of missing attribute values. Thus,

$[(Gender, man)] = \{1, 2\}$,
$[(Gender, woman)] = \{4, 5, 6\}$,
$[(Cholesterol, 180..210)] = \{1, 2\}$,
$[(Cholesterol, 210..320)] = \{2, 3, 4, 6\}$,
$[(Cholesterol, 180..260)] = \{1, 2, 4\}$,
$[(Cholesterol, 260..320)] = \{2, 3, 6\}$,
$[(Cholesterol, 180..300)] = \{1, 2, 3, 4\}$,
$[(Cholesterol, 300..320)] = \{2, 6\}$.

Using the MLEM2 algorithm, the following rules are induced:

certain rule set (induced from the concept lower approximations):

2, 1, 1
(Gender, woman) & (Cholesterol, 180..260) -> (Stroke, no)
1, 3, 3
(Cholesterol, 260..320) -> (Stroke, yes)

possible rule set (induced from the concept upper approximations):

1, 2, 3
(Gender, woman) -> (Stroke, no)
1, 1, 3
(Cholesterol, 180..260) -> (Stroke, no)
1, 3, 3
(Cholesterol, 260..320) -> (Stroke, yes)

4 Conclusions

We demonstrated that both rough set theory and calculus of attribute-value pair blocks are useful tools for data mining from numerical data. The same idea of an attribute-value pair block may be used in the process of data mining not only for computing elementary sets (for complete data sets) but also for rule induction. The MLEM2 algorithm induces rules from raw data with numerical attributes, without any prior discretization, and MLEM2 provides the same results as LEM2 for data with all symbolic attributes. Additionally, experimental results show that rule induction based on MLEM2 is one of the best approaches to data mining from numerical data [15].

References

1. Bajcar, S., Grzymala-Busse, J.W., Hippe, Z.S.: A comparison of six discretization algorithms used for prediction of melanoma. In: Proc. of the Eleventh International Symposium on Intelligent Information Systems, IIS'2002, Sopot, Poland, 2002, pp. 3–12. Physica-Verlag, Heidelberg (2003)

2. Bay, S.D.: Multivariate discretization of continous variables for set mining. In: Proc. of the 6-th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Boston, MA, 2000, pp. 315–319 (2000)
3. Biba, M., Esposito, F., Ferilli, S., Mauro, N.D., Basile, T.M.A.: Unsupervised discretization using kernel density estimation. In: Proc. of the 20-th Int. Conf. on AI, Hyderabad, India, 2007, pp. 696–701 (2007)
4. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth & Brooks, Monterey CA (1984)
5. Catlett, J.: On changing continuous attributes into ordered discrete attributes. In: EWSL-91, Proc. of the European Working Session on Learning, Porto, Portugal, March 1991. LNCS (LNAI), pp. 164–178. Springer, Berlin (1991)
6. Chan, C.C., Batur, C., Srinivasan, A.: Determination of quantization intervals in rule based model for dynamic systems. In: Proc. of the IEEE Conference on Systems, Man, and Cybernetics, Charlottesville, VA, 1991, pp. 1719–1723 (1991)
7. 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, December 1991, p. 20 (1991)
8. 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)
9. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: Proc of the 12-th Int. Conf. on Machine Learning, Tahoe City, CA, July 9–12, 1995, pp. 194–202 (1995)
10. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proc. of the 13th Int. Joint Conference on AI, Chambery, France, 1993, pp. 1022–1027 (1993)
11. Grzymala-Busse, J.W.: LERS—A system for learning from examples based on rough sets. In: Slowinski, R. (ed.) Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory, pp. 3–18. Kluwer Academic Publishers, Dordrecht (1992)
12. Grzymala-Busse, J.W.: A new version of the rule induction system LERS. Fundamenta Informaticae 31, 27–39 (1997)
13. 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)
14. Grzymala-Busse, J.W.: MLEM2: A new algorithm for rule induction from imperfect data. In: Proc. of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002, Annecy, France, 2002, pp. 243–250 (2002)
15. Grzymala-Busse, J.W.: A comparison of three strategies to rule induction from data with numerical attributes. In: Proc. of the Int. Workshop on Rough Sets in Knowledge Discovery (RSKD 2003), in conjunction with the European Joint Conferences on Theory and Practice of Software, Warsaw, 2003, pp. 132–140 (2003)
16. Grzymala-Busse, J.W.: Rough set strategies to data with missing attribute values. Workshop Notes, Foundations and New Directions of Data Mining. In: conjunction with the 3-rd International Conference on Data Mining, Melbourne, FL, 2003, pp. 56–63 (2003)
17. Grzymala-Busse, J.W.: Data with missing attribute values: Generalization of idiscernibility relation and rule induction. In: Transactions on Rough Sets. Lecture Notes in Computer Science Journal Subline, vol. 1, pp. 78–95. Springer, Heidelberg (2004)

18. Grzymala-Busse, J.W.: Incomplete data and generalization of indiscernibility relation, definability, and approximations. In: Proc. of the RSFDGrC'2005, the Tenth International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, pp. 244–253. Springer, Regina, Canada (2005)
19. Grzymala-Busse, J.W., Stefanowski, J.: Discretization of numerical attributes by direct use of the rule induction algorithm LEM2 with interval extension. In: Proc. of the Sixth Symposium on Intelligent Information Systems (IIS'97), Zakopane, Poland, 1997, pp. 149–158 (1997)
20. Grzymala-Busse, J.W., Stefanowski, J.: Three discretization methods for rule induction. Int. Journal of Intelligent Systems 16, 29–38 (2001)
21. Kerber, R.: ChiMerge: Discretization of numeric attributes. In: Proc. of the 10th National Conf. on AI, San Jose, CA, 1992, pp. 123–128 (1992)
22. Kohavi, R., Sahami, M.: Error-based and entropy-based discretization of continous features. In: Proc of the 2-nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR, 1996, pp. 114–119 (1996)
23. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: An enabling technique. Data. Mining and Knowledge Discovery 6, 393–423 (2002)
24. Nguyen, H.S., Nguyen, S.H.: Discretization methods for data mining. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, pp. 451–482. Physica-Verlag, Heidelberg (1998)
25. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
26. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
27. Pensa, R.G, Leschi, C., Besson, J., Boulicaut, J.F.: Assessment of discretization techniques for relevant pattern discovery from gene expression data. In: Proc. of the 4-th ACM SIGKDD Workshop on Data Mining in Bioinformatics, 2004, pp. 24–30 (2004)
28. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA (1993)
29. Stefanowski, J.: Handling continuous attributes in discovery of strong decision rules. In: Proc. of the 1-st Int. Conference on Rough Sets and Current Trends in Computing, Warsaw, pp. 394–401. Springer, Berlin (1998)
30. Stefanowski, J.: Algorithms of Decision Rule Induction in Data Mining. Poznan University of Technology Press, Poznan, Poland (2001)