Ensemble of Decision Rules for Ordinal Classification with Monotonicity Constraints

Krzysztof Dembczyński[1], Wojciech Kotłowski[1], and Roman Słowiński[1,2]

[1] Institute of Computing Science, Poznań University of Technology,
60-965 Poznań, Poland
{kdembczynski, wkotlowski, rslowinski}@cs.put.poznan.pl
[2] Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland

Abstract. Ordinal classification problems with monotonicity constraints (also referred to as multicriteria classification problems) often appear in real-life applications, however they are considered relatively less frequently in theoretical studies than regular classification problems. We introduce a rule induction algorithm based on forward stagewise additive modeling that is tailored for this type of problems. The algorithm monotonizes the dataset (excludes highly inconsistent objects) using Dominance-based Rough Set Approach and generates monotone rules. Experimental results indicate that taking into account the knowledge about order and monotonicity constraints in the classifier can improve the prediction accuracy.

1 Introduction

An *ordinal classification problem with monotonicity constraints* consists in assignment of objects to finite number of ordered *classes*. Objects are described by attributes with ordered value sets and *monotonicity constraints* are present in the data: a higher value of an object on an attribute, with other values being fixed, should not decrease its class assignment. Problems of ordinal classification in the presence of monotonicity constraints are commonly encountered in real-life applications. A typical representative is multiple-criteria classification considered within multiple-criteria decision analysis (MCDA) [11]. Moreover, in many other domains ordinal and monotone properties follow from the domain knowledge about the problem and should not be neglected. They are encountered in such problems as bankruptcy risk prediction [10], breast cancer diagnosis [18], house pricing [14], credit rating [6] and many others.

In order to solve ordinal classification problem with monotonicity constraints, one can apply two steps for improving the accuracy of the classifier. The first one consists in "monotonization" of the dataset, i.e. exclusion of objects highly violating the monotone relationships. The second one consists in imposing the constraints such that only monotone functions are taken into account.

Dominance-based Rough Set Approach (DRSA) [11] is one of the first approaches introduced to deal with this type of problems. By replacing indiscernibility relation, considered in classical rough sets [15], with a dominance relation,

DRSA is able to handle inconsistencies following from violation of monotone relationships. In this context several specialized decision rule induction algorithms were proposed that were able to capture the ordinal nature of data and handle domain knowledge in the form of monotonicity constraints [12, 5] (we will refer to rules consistent with monotonicity constraints as *monotone* rules). Among them, DOMLEM [12] seems to be the most popular one. It aims at finding a minimal set of monotone rules covering the dataset, using the well-known sequential covering procedure as a search heuristic.

We follow a different methodology for monotone rule induction that is based on *forward stagewise additive modeling* (FSAM) [7], i.e. greedy procedure for minimizing a loss function on the dataset. The algorithm introduced in this paper, called MORE (from MOnotone Rule Ensembles), treats a single rule as a subsidiary base classifier in the ensemble. The rules are added to the ensemble iteratively, one by one. Each rule is fitted by concentrating on the examples which were hardest to classify correctly by rules that have already been generated. The advantage of this approach is that we use a single measure only (value of the empirical risk) at all stages of learning procedure: setting the best cuts (conditions), stopping the rule's growth and determining the weight of the rule; no additional features (e.g. impurity measures, pruning procedures) are considered. Such an approach was already considered in ordinary classification problems and algorithms such as RuleFit [9], SLIPPER [2], LRI [21] or Ensemble of Decision Rules [1]. The algorithm presented here can be seen as an extension of the last from the mentioned above methods. It monotonizes the dataset (excludes highly inconsistent objects) using DRSA and then generates monotone rules.

2 Problem Statement

In the *classification* problem, the aim is to predict the unknown class label $y \in Y = \{1, \ldots, K\}$ (decision value) of an object $\mathbf{x}$ using the description of the object in terms of p (condition) attributes, $\mathbf{x} = (x_1, x_2, \ldots, x_p) \in X$, where X is the *attribute space*. Here, we assume without loss of generality that value set of each attribute is a subset of $\mathbb{R}$, so that $X \subseteq \mathbb{R}^p$. In the *ordinal classification*, it is assumed that there is a meaningful order between classes which corresponds to the natural order between class labels. We also assume the presence of *monotonicity constraints* in the data.

In order to formalize the concept of monotonicity, we define the *dominance* relation as a binary relation on X in the following way: for any $\mathbf{x}, \mathbf{x}' \in X$ we say that $\mathbf{x}$ *dominates* $\mathbf{x}'$, denoted $\mathbf{x} \succeq \mathbf{x}'$, if on every attribute, $\mathbf{x}$ has value not smaller than $\mathbf{x}'$, $x_j \geq x'_j$, for all $j = 1, \ldots, p$. The dominance relation is a partial pre-order on X, i.e. it is reflexive and transitive. Having defined the dominance relation, we define the *monotone function* to be any function $f: X \to Y$ satisfying the monotonicity constraints:

$$\mathbf{x} \succeq \mathbf{x}' \to f(\mathbf{x}) \geq f(\mathbf{x}') \tag{1}$$

for any $\mathbf{x}, \mathbf{x}' \in X$.

Now, the problem of ordinal classification with monotonicity constraints can be stated as a problem of finding the *monotone classification function* $f(\mathbf{x})$ that predicts accurately values of y. The accuracy is measured in terms of the *loss function* $L(y, f(\mathbf{x}))$, which is the penalty for predicting $f(\mathbf{x})$ when the actual value is y. The overall accuracy of function $f(\mathbf{x})$ is defined as the expected loss (*risk*) according to the probability distribution of data to be predicted:

$$R(f) = E[L(y, f(\mathbf{x}))] \tag{2}$$

Since the data probability distribution is unknown, the function is learned from a set of n training examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ (*training set*). In order to minimize the value of risk (2), the learning procedure usually performs minimization of the *empirical risk*:

$$R_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)), \tag{3}$$

which is the value of a loss function on the training set (training error). It is possible to use a variety of loss functions for measuring accuracy; here, for simplicity, we assume the loss function to be the *absolute error loss*,

$$L_{\text{abs}}(y, f(\mathbf{x})) = |y - f(\mathbf{x})|. \tag{4}$$

Although in classification a 0-1 loss is often considered (defined as $L_{0-1}(y, f(\mathbf{x})) = 1$ if $y \neq f(\mathbf{x})$, 0 otherwise), absolute error loss has the advantage over 0-1 loss of being sensitive to the difference between predicted and actual class label, therefore taking the order between classes into account.

Solution to the ordinal classification problem with loss function (4) can be obtained by reducing the problem to $K-1$ binary problems. Let us define for a given class label y, auxiliary class labels y_k equal to 1 if $y \geq k$, otherwise -1, for each $k = 2, \ldots, K$. Then, we have $y = 1 + \sum_{k=2}^{K} \frac{1}{2}(y_k + 1)$. Moreover, let $f_k(\mathbf{x}) \in \mathbb{R}$ to be a function such that if $f(\mathbf{x}) \geq k$, then $f_k(\mathbf{x}) > 0$, and $f_k(\mathbf{x}) < 0$ otherwise. Then, we have:

$$L_{\text{abs}}(y, f(\mathbf{x})) = |y - f(\mathbf{x})| = \sum_{k=2}^{K} \left| \frac{1}{2}(y_k - \text{sgn}(f_k(\mathbf{x}))) \right| = \sum_{k=2}^{K} L_{0-1}(y_k f_k(\mathbf{x}))$$

where $L_{0-1}(y_k f_k(\mathbf{x}))$ is so called *margin 0-1 loss* defined for binary problems as $L_{0-1}(y_k f_k(\mathbf{x})) = \theta(-y_k f_k(\mathbf{x}))$, where $\theta(a)$ is a step function, equal to 1 for $a \geq 0$, and 0 elsewhere. The only problem is to satisfy $f_k(\mathbf{x}) \geq f_{k-1}(\mathbf{x})$, for $k = 3, \ldots, K$. If this condition is violated, one can try to find $g_k(\mathbf{x})$, $k = 2, \ldots, K$ satisfying this condition and being as close to $f_k(\mathbf{x})$, $k = 2, \ldots, K$ as possible:

$$\min \sum_{k=2}^{K} \left(f_k(\mathbf{x}) - g_k(\mathbf{x}) \right)^2$$

This is the problem of *isotonic regression* [17]. The final prediction is then $f(\mathbf{x}) = 1 + \sum_{k=2}^{K} \frac{1}{2}(\text{sgn}(g_k(\mathbf{x})) + 1)$. However, one does not need to solve isotonic

regression at all. It can be shown that simple voting procedure gives the same prediction. For a given object $\mathbf{x}$, if $f_k(\mathbf{x}) > 0$, then each class indicated by labels $k, \ldots, K$ gets vote $|f_k(\mathbf{x})|$; if $f_k(\mathbf{x}) < 0$, each class indicated by labels $1, \ldots, k-1$ gets vote $|f_k(\mathbf{x})|$. Votes are summed for each $k = 1, \ldots, K$ and $\mathbf{x}$ is classified to the class with the highest score.

From the monotonicity assumption *dominance principle* follows: for any two objects $\mathbf{x}_i, \mathbf{x}_j$ from the dataset, such that $\mathbf{x}_i \succeq \mathbf{x}_j$, it should hold $y_i \geq y_j$. However, it still may happen that in the dataset there exists an object $\mathbf{x}_i$, dominating another object $\mathbf{x}_j$, while it holds $y_i < y_j$. Such a situation violates the monotonicity assumption, so we shall call objects $\mathbf{x}_i$ and $\mathbf{x}_j$ *inconsistent*. Notice that no monotone function can approximate accurately inconsistent objects. Therefore, DRSA [11] and its stochastic extension [4] is applied in order to monotonize the data. Instead of using all data, we remove the inconsistent objects taking into account only stochastic lower approximations of decision classes:

$$\underline{Cl}_k^{\geq} = \{\mathbf{x}_i : \Pr(y \geq k|\mathbf{x}_i) \geq \alpha, i = 1, \ldots, n\},$$
$$\underline{Cl}_k^{\leq} = \{\mathbf{x}_i : \Pr(y \leq k|\mathbf{x}_i) \geq \alpha, i = 1, \ldots, n\}$$

where $\Pr(y \geq k|\mathbf{x}_i)$ ($\Pr(y \leq k|\mathbf{x}_i)$) is a probability, conditioned on $\mathbf{x}_i$, of class label at least (at most) k and $\alpha \in (0.5, 1]$ is a chosen *consistency level*. The probabilities are obtained using maximum likelihood estimation taking into account monotonicity constraints [4].

3 Ensemble of Decision Rules

This section describes the general scheme for decision rule induction. Here we focus on the binary classification case and assume that $Y = \{-1, 1\}$, where a "positive" class is ranked higher (in the order) to a "negative" class. This algorithm can be used in to each of the $K - 1$ binary problems resulting from the reduction of the ordinal classification problem. One can also use lower approximations instead of whole dataset.

Decision rule is a logical statement of the form: if [*condition*], $then$ [*decision*]. Let X_j be the set of all possible values for the j-th attribute. Condition part of the rule consist of elementary expressions of the form $x_j \geq s_j$ or $x_j \leq s_j$ for some $s_j \in X_j$. Let Φ denote the set of elementary expressions constituting the condition part of the rule, and let $\Phi(\mathbf{x})$ be an indicator function equal to 1 if an objects $\mathbf{x}$ satisfies the condition part of the rule (we also say that a rule *covers* an object), otherwise $\Phi(\mathbf{x}) = 0$. The decision is a single real value, denoted by α. Therefore, we define a decision rule as:

$$r(\mathbf{x}) = \alpha\Phi(\mathbf{x}). \tag{5}$$

Notice that the decision rule takes only two values, $r(\mathbf{x}) \in \{\alpha, 0\}$, depending whether $\mathbf{x}$ satisfies the conditions or not. In this paper, we assume the classification function is a linear combinations of M decision rules:

$$f(\mathbf{x}) = \alpha_0 + \sum_{m=1}^{M} r_m(\mathbf{x}), \tag{6}$$

Algorithm 1: Monotone Rule Ensemble – MORE

input : set of n training examples $\{(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)\}$,
M – number of decision rules to be generated.
output: default rule α_0, ensemble of decision rules $\{r_m(\mathbf{x})\}_1^M$.

$\alpha_0 = \arg\min_\alpha \sum_{i=1}^n \sigma(\alpha y_i)$;
$f_0(\mathbf{x}) = \alpha_0$;
for $m = 1$ *to* M **do**
  $\Phi_m(\mathbf{x}) = \arg\max_\Phi \left| \sum_{\Phi(\mathbf{x}_i)=1} y_i \sigma'(y_i f_{m-1}(\mathbf{x}_i)) \right|$;
  $\alpha_m = \arg\min_\alpha \sum_{\Phi_m(\mathbf{x}_i)=1} \sigma(y_i(f_{m-1}(\mathbf{x}_i) + \alpha))$;
  $r_m(\mathbf{x}) = \alpha_m \Phi_m(\mathbf{x})$;
  $f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + r_m(\mathbf{x})$;
end

where α_0 is a constant value, which can be interpreted as a default rule, covering the whole X. Object $\mathbf{x}$ is classified to the class indicated by the sign of $f(\mathbf{x})$. The combination (6) has very simple interpretation as a voting procedure: rules with positive α vote for the positive class, while rules with negative α – for the negative class. Object $\mathbf{x}$ is classified to the class with higher vote (which is equivalent to the sign of $f(\mathbf{x})$). Notice that in order to maintain monotonicity of $f(\mathbf{x})$, it is necessary and sufficient that for the m-th rule, α_m is positive when all elementary expressions in Φ_m are of the form $x_j \geq s_j$; similarly, for negative α_m all the conditions must be of the form $x_j \leq s_j$.

Rule induction is performed by minimizing the margin 0-1 loss function (classification error) on the set of n training examples (empirical risk). Notice that this loss function, is neither smooth nor differentiable. Therefore, we approximate it with the sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^x} \tag{7}$$

Thus, we minimize the following empirical risk:

$$R_{\text{emp}}(f) = \sum_{i=1}^{n} \sigma(y_i f(\mathbf{x}_i)) \tag{8}$$

However, finding a set of rules minimizing (8) is computationally hard, that is why we follow here FSAM, i.e. the rules are added one by one, greedily minimizing (8). We start with the default rule defined as:

$$\alpha_0 = \arg\min_\alpha R_{\text{emp}}(\alpha) = \arg\min_\alpha \sum_{i=1}^{n} \sigma(\alpha y_i). \tag{9}$$

Let $f_{m-1}(\mathbf{x})$ be a classification function after $m-1$ iterations, consisting of first $m-1$ rules and the default rule. The m-th decision rule $r_m(\mathbf{x}) = \alpha_m \Phi_m(\mathbf{x})$ should be obtained from $r_m = \arg\min_r R_{\text{emp}}(f_{m-1} + r)$, but in order to speed up

computations, it is built in two steps. First, we obtain value of $\Phi_m(\mathbf{x})$. Then, we obtain value of α_m by solving the line-search problem with formerly determined $\Phi_m(\mathbf{x})$. To explain the procedure for determining $\Phi_m(\mathbf{x})$, let us expand the value of the loss function up to the first order using $g(x+\alpha) \simeq g(x) + \alpha\frac{dg(x)}{dx}$:

$$\sigma(y_i(f_{m-1}(\mathbf{x}_i) + \alpha)) = \sigma(y_i f_{m-1}(\mathbf{x}_i)) + \alpha y_i \sigma'(y_i f_{m-1}(\mathbf{x}_i)), \tag{10}$$

where $\sigma'(x)$ is a derivative of sigmoid function $\sigma(x)$. Using (10) in (8) we approximate the empirical risk $R_{\text{emp}}(f_{m-1} + r)$ as:

$$\sum_{\Phi(\mathbf{x}_i)=1} \left[\sigma(y_i f_{m-1}(\mathbf{x}_i)) + \alpha y_i \sigma'(y_i f_{m-1}(\mathbf{x}_i))\right] + \sum_{\Phi(\mathbf{x}_i)=0} \sigma(y_i f_{m-1}(\mathbf{x}_i)). \tag{11}$$

However, minimizing (11) is equivalent to minimizing:

$$\mathcal{L}_m(\Phi) = \sum_{\Phi(\mathbf{x}_i)=1} y_i \sigma'(y_i f_{m-1}(\mathbf{x}_i)) \tag{12}$$

for any positive value of α or maximizing (12) for any negative value of α. Thus, the general idea of algorithm for finding Φ_m is the following: first we search for Φ_m^+ with positive α by minimizing $\mathcal{L}_m(\Phi)$, next we search for Φ_m^- with negative α by maximizing $\mathcal{L}_m(\Phi)$, and we choose Φ_m with higher $|\mathcal{L}_m(\Phi)|$, $\Phi_m = \arg\max\{|\mathcal{L}_m(\Phi_m^+)|, |\mathcal{L}_m(\Phi_m^-)|\}$. The procedure for finding Φ_m^+ (Φ_m^-) resembles the way the decision trees are generated. Here, we look for only one branch instead of the whole decision tree. At the beginning, Φ_m^+ (Φ_m^-) is empty and in each next step an elementary expression $x_j \geq s_j$ ($x_j \leq s_j$) is added to Φ_m^+ (Φ_m^-) until $\mathcal{L}_m(\Phi_m^+)$ ($\mathcal{L}_m(\Phi_m^-)$) cannot be decreased. Let us underline that a minimal value of $\mathcal{L}_m(\Phi_m^+)$ ($\mathcal{L}_m(\Phi_m^-)$) is a natural stop criterion, what differs this procedure from those used for decision trees generation. After Φ_m has been determined, α_m can be obtained by simply using the line search procedure to:

$$\alpha_m = \arg\min_{\alpha} \sum_{\Phi_m(\mathbf{x}_i)=1} \sigma(y_i(f_{m-1} + \alpha)). \tag{13}$$

In our implementation, to speed up the computations, instead of solving (9) and (13) we perform a gradient search with short step – we choose α_m to be a small, fixed value $\pm\gamma$ that corresponds to a learning rate.

4 Experimental Results

In order to test how our approach to rule induction behaves in practice, we selected eight datasets, for which it is known that monotonicity constraints make sense. Five datasets come from UCI repository [19]: Wisconsin Breast Cancer, CPU Performance, Ljubljana Breast Cancer, Boston House Pricing, Car. The other three were obtained from different sources: Den Bosch House Pricing [3], Bankruptcy Risk [10] and Windsor House Pricing [13]. Due to lack of space we omit the detailed characteristics of each dataset.

For each dataset we tested three regular classifiers which do not take order nor monotonicity into account: support vector machines (SVM) with linear kernel [20], j48 decision trees [16] and AdaBoost [8] with decision stump as a base learner. We used their implementations from Weka package [22]. Moreover, we also used two versions of our MORE algorithm. The first one induces decision rules from class unions. The second ("MORE+") employs Stochastic DRSA [4] and induces rules from lower approximations with consistency level 0.5. For SVM and j48, typical parameters from Weka were chosen; for AdaBoost we increased the number of iteration to 100 to make it more competitive; for MORE we have chosen $M = 100$ and $\gamma = 0.5$). For each dataset and for each algorithm, 10-fold cross validation was used and repeated 20 times to decrease the variance of the results. The measured error rate is mean absolute error, which is the value of absolute error loss on the testing set.

The results shown in Table 1 show a great improvement in accuracy when using monotone rule ensembles over the regular classifiers. This is probably due to the fact, that MORE utilizes the domain knowledge. Poor results of ordinary algorithms, e.g. AdaBoost for Windsor dataset, can be explained by the fact, that those algorithms are not adjusted to minimize absolute error. On the other hand, there is only a small improvement (if any) in using lower approximations in rule induction comparing to the rule induction from raw class unions.

5 Conclusions

We introduced a novel rule induction algorithm for ordinal classification problem in the presence of monotonicity constraints. The algorithm uses forward stage-wise additive modeling scheme for generating the ensemble of decision rules for binary problems. We show how to solve the ordinal classification problem with absolute error by solving binary subproblems with zero-one error. Due to specific nature of the problem, a syntax typical to monotone rules was used to find the statistically best ensemble. Moreover, we show how to use DRSA to deal with inconsistent objects. The experimental results show that incorporating such domain knowledge into classification algorithms can dramatically improve the prediction accuracy.

Table 1. Mean absolute error ± standard error. For each dataset, the best method and all methods within one standard error below the best are marked with bold.

Dataset	SVM	j48	Adaboost	MORE	MORE+
DenBosch	0.2055±.0042	0.1689±.0041	0.1294±.0025	**0.1181**±.0031	0.1303±.0035
CPU	0.4366±.0028	0.1261±.0037	0.5727±.0027	**0.0641**±.0023	**0.0641**±.0023
Wisconsin	**0.0324**±.0004	0.0536±.0015	0.0406±.0007	0.0359±.0007	0.0331±.0008
Bankruptcy	0.1692±.0039	0.1756±.0028	0.2692±.0090	**0.1256**±.0059	**0.1256**±.0059
Ljubljana	0.3203±.0035	**0.2437**±.0015	0.2727±.0024	0.2781±.0018	0.2510±.0028
Boston	0.3856±.0016	0.3813±.0042	0.7659±.0029	**0.3118**±.0019	**0.3101**±.0019
Windsor	0.5774±.0028	0.6440±.0042	0.9294±.0029	**0.5046**±.0025	**0.5040**±.0029
Car	0.6752±.0012	0.6517±.0016	0.4005±.0001	**0.0473**±.0007	0.0490±.0007

References

1. Błaszczyński, J., Dembczyński, K., Kotłowski, W., Słowiński, R., Szeląg, M.: Ensemble of decision rules. Foundations of Comp. and Decision Sc. 3–4 (31) (2006).
2. Cohen, W., Singer, Y.: A simple, fast, and effective rule learner. National Conference on Artificial Intelligence, pp. 335–342 (1999)
3. Daniels, H., Kamp, B.: Applications of MLP networks to bond rating and house pricing. Neural Computing and Applications 9, 226–234 (1999)
4. Dembczyński, K., Greco, S., Kotłowski, W., Słowiński, R.: Statistical model for rough set approach to multicriteria classification. In: Kok, J., Koronacki, J, Mántaras, R., Matwin, S, Mladenic, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 164–175 Springer, Heidelberg (2007)
5. Dembczyński, K., Pindur, R., Susmaga, R.: Generation of Exhaustive Set of Rules within Dominance-based Rough Set Approach. Electr. Notes Theor. Comp. Sc. 82 (2003)
6. Doumpos, M., Pasiouras, F.: Developing and testing models for replicating credit ratings: A multicriteria approach.Computational Economics 25, 327–341 (2005)
7. Hastie, T., Tibshirani, R., Friedman, J.: Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Heidelberg (2003)
8. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. J. of Comp. and System Sc. 55, 119–139 (1997)
9. Friedman, J., Popescu, B.: Predictive Learning via Rule Ensembles. Technical Report, Dept. of Statistics, Stanford University (2005)
10. Greco, S., Matarazzo, B., Słowiński, R.: A new rough set approach to evaluation of bankruptcy risk. In: Zopounidis, C. (ed.) Operational Tools in the Management of Financial Risks, pp. 121–136. Kluwer Academic Publishers, Dordrecht (1998)
11. Greco S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research 129, 1–47 (2001)
12. Greco, S., Matarazzo, B., Słowiński, R., Stefanowski, J.: An algorithm for induction of decision rules consistent with the dominance principle. In: Ziarko, W., Yao, Y. Y. (eds.) RSCTC 2000. LNCS (LNAI) 2005, pp. 304–313. Springer, London (2000)
13. Koop, G.: Analysis of Economic Data. John Wiley and Sons (2000)
14. Potharst, R., Feelders, A.J.: Classification trees for problems with monotonicity constraints. SIGKDD Explorations 4, 1–10 (2002)
15. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
16. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993)
17. Robertson, T., Wright, F., Dykstra, R.: Order Restricted Statistical Inference. John Wiley & Sons (1998)
18. Ryu, Y.U., Chandrasekaran, R., Jacob, V.: Data classification using the isotonic separation technique: Application to breast cancer prediction. European Journal of Operational Research 181, 842–854 (2007)
19. Asuncion, A., Newman, D.: (UCI) Repository of machine learning databases. `www.ics.uci.edu/~mlearn/MLRepository.html`, University of California, Irvine, School of Information and Computer Sciences (1998)
20. Vapnik, V.: The Nature of Statistical Learning Theory (2nd edn). Springer (1998)
21. Weiss, S., Indurkhya, N.: Lightweight rule induction. International Conference on Machine Learning, pp. 1135–1142 (2000)
22. Witten, I., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)