Journal of Intelligent Manufacturing, 16, 565–573, 2005

A comparison of two approaches to data mining from imbalanced data

JERZY W. GRZYMALA-BUSSE[1,2], JERZY STEFANOWSKI[3] and SZYMON WILK[3]

[1]Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA
E-mail: Jerzy@ku.edu
[2]Institute of Computer Science, Polish Academy of Sciences, 01-237 Warsaw, Poland
[3]Institute of Computing Science, Poznan University of Technology, 60-965 Poznan, Poland
E-mails: Szymon.Wilk@cs.put.poznan.pl, Jerzy.Stefanowski@cs.put.poznan.pl

Received May 2004 and accepted November 2004

Our objective is a comparison of two data mining approaches to dealing with imbalanced data sets. The first approach is based on saving the original rule set, induced by the LEM2 (Learning from Example Module) algorithm, and changing the rule strength for all rules for the smaller class (concept) during classification. In the second approach, rule induction is split: the rule set for the larger class is induced by LEM2, while the rule set for the smaller class is induced by EXPLORE, another data mining algorithm. Results of our experiments show that both approaches increase the sensitivity compared to the original LEM2. However, the difference in performance of both approaches is statistically insignificant. Thus the appropriate approach for dealing with imbalanced data sets should be selected individually for a specific data set.

Keywords: Data mining, EXPLORE rule induction algorithm, imbalanced data sets, LEM2 rule induction algorithm

1. Introduction

During data mining from real-life data, sizes of classes (concepts) are frequently different. Quite often the class which is critical from the domain point of view (the primary class) includes a much smaller number of cases while other (secondary) classes form the majority of cases (Japkowicz, 2000). Such data sets are called imbalanced. This situation is typical in medical problems, where the task is to diagnose a specific disease. The primary class usually describes patients requiring special attention while all remaining cases are members of the secondary class (e.g., healthy patients). Similar situations also occur in other domains, e.g., in financial analysis of loan policy or bankruptcy.

Recently we observe an increase of research activity in data mining from imbalanced data sets. For example, the Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining, *SIGKDD Explorations*, published a special issue on Learning From Imbalanced Data Sets (June 2004).

Standard classifiers derived from such data sets are affected by a lack of balance. That is, their predictive accuracy is biased towards majority classes and they usually have difficulties with correct classification of objects from the primary classes. Since the primary class is more important, costs of false positives and false negatives may drastically differ. Using again an example of medical diagnosis, the total classification accuracy is misleading as an indicator of the classifier quality for imbalanced data. Diagnosis is characterized by sensitivity (the conditional probability of the set of correctly classified cases from the primary class, given the primary class) and by specificity (the conditional probability of the set of correctly recognized cases from the secondary class, given the secondary class). In such applications more attention is given to sensitivity than to specificity.

In our research, we tested two approaches to increasing the sensitivity of the primary class for rule-based classifiers. In both approaches, initial rules were induced by the LEM2 (Learning from Examples Module) algorithm. An original version of LEM2 induces a minimal set of rules from rough approximations of classes (Grzymala-Busse, 1992). Generated rules are then used by the Learning from Examples-based on Rough Sets (LERS) "bucket brigade" classification strategy. The first technique to improve sensitivity is based on increasing strengths of rules describing the primary class. The rule strength is defined as the number of training cases correctly classified by the rule. The idea is to multiple the strengths of all primary class rules by the same real number, called strength multiplier, while not changing the strength of rules from the secondary classes. As a result, during classification of new objects, such primary class rules have an increased chance to classify these objects as being members of the primary class. (Grzymala-Busse *et al.*, 2000)

The second technique is based on a different principle. A minimal set of rules for the primary class is replaced by a new set of rules, with the strength greater than a certain threshold. Such rules are discovered by a special algorithm, called EXPLORE (Stefanowski and Vaderpooten, 2001). If the strength threshold is sufficiently low, then EXPLORE may generate much more rules then LEM2. Thus, by using such rules for the primary class, while preserving the original set of rules for the secondary class, the chance that a case from the primary class is selected by a classifier is increased and sensitivity should improve.

The main aim of this study is to evaluate the performance of both techniques on several imbalanced data sets. Moreover, we compare both techniques using a standard scheme of applying LEM2 with LERS classification strategy.

The paper is organized as follows. We begin with a brief description of the LEM2 rule induction algorithm, which is a basis of the first approach and partially employed by the second approach. The LERS classification system is described later. The next section is devoted to explaining sensitivity and specificity. Then we present both approaches to dealing with imbalanced data sets. Finally, we quote results of our experiments and conclusions.

A preliminary version of this paper was presented at the eighth International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Wellington, New Zealand, September 20–24, 2004 (Grzymala-Busse *et al.*, 2004).

2. Rule induction with LEM2

Both presented approaches to some extent employ the LEM2 algorithm which uses rough set theory for inconsistent data. In rough set theory (Pawlak, 1982, 1991; Pawlak *et al.*, 1995) inconsistencies are not removed from consideration. Instead, lower and upper approximations of the concept are computed. On the basis of these approximations, two corresponding sets of rules: certain and possible, are induced.

The following is a summary of the main ideas of the LEM2 algorithm. The LEM2 is a component of the LERS data mining system (Grzymala-Busse, 1992, 1997). A block of an attribute-value pair $t=(a,v)$, denoted $[t]$, is the set of all examples that for attribute a have value v. A concept, described by the value w of decision d, is denoted $[(d, w)]$, and it is the set of all examples that have value w for decision d. Let C be a concept and let T be a set of attribute-value pairs. Concept C *depends on* a set T if and only if

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

Set T is a *minimal complex* of concept C if and only if C depends on T and T is minimal. Let T be a nonempty collection of nonempty sets of attribute-value pairs. Set T is a *local covering of*

C if and only if the following three conditions are satisfied:

(1) each member of T is a minimal complex of *C*,
(2) $\bigcup_{T \in \mathbf{T}}[T] = B$

and

(3) **T** is minimal, i.e., **T** has the smallest possible number of members.

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

For a set X, the cardinality of X is denoted by |X|.

For each concept *C*, the LEM2 algorithm induces rules by computing a local covering T. Any set *T*, a minimal complex which is a member of T, is computed from attribute-value pairs selected from the set $T(G)$ of attribute-value pairs relevant with a current goal *G*, i.e., pairs whose blocks have nonempty intersection with *G*. The initial goal *G* is equal to the concept and then it is iteratively updated by subtracting from *G* the set of examples described by the set of minimal complexes computed so far. Attribute-value pairs from T which are selected as the most relevant, i.e., on the basis of maximum of the cardinality of $[t] \cap G$, if a tie occurs, on the basis of the smallest cardinality of $[t]$. The last condition is equivalent to the maximal conditional probability of goal *G* given attribute-value pair *t*.

3. Classification of unseen cases

In our experiments, we used the LERS classification system. For classification of unseen cases system LERS employees a modified "bucket brigade algorithm" (Booker *et al.*, 1990; Holland *et al.*, 1986). In this approach, the decision to which concept a case belongs is made using two factors: *strength* and *support*. They are defined as follows: *Strength factor* is a measure of how well the rule has performed during training. The second factor, *support*, is related to a concept and is defined as the sum of scores of all matching rules from the concept. The concept getting the largest support wins the contest.

In LERS, the strength factor is adjusted to be the *strength* of a rule, i.e., the total number of examples correctly classified by the rule during training. The concept *C* for which the support, i.e., the following expression

$$\sum_{R \in Rul} Strength\_factor(R) * Specifity\_factor(R)$$

is the largest is the winner and the example is classified as being a member of *C*, where *Rul* denotes the set of all completely matching rules *R* describing *C*. This process reminds voting by rules for concepts.

If an example is not completely matched by any rule, some classification systems use *partial matching*. If complete matching is impossible, all partially matching rules are identified. These are rules with at least one attribute-value pair matching the corresponding attribute-value pair of an example.

For any partially matching rule R, the additional factor, called *Matching factor* (R), is computed. Matching factor (R) is defined as the ratio of the number of matched attribute-value pairs of a rule R with the case to the total number of attribute-value pairs of the rule R. In partial matching, the concept C for which the following expression

$$\sum_{R \in Rul'} Matching\_factor\,(R) * Strength\_factor\,(R) * Specifity\_factor\,(R)$$

is the largest is the winner and the example is classified to C, where Rul' is the set of all partially matching rules R describing C.

4. Sensitivity and specificity

In many applications, e.g., in medicine, we distinguish between two classes: primary and secondary. In medicine the primary class, defined as the class of all cases that should be diagnosed as affected by a disease, is more important.

The set of all correctly classified cases from the primary class are called true-positives, incorrectly classified primary cases are called false-negatives, correctly classified secondary cases are called true-negatives, and incorrectly classified secondary cases are called false-positives.

Sensitivity is the conditional probability of true-positives given primary class, i.e., the ratio of the number of true-positives to the sum of the number of true-positives and false-negatives. Specificity is the conditional probability of true-negatives given secondary class, i.e., the ratio of the number of true-negatives to the sum of the number of true-negatives and false-positives.

Usually, by applying techniques described later, we may increase sensitivity at the cost of specificity. It is difficult to estimate what are the optimal values of sensitivity and specificity. In our experiments, we applied an analysis presented by Bairagi and Suchindran (1989). Let p be a probability of the correct prediction, i.e., the ratio of all true positives and all false positives to the total number of all cases. Let P be the probability of an actual primary class, i.e., the ratio of all true positives and all false negatives to the total number of all cases. Then

$$p = \text{Sensitivity} * P + (1 - \text{Specificity}) * (1 - P).$$

As Bairagi and Suchindran observed (1989), we would like to see the change in p as large as possible with a change in P, i.e., we would like to maximize

$$\frac{dp}{dP} = \text{Sensitivity} + \text{Specificity} - 1.$$

Thus the optimal values of sensitivity and specificity correspond to the maximal value of Sensitivity + Specificity − 1. The sum of sensitivity and specificity is called a *gain*. Thus, in our experiments the objective was to maximize gain.

5. Increasing the strength of rules

As a result of rule induction, the average of all rule strengths for the bigger class is greater than the average of all rule strengths for the more important but smaller primary class. During classification of unseen cases, rules matching a case and voting for the primary class are outvoted by rules voting for the bigger, secondary class. Thus the sensitivity is low and the resulting classification system would be rejected by the users.

Therefore it is necessary to increase sensitivity. The simplest way to increase sensitivity is to add cases to the primary class in the data set, e.g., by adding duplicates of the available cases. The total number of training cases will increase, hence the total running time of the rule induction system will also increase. Adding duplicates will not change the knowledge hidden in the original data set, but it may create a balanced data set so that the average rule set strength for both classes will be approximately equal. The same effect may be accomplished by increasing the average rule strength for the primary class. In our first approach to dealing with imbalanced data sets, we selected the optimal rule set by multiplying the rule strength for all rules describing the primary class by the same real number called a *strength multiplier* (Grzymala-Busse *et al.*, 1999).

In general, the sensitivity increases with the increase of the strength multiplier. At the same time, the specificity decreases.

In our experiments, rule strength for all rules describing the primary class was increased incrementally. The process was terminated when gain was decreased.

6. Replacing rules

Unlike the previous technique, this approach is based on replacing the rule set for the primary class by another rule set that improves the chance of a case from the primary class to be selected by the "bucket brigade" algorithm, because this case can be matched by multiple rules voting for the primary class. Although strengths of new rules may be generally lower than for those for the secondary class, their number leads to an increased number of votes. New rules are generated directly from data, and their strength is modified neither by an induction algorithm, nor by a classifier.

Rules for the primary class were generated by the EXPLORE algorithm (Stefanowski and Vanderpooten, 2001). As opposed to LEM2, EXPLORE induces all rules that satisfy certain requirements, e.g., the strength greater than a given value, or the length of a rule smaller than a specified threshold.

The EXPLORE algorithm is based on a breadth-first exploration to generate rules of increasing size, starting from rules with single conditions. Exploration of a specific branch is terminated when a rule satisfying the requirements is generated or a stopping condition SC is satisfied (i.e., it is impossible to satisfy the requirements).

We use similar definitions as in Section 2. Decision rule R has a condition part, a conjunction of q elementary conditions $T = t_1 \wedge t_2 \wedge \ldots \wedge t_q$, where t is an attribute value pair (a, v), and a decision part indicating the concept. Set $[T]_C^+ = [T] \cap C$ denotes the set of positive learning examples covered by the rule. The rules are evaluated mainly by the measure of the (relative) strength $strength(R) = |[T]_C^+|/|[C]|$. An initial list LS of elementary conditions (e.g., simple attribute-value conditions) is created and given as an input to Explore. Obviously, conditions in LS must cover at least one example from a decision concept, they may also be subject to specific constraints on the syntax. This initial list is first pruned to discard selectors which directly correspond to rules as well as those which already satisfy SC and thus cannot be a basis for rules (subprocedure *Is_Good_Candidates*). Selectors remaining in LS are then combined to form complexes (i.e., conjunctions of selectors) which will be candidates for conditions of rules. The resulting complexes are then tested by subprocedure *Is_Good_Candidates*.

In the *Explore* algorithm the search for rules is controlled by the parameters called *stopping conditions SC* defined by the user. Since EXPLORE is rule strength driven, the definition of SC is based on determining the threshold value for the minimal strength of the conjunction of conditions that is a candidate for the condition part of the rule. If its strength is lower than SC, then the conjunction is discarded, otherwise it can be further evaluated. Additionally, the user may also define a threshold expressing the minimum value of the level of discrimination $D(R)$ of the rules to be generated.

```
procedure Explore
(input LS: list of valid elementary
conditions,
SC: stopping conditions,
output R: set of rules)
{Main procedure}
begin
R:=∅;
     IsGoodCandidate (LS,  R); {LS
                 is a list of valid
                 elementary conditions
                 t1, t2, ... tn – ordered
                 according to the
                 decreasing strength}
     Q := LS;   {Copy current LS to
     queue Q}
     while  Q <> ∅ do
            begin
                 select the first con-
                 junction T from Q;
```

```
            Q:= Q −{ T };
            gene-rate the set LC of
            all the conjunctions
            T ∧ t_{h+1}, T ∧ t_{h+2}, ...
            T ∧ t_{h+n}, where h is
            the highest index of
            the condition involved
            in T
            {generate extensions
            of T using LS}
            IsGoodCandidate (LC, R);
            Q := Q ∪LC; {Place all
            candidates from LC at
            the end of Q}
      end
end

procedure IsGoodCandidate
(input  L: list of conjunctions,
R:  set of rules)
{This procedure prunes list L discard-
ing conjunctions whose extension
cannot generate rules due to SC and
conjunctions corresponding to rules
which are already stored in R}
begin
      for each  T in L do
      begin
            if T satisfies SC then
            L := L − {T}
            else  [T] ⊆ [C] then
                  begin
                        R := R ∪ {T};
                        L := L − {T};
                  end
      end
end
```

For further details of the EXPLORE algorithm see (Stefanowski, 1998; Stefanowski and Vanderpooten, 2001; Stefanowski and Wilk, 2001).

Although there are several requirements that can be specified for EXPLORE, we are focused only on the minimal strength of a rule (for discussion see Stefanowski 1998, Stefanowski and Vanderpooten, 2001). The threshold is modified in order to obtain an optimal set of rules, i.e., leading to the best classification outcome (Stefanowski and Wilk, 2001). To avoid repeating induction of rules with varying strengths, a set of rules is generated only once for the smallest acceptable threshold, and then appropriate subsets are selected. The smallest strength is set to the minimal strength observed for rules generated for the primary class by LEM2 (we have to induce a temporary set of rules that is later discarded and not used for classification). This prevents the induction of an overwhelming number of rules (it would happen, if the thresholds were too low). Rules for the secondary class are created as previously, using LEM2.

To find an optimal set of rules according to the gain criterion described in Section 4 we verify, in a number of steps, various subsets of rules for the primary class, starting from the strongest rules to all rules created by EXPLORE. In each step we consider rules for the primary class with strength greater than the current threshold (the threshold changes from the maximal value to the minimal one, which were observed in a temporary set of rules generated by LEM2) and combine them with rules obtained for the secondary class (this set does not change) into a final set used by the classifier. Having finished the process, we are able to point out the threshold and a set of rules leading to the best classification outcome. If there is a tie (the same outcome for several sets), we select the higher threshold (the smaller set), following the Occam's razor principle.

7. Experiments

Some of the original data sets, used for our experiments, contained numerical attributes. These attributes were discretized using cluster analysis. Clusters were first formed from data with numerical attributes. Then those clusters were projected on the attributes that originally were numerical. The resulting intervals were merged to reduce the number of intervals and, at the same time, to preserve consistency. Some data sets contained missing attribute values, which were substituted with the most frequent value among examples belonging to the considered class.

For calculation of classification performance we used twofold cross-validation. Both approaches used the same sets of cases, with the same split into

Table 1. Data sets used in experiments

Data set	*Number of objects*			*Ratio of objects*	
	Total	*Primary*	*Secondary*	*Primary (%)*	*Secondary (%)*
ABDOMINAL-PAIN	723	202	521	27.9	72.1
BREAST-SLOVENIA	294	89	205	30.3	69.7
BREAST-WISCONSIN	625	112	513	17.9	82.1
BUPA	345	145	200	42.0	58.0
GERMAN	666	209	457	31.4	68.6
HEPATITIS	155	32	123	20.6	79.4
PIMA	768	268	500	34.9	65.1
SCROTAL-PAIN	201	59	142	29.4	70.6
UROLOGY	498	155	343	31.1	68.9

Table 2. Results for the original LEM2 algorithm

Data set	*Sensitivity*	*Specificity*	*Gain*	*Error (%)*
ABDOMINAL-PAIN	0.5842	0.9290	1.5132	16.74
BREAST-SLOVENIA	0.3647	0.8856	1.2503	26.92
BREAST-WISCONSIN	0.3125	0.9259	1.2384	18.40
BUPA	0.3241	0.7400	1.0641	43.48
GERMAN	0.3014	0.8468	1.1482	32.43
HEPATITIS	0.4375	0.9512	1.3887	15.48
PIMA	0.3918	0.8260	1.2178	32.55
SCROTAL-PAIN	0.5424	0.8310	1.3734	25.37
UROLOGY	0.1218	0.8227	0.9445	39.60

Table 3. Best results for increasing rule strength

Data set	*Multiplier*	*Sensitivity*	*Specificity*	*Gain*	*Error (%)*
ABDOMINAL-PAIN	5.0	0.8069	0.8484	1.6553	16.32
BREAST-SLOVENIA	1.0	0.3647	0.8856	1.2503	26.92
BREAST-WISCONSIN	5.0	0.5714	0.8674	1.4388	18.56
BUPA	3.0	0.5586	0.5850	1.1436	42.61
GERMAN	4.0	0.5789	0.6411	1.2200	37.84
HEPATITIS	18.0	0.8438	0.7724	1.6162	21.29
PIMA	3.5	0.5933	0.7640	1.3573	29.56
SCROTAL-PAIN	3.0	0.6780	0.8099	1.4879	22.89
UROLOGY	14.0	0.5192	0.4942	1.0134	49.48

two subsets. Although twofold cross-validation may be not sufficient to estimate the actual error rate, our objective was to compare our approaches to handling imbalanced data sets.

Results of our experiments are presented in Tables 1–4. Most of the data sets, presented in Table 1, were taken from the Repository at the University of California, Irvine, CA, USA. Others come from medical applications of rule induction approaches (Wilk *et al.*, 2004). In Tables 2–4, sensitivity, specificity, gain and the total error are presented.

8. Conclusions

Results of our experiments show that an increase in gain, comparing with the original LEM2, may

Table 4. Best results for replacing rules (EXPLORE approach)

Data set	*Support*	*Sensitivity*	*Specificity*	*Gain*	*Error (%)*
ABDOMINAL-PAIN	16.0	0.6939	0.9175	1.6114	14.52
BREAST-SLOVENIA	3.0	0.4709	0.8411	1.3120	26.92
BREAST-WISCONSIN	2.0	0.6385	0.8160	1.4545	21.43
BUPA	2.0	0.4275	0.6300	1.0575	45.50
GERMAN	5.0	0.6271	0.7265	1.3536	30.50
HEPATITIS	6.0	0.5830	0.9175	1.5005	15.52
PIMA	3.0	0.5686	0.7829	1.3514	29.30
SCROTAL-PAIN	4.0	0.6887	0.8724	1.5611	18.44
UROLOGY	6.0	0.3403	0.7017	1.0420	41.57

be accomplished by both approaches: changing strength multipliers for rules describing the primary class and by replacing rule sets for the primary class using EXPLORE.

The purpose of our experiments was to compare both approaches for dealing with imbalanced data sets. In order to compare the overall performance of both approaches, the Wilcoxon Signed Ranks Test (Hamburg, 1983), a nonparametric test for significant differences between paired observations, was used. As a result, the difference in performance for both approaches for dealing with imbalanced data sets, in terms of gain, is statistically insignificant. Additionally, the same conclusion is true for the error rate: the difference in performance for both approaches, in terms of error rate, is also statistically insignificant. Therefore, the appropriate approach to dealing with imbalanced data sets should be selected individually for a specific data set. The first approach for increasing sensitivity, based on changing the rule strength for the primary class, is less expensive computationally than the second approach, based on replacing rule the set for the primary class.

We can extend both approaches by also postprocessing rule sets for stronger secondary class using rule truncation, i.e., removing weak rules describing only a few learning cases. Such possibilities can be explored in further research.

For many important applications, e.g., medical area, an increase in sensitivity is crucial, even if it is achieved at the cost of specificity. Thus, the suggested approaches for dealing with imbalanced data sets may be successfully applied for data mining from imbalanced data.

References

Bairagi, R. and Suchindran, C. M. (1989) An estimator of the cutoff point maximizing sum of sensitivity and specificity. *Sankhya*, Series B, *Indian Journal of Statistics*, **51**, 263–269.

Booker, L. B., Goldberg, D. E. and Holland, J. F. (1990) Classifier systems and genetic algorithms, in *Machine Learning. Paradigms and Methods*, Carbonell, J. G. (ed.), The MIT Press, Cambridge, MA, pp. 235–282.

Grzymala-Busse, J. W. (1992) 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, Norwell, MA, pp. 3–18.

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

Grzymala-Busse, J. W., Goodwin, L. K. and Zhang, X. (1999) Increasing sensitivity of preterm birth by changing rule strengths. *Proceedings of the 8th Workshop on Intelligent Information Systems (IIS'99)*, Ustron, Poland, 127–136.

Grzymala-Busse, J. W., Goodwin, L. K., Grzymala-Busse, W. J. and Zheng, X. (2000) An approach to imbalanced data sets based on changing rule strength. *Learning from Imbalanced Data Sets, AAAI Workshop at the 17th Conference on AI, AAAI-2000*, Austin, TX, 69–74.

Grzymala-Busse, J. W., Stefanowski, J. and Wilk, S. (2004) A comparison of two approaches to data mining from imbalanced data. *Proceedings of the KES 2004, 8th International Conference on Knowledge-based Intelligent Information + Engineering Systems, Wellington, New Zealand, September 20–24, 2004*. Part I, Lecture Notes in AI, Vol. 3213, Springer Verlag, Berlin Heidelberg, 2004, 757–763.

Hamburg, M. (1983) *Statistical Analysis for Decision Making*, 3rd edn., New York, NY: Harcourt Brace Jovanovich Inc.

Holland, J. H., Holyoak, K. J. and Nisbett, R. E. (1986) *Induction. Processes of Inference, Learning, and Discovery*. Cambridge, MA: The MIT Press.

Japkowicz, N. (2000) Learning from imbalanced data sets: a comparison of various strategies. *Learning from Imbalanced Data Sets, AAAI Workshop at the 17th Conference on AI, AAAI-2000*, Austin, TX, July 30–31, pp. 10–17.

Pawlak, Z., Grzymala-Busse, J. W., Slowinski, R. and Ziarko, W. (1995) Rough sets. *Communications of the ACM*, **38**, 89–95.

Pawlak, Z. (1982) Rough sets. *International Journal Computer and Information Sciences*, **11**, 341–356.

Pawlak, Z. (1991) *Rough Sets. Theoretical Aspects of Reasoning about Data*. Norwell, MA: Kluwer Academic Publishers.

Stefanowski, J. (1998) On rough set based approaches to induction of decision rules, in *Rough Sets in Knowledge Discovery*, Skowron, A. and Polkowski, L. (eds), Vol. 1, Physica Verlag, Heidelberg, pp. 500–529.

Stefanowski, J. and Vanderpooten, D. (2001) Induction of decision rules in classification and discovery-oriented perspectives. *International Journal of Intelligent Systems*, **16**(1), 13–28.

Stefanowski, J. and Wilk, S. (2001) Evaluating business credit risk by means of approach integrating decision rules and case based learning. *International Journal of Intelligent Systems in Accounting, Finance and Management*, **10**, 97–114.

Wilk, S., Slowinski, R., Michalowski, W. and Greco, S. (2004) Supporting triage of children with abdominal pain in the emergency room. *European Journal of Operation Research* **160**(3), 696–709.