Combination of Metric-Based and Rule-Based Classification

Arkadiusz Wojna

Institute of Informatics, Warsaw University
Banacha 2, 02-097, Warsaw, Poland
wojna@mimuw.edu.pl

Abstract. We consider two classification approaches. The metric-based approach induces the distance measure between objects and classifies new objects on the basis of their nearest neighbors in the training set. The rule-based approach extracts rules from the training set and uses them to classify new objects. In the paper we present a model that combines both approaches. In the combined model the notions of rule, rule minimality and rule consistency are generalized to metric-dependent form.
An effective polynomial algorithm implementing the classification model based on minimal consistent rules has been proposed in [2]. We show that this algorithm preserves its properties in application to the metric-based rules. This allows us to combine this rule-based algorithm with the k nearest neighbor (k-nn) classification method. In the combined approach the rule-based algorithm takes the role of nearest neighbor voting model. The presented experiments with real data sets show that the combined classification model have the accuracy higher than single models.

1 Introduction

Empirical comparison of rule-based systems [2] and metric-based methods [1] shows that each approach is more accurate than the other one for some classification problems but not for all. Therefore a lot of work has been done to construct hybrid classifiers that take the advantages of both approaches [4, 6, 8].

All these methods focus on how to use distance measure or the nearest neighbors of an object to be classified to improve the selection of rules for classification. However, in classification problems with many attributes the space of possible rules is enormous and searching for accurate rules is a very hard task. Therefore for many such problems the k nearest neighbors (k-nn) method is more accurate than rule-based systems and the approach where one uses rules to improve k-nn can be more effective than using k-nn to improve rule-based classification.

In the paper we propose the general hybridization framework where the notion of rules is generalized to a metric-dependent form and the rules generated from a training set are used to verify and improve selection of nearest neighbors in the k-nn. We apply this framework to the case where minimal consistent rules [9] are used to improve the k-nn. The idea of improving k-nn by rule induction was used in [7]. However, the rules in [7] have specific, non-uniform conditions and do not correspond to the metric-based model presented in the paper.

2 Metric based generalization of minimal consistent rules

We assume that a finite set of training examples U_{trn} is provided. Each training example $x \in U_{trn}$ is decribed by a vector of attribute values $(x_1, \ldots, x_n)$ corressponding to a fixed set of n attributes $A = \{a_1, \ldots, a_n\}$, and by its decision value $dec(x)$ from a discrete and finite set $V_{dec} = \{d_1, \ldots d_m\}$.

Originally the notions of rule minimality and consistency [9] were introduced for rules with equality conditions: $a_{i_1} = v_1 \wedge \ldots \wedge a_{i_p} = v_p \Rightarrow dec = d_j$. We generalize this approach to a metric-dependent form. We assume only that the metric ρ is an l_p-combination of metrics for particular attributes ($p \geq 1$):

$$\rho(x, y) = \left(\sum_{i=1}^{n} w_i \cdot \rho_i(x_i, y_i)^p \right)^{\frac{1}{p}}. \tag{1}$$

The equality $a_{i_j} = v_j$ as the condition in the premise of a rule represents selection of attribute values, in this case always a single value. We replace equality conditions with a more general metric based form of conditions. This form allows us to select more than one attribute value in a single attribute condition, and thus, to obtain more general rules.

Definition 1 *A* generalized rule *consists of a premise and a consequent:*

$$\rho_{i_1}(v_1, *) \leq r_1 \wedge \ldots \wedge \rho_{i_p}(v_p, *) < r_p \Rightarrow dec = d_j.$$

Each condition $\rho_{i_q}(v_q, *) \leq r_q$ or $\rho_{i_q}(v_q, *) < r_q$ in the premise of the generalized rule represents the range of acceptable values of a given attribute a_{i_q} around a given value v_q. The range is specified by the distance function ρ_{i_q} that is the component of the total distance ρ and by the threshold r_q.

The definition of rule consistency with a training set for the generalized rules is analogous to the equality-based rules. This describes the rules that classify correctly all the covered objects in a given training set:

Definition 2 *A generalized rule $\alpha \Rightarrow dec = d_j$ is* consistent *with a training set U_{trn} if for each object $x \in U_{trn}$ matching the rule the decision of the rule is correct, i.e., $dec(x) = d_j$.*

Next, we generalize the notion of rule minimality.

Definition 3 *A consistent generalized rule $\rho_{i_1}(v_1, *) < r_1 \wedge \ldots \wedge \rho_{i_p}(v_p, *) < r_p \Rightarrow dec = d_j$ is* minimal *in a training set U_{trn} if for each attribute $a_{i_q} \in \{a_{i_1}, \ldots, a_{i_p}\}$ occurring in the premise of the generalized rule the rule $\rho_{i_1}(v_1, *) < r_1 \wedge \ldots \wedge \rho_{i_q}(v_q, *) \leq r_q \wedge \ldots \wedge \rho_{i_p}(v_p, *) < r_p \Rightarrow dec = d_j$ with the enlarged range of acceptable values on this attribute (obtained by replacing $<$ by $\leq$ in the condition of the original rule) is inconsistent with the training set U_{trn}.*

Observe, that each condition in the premise of a minimal consistent generalized rule is always a strict inequality. It results from the assumption that a training set U_{trn} is finite.

Both the metric and the metric-based rules can be used to define tolerance relations which are used in construction of generalized approximation spaces [10].

Algorithm 1 Algorithm $decision_{local-rules}(x)$ classifying a given test object x based on lazy induction of local rules.

```
for each d_j ∈ V_dec  support[d_j] := ∅
for each y ∈ U_trn
   if r_local(x, y) is consistent with U_trn then
      support[dec(y)] := support[dec(y)] ∪ {y}
return arg max_{d_j ∈ V_dec} |support[d_j]|
```

3 Effective classification by minimal consistent rules

In this section we recall the classification model based on all minimal consistent rules in the original equality-based form [2]. The complete set of all minimal consistent rules has good theoretical properties: it corresponds to the set of all rules generated from all local reducts of a given training set [12]. The original version of the classification model [2] uses the notion of rule support:

Definition 4 *The* support *of a rule $a_{i_1} = v_1 \wedge \ldots \wedge a_{i_p} = v_p \Rightarrow dec = d_j$ in a training set U_{trn} is the set of all the objects from U_{trn} matching the rule and with the same decision d_j:*

$$\begin{aligned}&support(a_{i_1} = v_1 \wedge \ldots \wedge a_{i_p} = v_p \Rightarrow dec = d_j) = \\ &\quad \{x = (x_1, \ldots, x_n) \in U_{trn} : x_{i_1} = v_1 \wedge \ldots \wedge x_{i_p} = v_p \wedge dec(x) = d_j\}.\end{aligned}$$

The rule support based models compute the support set for each rule $r \in R$ covering a test object x from a given set of rules R and then they select the decision with the greatest total number of the supporting objects:

$$dec_{rules}(x, R) := \arg\max_{d_j \in V_{dec}} \left| \bigcup_{\alpha \Rightarrow dec = d_j \in R:\, x \text{ satisfies } \alpha} support(\alpha \Rightarrow dec = d_j) \right|. \tag{2}$$

The classification model proposed in [2] is the rule support model where R is assumed to be the set of all minimal consistent rules.

The number of all minimal consistent rules can be exponential. Therefore Bazan [2] proposed Algorithm 1 that classifies objects on the basis of the set of all minimal consistent rules without computing them explicitly. It simulates the rule support based classifier dec_{rules} by lazy induction of local rules.

Definition 5 *The* local rule *for a given pair of a test object x and a training object $y \in U_{trn}$ is the rule $r_{local}(x, y)$ defined by*

$$\bigwedge_{a_i \in A:\, y_i = x_i} a_i = y_i \Rightarrow dec = dec(y).$$

The conditions in the premise of the local rule $r_{local}(x, y)$ are chosen in such a way that both the test object x and the training object y match the rule and the rule is maximally specific relative to the matching condition. The following relation holds between minimal consistent rules and local rules:

Fact 6 *[2] The premise of a local rule $r_{local}(x, y)$ implies the premise of a certain minimal consistent rule if and only if the local rule $r_{local}(x, y)$ is consistent with the training set U_{trn}.*

This property made it possible to prove that Algorithm 1 simulates correctly the classifier based on all minimal consistent rules:

Corollary 7 *[2] The classification result of the rule support based classifier from Equation 2 with the set R of all minimal consistent rules and the lazy local rule induction classifier (Algorithm 1) is the same for each test object x:*

$$dec_{rules}(x, R) = decision_{local-rules}(x).$$

The consistency checking of a local rule $r_{local}(x, y)$ can be made in $O(|U_{trn}|\,|A|)$ time. Hence, the classification of a single object by Algorithm 1 has the polynomial time complexity $O(|U_{trn}|^2\,|A|)$.

4 Metric based generalization of classification by minimal consistent rules

The original version of Algorithm 1 was proposed for data with nominal attributes only and it uses equality as the only form of conditions on attributes in the premise of a rule. We generalize this approach to the metric-dependent form of rules introduced in Section 2. This allows us to apply the algorithm to data both with nominal and with numerical attributes.

For the generalized version of the classifier based on the set of all generalized minimal consistent rules we use the notion of generalized rule center.

Definition 8 *An object $(x_1, \ldots, x_n)$ is the* center *of the generalized rule from Definition 1 if for each attribute condition $\rho_{i_q}(v_q, *) < r_q$ (or $\rho_{i_q}(v_q, *) \leq r_q$) occuring in its premise we have $x_{i_q} = v_q$.*

For a given set of generalized rules R and an object x by $R(x)$ we denote the set of all rules in R centered at x. Observe, that a rule can have many centers if there are attributes that do not occur in the premise of the rule.

In the generalized rule support based classification model the support set for a test object x is counted using all generalized minimal consistent rules centered at x:

$$decision_{gen-rules}(x, R) := \arg\max_{d_j \in V_{dec}} \left| \bigcup_{r \in R(x)} support(r) \right| \tag{3}$$

where R contains all generalized minimal consistent rules. Although in the generalized version we consider only minimal consistent rules centered at a test object the number of these rules can be exponential as in the non-generalized version.

Since it is impossible to enumerate all generalized minimal consistent rules in practice, we propose to simulate the generalized rule support based classification model from Equation 3 by analogy to Algorithm 1. First, we introduce the definition of a generalized local rule analogous to Definition 5. The conditions in generalized local rule are chosen in such a way that both the test and the training object match the rule and the conditions are maximally specific.

Definition 9 *The* generalized local rule *for a given pair of a test object x and a training object $y \in U_{trn}$ is the rule $r_{gen-local}(x, y)$:*

$$\bigwedge_{a_i \in A} \rho_i(x_i, *) \leq \rho_i(x_i, y_i) \Rightarrow dec = dec(y).$$

First, we identify the relation between the original and the generalized notion of local rule. Let us consider the case where to define the generalized rules the Hamming metric is used for all the attributes:

$$\rho_i(x_i, y_i) = \begin{cases} 1 \text{ if } x_i \neq y_i \\ 0 \text{ if } x_i = y_i. \end{cases}$$

It is easy to check that:

Fact 10 *For the Hamming metric the generalized local rule $r_{gen-local}(x, y)$ in Definition 9 is equivalent to the local rule $r_{local}(x, y)$ in Definition 5.*

The most important property of the generalization is the relation between generalized minimal consistent rules and generalized local rules analogous to Fact 6.

Theorem 11 *The premise of the generalized local rule $r_{gen-local}(x, y)$ implies the premise of a certain generalized minimal consistent rule centered at x if and only if the generalized local rule $r_{gen-local}(x, y)$ is consistent with U_{trn}.*

Proof. First, we show that each generalized local rule $r_{gen-local}(x, y)$ consistent with U_{trn} extends to the generalized minimal rule centered at x. We define the sequence of rules $r^0, \ldots, r^n$. The first rule is the local rule $r^0 = r_{gen-local}(x, y)$. To define each next rule r_i we assume that the previous rule r_{i-1}:

$$\bigwedge_{1 \leq j < i} \rho_j(x_j, *) < M_j \bigwedge_{i \leq j \leq n} \rho_j(x_j, *) \leq \rho_j(x_j, y_j) \Rightarrow dec = dec(y).$$

is consistent with the training set U_{trn} and the first $i - 1$ conditions of the rule r_{i-1} are maximally general, i.e., replacing any strong inequality $\rho_j(x_j, *) < M_j$ for $j < i$ by the weak makes this rule inconsistent. Let S_i be the set of all the object that satisfy the premise of the rule r_{i-1} with the condition on the attribute a_i removed:

$$S_i = \{z \in U_{trn} : z \text{ satisfies } \bigwedge_{1 \leq j < i} \rho_j(x_j, *) < M_j \bigwedge_{i < j \leq n} \rho_j(x_j, *) \leq \rho_j(x_j, y_j)\}.$$

In the rule r_i the i-th condition is maximally extended in such way that the rule remains consistent. It means that the range of acceptable values for the attribute a_i in the rule r_i has to be equal or less than the attribute distance from x to any object in S_i with a decision different from $dec(y)$. If S_i does not contain an object with a decision different from $dec(y)$ the range remains unlimited:

$$M_i = \begin{cases} \infty & \text{if } \forall z \in S_i\, dec(z) = dec(y) \\ \min\{\rho_i(x_i, z_i) : z \in S_i \wedge dec(z) \neq dec(y)\} & \text{otherwise.} \end{cases}$$

By limiting the range of values on the attribute a_i in the rule r_i to M_i:

$$\bigwedge_{1 \leq j < i} \rho_j(x_j, *) < M_j \wedge \rho_i(x_i, *) < M_i \bigwedge_{i < j \leq n} \rho_j(x_j, *) \leq \rho_j(x_j, y_j) \Rightarrow dec = dec(y)$$

we ensure that the rule r_i remains consistent. On the other hand, the value M_i is maximal: replacing the strong inequality by the weak inequality or replacing M_i by a larger value makes an inconsistent object $z \in S_i$ match the rule r_i.

Since r_{i-1} was consistent the range M_i is greater than the range for the attribute a_i in the rule r_{i-1}: $M_i > \rho(x_i, y_i)$. Hence, the ranges for the previous attributes $M_1, \ldots, M_{i-1}$ remain maximal in the rule r_i: widening of one of these ranges in the rule r_{i-1} makes an inconsistent object match r_{i-1} and the same happens for the rule r_i.

By induction the last rule $r_n : \bigwedge_{1 \leq j \leq n} \rho_j(x_j, *) < M_j \Rightarrow dec = dec(y)$ in the defined sequence is consistent too and all the conditions are maximally general. Then r_n is consistent and minimal. Since the premise of each rule r_{i-1} implies the premise of the next rule r_i in the sequence and the relation of implication is transitive the first rule r_0 that is the generalized local rule $r_{gen-local}(x, y)$ of the objects x, y implies the last rule r_n that is a minimal consistent rule. Thus we have proved the theorem for the case when the generalized local rule is consistent.

In case where the generalized local rule $r_{gen-local}(x, y)$ is inconsistent with the training set each rule centered at x implied by $r_{gen-local}(x, y)$ covers all objects covered by $r_{gen-local}(x, y)$, in particular it covers an object causing inconsistency. Hence, each rule implied by $r_{gen-local}(x, y)$ is inconsistent too. □

Consider the classifier $decision_{gen-local-rules}(x)$ defined by Algorithm 1 with a single change: the generalized local rules $r_{gen-local}(x, y)$ are used instead of original local rules $r_{local}(x, y)$. Theorem 11 ensures that for each object x this algorithm counts all and only those objects that are covered by a certain generalized minimal consistent rule centered at x. Hence, we obtain the final conclusion.

Corollary 12 *The classification result of the generalized rule support based classifier from Equation 3 with the set R of all the generalized minimal consistent rules and Algorithm 1 used with the generalized local rules is the same for each test object x:*

$$decision_{gen-rules}(x, R) = decision_{gen-local-rules}(x).$$

The time complexity of the generalized lazy rule induction algorithm is the same as the complexity of the non-generalized version: $O(|U_{trn}|^2 |A|)$.

5 Combination of k nearest neighbors with generalized rule induction

To classify an object x the k-nn classifier finds the set $NN(x,k)$ of k nearest neighbors of x and it assigns the most frequent decision in $NN(x,k)$ to x:

$$decision_{knn}(x) := \arg \max_{d_j \in V_{dec}} \left|\{y \in NN(x,k) : dec(y) = d_j\}\right| . \tag{4}$$

The k-nn model implements the lazy learning approach: the k nearest neighbors of a test object x are searched during the classification. The previous approaches [4, 6, 8] combining k-nn with rule induction do not preserve the laziness of learning. We propose the algorithm that preserves lazy learning, i.e., rules are constructed in lazy way at the moment of classification. The proposed combination uses the metric based generalization of rules described in Section 4.

For each test object x Algorithm 1 looks over all the training examples $y \in U_{trn}$ during construction of the support sets $support[d_j]$. Instead of that we can limit the set of the considered examples to the set of the k nearest neighbors of x. The intuition is that training examples far from the object x are less relevant for classification than closer objects. Therefore in the combined method we use the modified definition of the rule support, depending on the object x:

Definition 13 *The* k-support *of the generalized rule* $\alpha \Rightarrow dec = d_j$ *for a test object* x *is the set:*

$$k - support(x, \alpha \Rightarrow dec = d_j) = \{y \in NN(x,k) : y \ matches \ \alpha \wedge dec(y) = d_j\}.$$

The k-support of the rule contains only those objects from the original support set that belong to the set of the k nearest neighbors.

Now, we define the classification model that combines the k-nn method with rule induction by using the k-supports of the rules:

$$decision_{knn-rules}(x, R) := \arg \max_{d_j \in V_{dec}} \left| \bigcup_{r \in R(x)} k - support(x, r) \right| . \tag{5}$$

where R is the set of all generalized minimal consistent rules. The classifier $decision_{knn-rules}(x, R)$ can be defined by the equivalent formula:

$$\arg \max_{d_j \in V_{dec}} \left|\{y \in NN(x,k) : \exists r \in R(x) \text{ supported by } y \wedge dec(y) = d_j\}\right| .$$

This formula shows that the combined classifier can be viewed as the k-nn classifier with the specific rule based zero-one voting model.

As for the generalized rule support classifier we propose an effective algorithm simulating the combined classifier $decision_{knn-rules}$ based on the generalized local rules. The operation of consistency checking for a single local rule in Algorithm 1 takes $O(|U_{trn}|\,|A|)$ time. We can use the following fact to accelerate this consistency checking operation in the generalized algorithm:

Algorithm 2 Algorithm $decision_{knn-local-rules}(x)$ simulating the classifier $decision_{knn-rules}(x, R)$ with lazy induction of the generalized local rules.

```
for each d_j ∈ V_dec  support[d_j] := ∅
neighbor_1, ..., neighbor_k := the k nearest neighbors of x
            sorted from the nearest to the farthest object
for each i := 1 to k
   if r_gen-local(x, neighbor_i) is consistent
   with neighbor_1, ..., neighbor_{i-1} then
      support[dec(neighbor_i)] :=support[dec(neighbor_i)]∪{neighbor_i}
return arg max_{d_j ∈ V_dec} |support[d_j]|
```

Fact 14 *For each training object $z \in U_{trn}$ matching a generalized local rule $r_{gen-local}(x, y)$ based on the distance ρ from Equation 1 the distance between the objects x and z is not greater than the distance between the objects x and y:*

$$\rho(x, z) \leq \rho(x, y).$$

Proof. The generalized local rule $r_{gen-local}(x, y)$ for a test object $x = (x_1, \ldots, x_n)$ and a training object $y = (y_1, \ldots, y_n)$ has the form

$$\bigwedge_{a_i \in A} \rho_i(x_i, *) \leq \rho_i(x_i, y_i) \Rightarrow dec = dec(y).$$

If $z = (z_1, \ldots, z_n)$ matches the rule then it satisfies the premise of this rule. It means that for each attribute $a_i \in A$ the attribute value z_i satisfies the following condition: $\rho_i(x_i, z_i) \leq \rho_i(x_i, y_i)$. Hence, we obtain that the distance between the objects x and z is not greater than the distance between the objects x and y:

$$\rho(x, z) = \left(\sum_{a_i \in A} w_i \rho_i(x_i, z_i)^p \right)^{\frac{1}{p}} \leq \left(\sum_{a_i \in A} w_i \rho_i(x_i, y_i)^p \right)^{\frac{1}{p}} = \rho(x, y). \qquad \square$$

The above fact proves that to check consistency of a local rule $r_{gen-local}(x, y)$ with a training set U_{trn} it is enough to check only those objects from the training set U_{trn} that are closer to x than the object y.

Algorithm 2 is the lazy simulation of the classifier $decision_{knn-rules}(x, R)$ combining the k nearest neighbors method with rule induction. The algorithm follows the scheme of Algorithm 1. There are two differences. First, only the k nearest neighbors of a test object x are allowed to vote for decisions. Second, the consistency checking operation for each local rule $r_{gen-local}(x, y)$ checks only those objects from the training set U_{trn} that are closer to x than the object y. Thus the time complexity of the consistency checking operation for a single neighbor is $O(k\,|A|)$. Hence, the cost of consistency checking in the whole procedure testing a single object is $O(k^2\,|A|)$. In practice, consistency checking takes less time than searching for the k nearest neighbors. Thus addition of the rule induction to the k nearest neighbors algorithm does not lengthen significantly the performance time of the k-nn method.

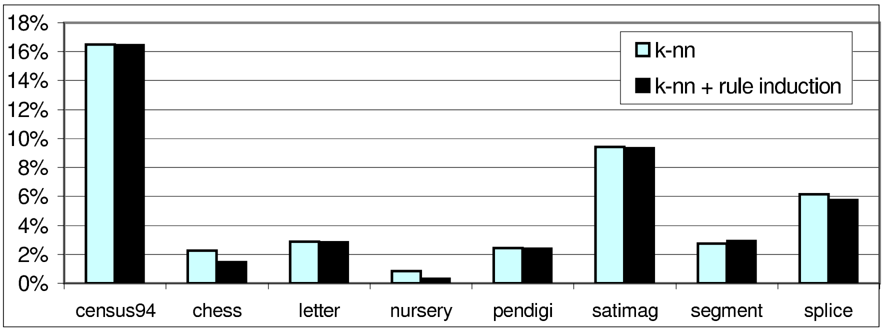

Fig. 1. The average classification error of the classical k-nn and the method combining k-nn with rule induction.

6 Experimental results

In this section we compare the performance of the classical k-nn with the combined model described in Section 5. To compare classification accuracy we used the 8 large data sets from the repository of University of California at Irvine [3]: *segment* (19 attr, 2310 obj), *splice-DNA* (60 attr, 2000 train, 1186 test obj), *chess* (36 attr, 3196 obj), *satimage* (36 attr, 4435 train, 2000 test obj), *pendigits* (16 attr, 7494 train, 3498 test obj), *nursery* (8 attr, 12960 obj), *letter* (16 attr, 15000 train, 5000 test obj) and *census94* (13 attr, 30160 train, 15062 test obj). The data provided originally as a single set (*segment, chess, nursery*) were randomly split into a training and a test part with the split ratio 2 to 1.

For each of these 8 data sets the classical k-nn and the combined model were trained and tested 5 times for the same partition of the data set and the average classification error was calculated for comparison. In each test of a given classification method, first, the metric defined by the City-VDM metric [4] was induced from the training set with $p = 1$ and attribute weighting [11], then the optimal value of k was estimated from the training set with the procedure [7] in the range $1 \leq k \leq 100$, and finally, the test part of a data set was tested with the previously estimated value of k.

Since distance-based voting by the k nearest neighbors outperforms majority voting [5], both in the k-nn and in the combined model we assigned the inverse squre distance weights $\frac{1}{\rho(x,y)^2}$ to the neighbors $y \in NN(x, k)$ instead of equal weights (used in Equations 4 and 5) while classifying a test object x.

Figure 1 shows that the method combining the k-nn with rule based induction is for all the data sets at least equally accurate as the k-nn alone, and sometimes it improves significantly the k-nn accuracy. For example, for *nursery* the combined model gives the 0.3% error in comparison to the 0.82% error of the pure k-nn and for *chess* the combined model gives the 1.46% error in comparison to the 2.24% error of the k-nn. Investigating lack of improvement for some data we observed that very few neighbors are rejected. For future we consider to apply rules that are more specific and selective than the local rules proposed.

7 Conclusions

In the paper we have introduced the new hybrid classification model that combines the rule based classification with the k nearest neighbors method. An important property of the combined model is that by adding rule based component we do not change essentially the performance time of the k nearest neighbors method. In this model the nearest neighbors of a test object are verified and filtered by the rule based-component. This gives more certainty that these neighbors are appropriate for decision making. The experiments confirm that the combined model can provide more accurate classification than the k-nn alone.
Acknowledgements. The research has been supported by the grants 4 T11C 040 24 and 3 T11C 002 26 from Ministry of Scientific Research and Information Technology of the Republic of Poland.

References

1. D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.
2. J. G. Bazan. Discovery of decision rules by matching new objects against data tables. In *Proceedings of the First International Conference on Rough Sets and Current Trends in Computing*, volume 1424 of *Lectures Notes in Artificial Intelligence*, pages 521–528, Warsaw, Poland, 1998. Springer-Verlag.
3. C. L. Blake and C. J. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html, Department of Information and Computer Science, University of California, Irvine, CA, 1998.
4. P. Domingos. Unifying instance-based and rule-based induction. *Machine Learning*, 24(2):141–168, 1996.
5. S. Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics*, 6:325–327, 1976.
6. A. R. Golding and P. S. Rosenbloom. Improving accuracy by combining rule-based and case-based reasoning. *Artificial Intelligence*, 87(1-2):215–254, 1996.
7. G. Góra and A. G. Wojna. RIONA: a new classification system combining rule induction and instance-based learning. *Fundamenta Informaticae*, 51(4):369–390, 2002.
8. J. Li, K. Ramamohanarao, and G. Dong. Combining the strength of pattern frequency and distance for classification. In *Proc. of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 455–466, Hong Kong, 2001.
9. A. Skowron and C. Rauszer. The discernibility matrices and functions in information systems. In R. Slowinski, editor, *Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory*, pages 331–362. Kluwer Academic Publishers, Dordrecht, 1992.
10. A. Skowron and J. Stepaniuk. Tolerance approximation spaces. *Fundamenta Informaticae*, 27(2-3):245–253, 1996.
11. A. G. Wojna. Center-based indexing in vector and metric spaces. *Fundamenta Informaticae*, 56(3):285–310, 2003.
12. J. Wróblewski. Covering with reducts - a fast algorithm for rule generation. In *Proceedings of the First International Conference on Rough Sets and Current Trends in Computing*, volume 1424 of *Lectures Notes in Artificial Intelligence*, pages 402–407, Warsaw, Poland, 1998. Springer-Verlag.