A C T A U N I V E R S I T A T I S L O D Z I E N S I S
FOLIA OECONOMICA 255, 2011

Mariusz Kubus[*]

THE INFLUENCE OF IRRELEVANT VARIABLES ON CLASSIFICATION ERROR IN RULES INDUCTION

Abstract. Typical *data mining* task is to extract unsuspected and systematic relations from the data, when there are no previously set expectations about the nature of this relations. When data sets are large and not collected for a purpose to answer the particular question, there are usually many irrelevant variables which may deteriorate the quality of discrimination model. In such situations feature selection methods are applied. In adaptive and nonparametric methods of discrimination (classification trees, rules induction) feature selection is a part of learning algorithm. Using simulations, the influence of irrelevant variables on classification error is examined in this methods.

Key words: irrelevant variables, discrimination, adaptive methods, rules induction.

I. INTRODUCTION

Together with a rapid progress of technology one can observe the emergence of huge databases. This has occurred in many areas of human activity, i.e. in wide understood economy we can list: supermarket transaction data, credit card usage records or telephone call details. The goal of such data analyse is to extract unsuspected and systematic relations from the data, when there are no previously set expectations about the nature of this relations. Researcher often deals with a data, that have been collected in some other purpose then his analyse. In such situations there are usually many irrelevant variables in the data what means in the case of discrimination that variables do not differ significantly in the classes. Great number of irrelevant variables may deteriorate the quality of discrimination model (error rate, interpretation), therefore various feature selection methods were proposed in the literature. Kohavi and John (1997) gave the classification of approaches to the feature selection problem. The first group of methods (so called filters) consists of variables rankings that constitute the pre-processing step. Among numerous criteria we can mention: Fisher's criterion or correlation criterion. The second approach uses learning algorithm to

[*] Ph.D., Department of Mathematics and Computer Science Application, Opole University of Technology.

evaluate the subset of variables (so called wrappers). The learning algorithm can be seen here as an inner loop of feature selection procedure. The task can be also formulated as a heuristic search of the space of all possible combinations of features with a function of criterion which evaluates the quality of learned model. The typical example is linear discriminant analysis with stepwise feature selection. The third approach concerns the situation where feature selection mechanism is an integral part of learning algorithm. As the examples we can give nonparametric and adaptive methods of discrimination such as classification trees or rules induction.

II. RULES INDUCTION

The rules induction derives from machine learning and developed in parallel to classification trees. There are many similarities between this two methods. They are both nonparametric, adaptive and can deal with metric and nonmetric variables. The model has a form of a set of rules. The conditional part of the rule is a conjunction of simple conditions. Usually they are equalities for nominal variables or inequalities for other variables. For example, in credit approval problem we can have the rule:

$$\begin{gathered} \textit{If } [\text{CURRENT ACCOUNT} < 0] \wedge [\text{PURPOSE} = \\ = \text{education}] \wedge [\text{HOUSING} = \text{own}], \\ \textit{then } \text{CREDIT} = \textit{no}. \end{gathered}$$

In such rule all predictors are not required. In practice there are this that locally and optimally separate classes. Similarity to one path from the root to the leaf in classification tree is obvious. Actually, classification tree can be perceived as a set of rules. The differences between this two methods are that rules generated by rules induction usually do not have a hierarchical structure of the tree and are learned in completely different way. To learn a single rule one uses heuristic search of the space of all possible conjunctions of simple conditions. Particular algorithms differ with search strategy, function of criterion and the solution of overfitting problem. The considerable number of them follow the separate-and-conquer scheme that was proposed for the first time by Michalski (1969). Set of rules is generated by learning one rule at a time. After each rule is learned (conquer step), the algorithm removes from the training set the cases for which the rule is true (separate step). Usually, only cases from the class which description we are learning are removed. The process is then iterated on the remaining training cases. It can be continued until no cases remain or some stopping criteria are fulfilled. On the other hand, Quinlan (1993) proposed to decompose classification tree and then prune it to obtain simpler model that

would better classify the new cases. Analogously to classification trees further development of this method focuses on ensemble approach.

As classification trees, rules induction methods had to solve the overfitting problem. The algorithms give the possibility of building models that ideally separate classes. However such models often weakly classify new cases especially when one deals with noisy domains. Rules which satisfy many cases from the target class and few cases from other classes have usually higher predictive accuracy on classifying new cases. There were various pruning techniques for overfitting avoidance proposed in the literature (see Fürnkranz 1999). All of them simplify the structure of the model. In rules induction we can list: removing single conditions from the rules, removing entire rules, stopping criteria which stop learning the rule or model. It is worth to pay attention that removing conditions (rules) has different structural consequences then pruning classification tree (see Fürnkranz 1999). Pruning techniques can be applied during learning the model or in post-processing step.

One of the first algorithm CN2 following separate-and-conquer scheme was proposed by Clark and Niblett (1989) and modified afterwards by Clark and Boswell (1991). It uses beam search to learn single rule what means using a few hill-climbing strategies in parallel. Such approach may avoid reaching the local optimum. For evaluation of rules quality CN2 uses entropy or Laplace function and for overfitting avoidance it uses the test of significance of the rules. Algorithm compares the observed distribution among classes of cases satisfying the conditional part of the rule with the expected distribution that would result if the cases occurred randomly. If they do not differ significantly the rule is not introduced to the model.

Advanced algorithm and considered as one of the best is RIPPER (Cohen 1995). It uses hill-climbing to learn single rule and very sophisticated techniques of pruning (combination of pre- and post-pruning). Additional stop criteria base on *Minimum Description Length Principle* (Rissanen 1978). Experiment with 21 real world data from UCI of Machine Learning Database (www.ics.uci.edu/~mlearn/MLRepository.html) showed that RIPPER was comparable with popular CART as regard the error rate (see Kubus 2009).

III. ENSEMBLES

Ensemble approach for classification trees gained a great interest because of error rate reduction and growth of stability. The basic idea is to learn many classifiers using different subsamples of the training set and then to aggregate the classifications of the ensemble members. In this way the information in training set is used many times. The different training subsamples can be obtained by randomly drawing from original training set or by using system of weights or by random selection of predictors (see Gatnar 2008).

AdaBoost algorithm (Freund, Schapire 1997) is commonly considered as one of the most effective tool of discrimination. Weighting the cases causes that algorithm focuses on previously misclassified cases in every iteration. It improves the predictive ability of the model. Other effective and popular algorithm is Random Forests (Breiman 2001). It uses not only bagging but also the variables are randomly selected at each node, so that to chose the best split only among them.

There are also a few proposition of ensembles in rules induction. The most popular are SLIPPER (Cohen, Singer 1999) and RuleFit (Friedman, Popescu 2005). SLIPPER uses the same scheme of learning rules as RIPPER but boosting is applied instead of separate step. Cases and rules are weighted and all training set is used to learn every single rule. SLIPPER usually outperforms CART but is not as effective as AdaBoost. RuleFit combines bagging, decomposition of trees to the rules and regularized linear regression. In the first step M classification trees are learned on different training subsamples. Their size is constrained by prespecified number of leaves. In second step trees are decomposed to the rules and linear model is build:

$$F(\boldsymbol{x}) = a_0 + \sum_{k=1}^{K} a_k r_k(\boldsymbol{x}), \tag{1}$$

where $r_k(\boldsymbol{x})$ yields 1 when the case $\boldsymbol{x}$ satisfy the rule $\boldsymbol{r}$ or 0 otherwise. Parameters of such model are estimated by minimization:

$$\{\hat{a}_k\}_0^K = \arg\min_{\{a_k\}_0^K} \left(\sum_{i=1}^{N} L\left(y_i, a_0 + \sum_{k=1}^{K} a_k r_k(\boldsymbol{x_i}) \right) + \lambda \sum_{k=1}^{K} |a_k| \right), \tag{2}$$

where L is a loss function. In practice, various iterative algorithms are applied for this task: gradient decent (see Friedman, Popescu 2004) or forward stagewise additive modelling with shrinkage (see Hastie et al. 2001).

IV. EXPERIMENT

For empirical experiment we used well known data set *Pima Indian Diabetes* from UCI Repository of Machine Learning Databases. The set contains 8 metric predictors, 768 cases (that were randomly divided into 512 cases for training and 256 cases for testing) and 2 classes. We chose this data because linear discriminant analysis (LDA) yields here the lowest error rate in comparison to adaptive methods (see Table 1). In this case it is especially

interesting how will the situation change when we add to the original data the irrelevant variables.

Table 1. Error rates (%) estimated on test set

Data set	LDA	CART	AdaBoost	Random Forests	CN2	RIPPER	SLIPPER	RuleFit
Pima	19,9	22,67	24,22	23,24	24,2	24,6	24,22	22,27

Source: own computations.

Thus we generated randomly 30, 60 or 100 irrelevant variables (in every class according to the same scheme) from various distributions: standarized normal distribution, mixture of normal distributions: ¼ cases from N(0,1) and the rest from N(5,1), exponential distribution with parameter $l = 2$; and added them into the original data set *Pima*. Then we examined the error rates estimated on test set and the number of irrelevant variables introduced to the model. All results are presented in the Tables 2-3.

Table 2. Error rates (%) for *Pima* with irrelevant variables from various distributions

Methods	Normal distribution			Mixture			Exponential distribution		
	number of irrelevant variables			number of irrelevant variables			number of irrelevant variables		
	30	60	100	30	60	100	30	60	100
LDA	23,44	23,44	26,95	25,00	26,17	31,64	22,66	24,22	25,78
LDA (fs)*	21,87	24,61	25,00	23,83	23,05	24,61	22,66	22,66	23,44
LDA (bs)**	22,66	24,61	24,61	22,66	23,05	24,61	22,66	22,66	22,27
CART	22,67	21,87	21,87	21,87	21,87	21,87	21,87	21,87	21,87
AdaBoost	22,27	21,87	21,87	21,87	22,66	22,66	21,87	22,27	23,44
Random Forests	22,46	23,14	24,71	23,10	24,52	25,61	23,09	24,45	26,00
CN2	27,00	25,00	30,47	29,30	27,70	29,69	27,34	26,56	29,30
RIPPER	20,31	21,09	25,39	25,00	21,09	24,22	23,83	23,83	22,66
SLIPPER	22,27	23,83	23,44	23,83	21,09	23,05	26,95	21,87	28,52
RuleFit	27,34	25,00	28,91	29,30	28,52	27,34	29,69	25,39	23,83

* LDA with forward stepwise feature selection.
**LDA with backward stepwise feature selection.
Source: own computations.

Table 3. Number of irrelevant variables introduced to the model

Methods	Normal distribution			Mixture			Exponential distribution		
	number of irrelevant variables			number of irrelevant variables			number of irrelevant variables		
	30	60	100	30	60	100	30	60	100
LDA (fs)*	7	3	5	9	4	7	7	1	5
LDA (bs)**	0	3	6	0	4	7	0	1	6
CART	0	0	0	0	0	0	0	0	0
RIPPER	0	1	2	2	0	0	1	4	1
SLIPPER	4	5	4	9	3	2	9	0	9

* LDA with forward stepwise feature selection.
** LDA with backward stepwise feature selection.
Source: own computations.

Except classical linear discriminant analysis we used LDA with stepwise feature selection (forward or backward). As the tables (2-3) show it is good remedy on irrelevant variables in comparison to classical LDA although the irrelevant variables are still introduced to the model. Evidently the best results were obtained by classification trees CART and secondly by boosted trees. Classification trees introduced no irrelevant variables to the model and yielded lower error rates then LDA (also with feature selection). The only exception was the case of 30 irrelevant variables from standarized normal distribution but LDA with forward stepwise feature selection introduced then 7 irrelevant variables. As regards rules induction algorithms, two of them (CN2 and RuleFit) completely do not cope with irrelevant variables. SLIPPER turned out to be better but still not always competitive with LDA with feature selection. It is also outperformed by CART. RIPPER yielded clearly better results then LDA when irrelevant variables were from standarized normal distribution. It was even a little better then CART but not in the case of 100 added variables. Surprising that such algorithms like CART, AdaBoost, RIPPER and SLIPPER yielded usually lower error rate after adding irrelevant variables then on original data.

V. CONCLUSIONS

The experiment showed that rules induction algorithms were sensitive on irrelevant variables. The only exception was RIPPER in the case when irrelevant variables were drawn from standarized normal distribution but even then the error rate increased when algorithm run with 100 artificially introduced variables. When one suspects many irrelevant variables in the data, CN2 and RuleFit algorithms are definitively not recommended. SLIPPER can be competitive with LDA but it introduces irrelevant variables to the model and is usually outperformed by CART.

The experiment also proved the view presented in the literature that classification trees are robust on irrelevant variables (see i.e. Hastie et al. 2001). They introduced no irrelevant variables and in the same time almost always yielded the lowest error rate in comparison to other methods.

REFERENCES

Breiman L. (2001), Random forests. „*Machine Learning*", 45, p. 5–32.

Clark P., Boswell R. (1991), Rule induction with CN2: some recent improvements. [in:] Kodratoff Y. (red.) *Machine learning – EWSL-91, European working session on learning*, p. 151–163, Springer Verlag, Berlin.

Clark P., Niblett T. (1989), The CN2 induction algorithm. „*Machine Learning*", 3(4), p. 261–283, Kluwer.

Cohen W.W. (1995), Fast effective rule induction. In Prieditis A., Russell S. (Eds.) *Proceedings of the 12th International Conference on Machine Learning*.

Cohen W.W., Singer Y. (1999), A Simple, Fast, and Effective Rule Learner. In *Proceedings of Annual Conference of American Association for Artificial Intelligence* (p.335–342).

Freund Y., Schapire R. E. (1997), A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *„Journal of Computer and System Sciences"*, No 55, p. 119–139.

Friedman J. H., Popescu B. E. (2004), Gradient directed regularization for linear regression and classification. (*Technical Report*). Dept. of Statistics, Stanford University

Friedman J. H., Popescu B. E. (2005), Predictive learning via rule ensembles. (*Technical Report*). Dept. of Statistics, Stanford University

Fürnkranz J. (1999), Separate-and-Conquer Rule Learning. *Artificial Intelligence Review* 13(1).

Gatnar E. (2008), *Podejście wielomodelowe w zagadnieniach dyskryminacji i regresji.* PWN, Warszawa.

Hastie T., Tibshirani R., Friedman J. (2001), *The Elements of Statistical Learning: Data Mining, Inferance, and Prediction.* Springer, New York

Kohavi R., John G. (1997), Wrappers for feature selection. *Artificial Intelligence*, 97(1–2): 273–324.

Kubus M. (2009), Porównanie indukcji reguł z wybranymi metodami dyskryminacji. [in:] K. Jajuga, M. Walesiak (red.), *Taksonomia 16, Klasyfikacja i analiza danych – teoria i zastosowania,* Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu, No 47, p. 367–374.

Michalski R.S. (1969), On the quasi-minimal solution of the covering problem. In *Proceedings of the 5th International Symposium on Information Processing* (FCIP-69), Vol. A3 (Switching Circuits), p.125–128 Bled, Yugoslavia.

Quinlan J.R. (1993), *C4.5 programs for machine learning.* Morgan Kaufmann, San Mateo.

Rissanen J. (1978), Modeling by shortest data description. *Automatica*, 14, p. 465–471.

Mariusz Kubus

WPŁYW ZMIENNYCH NIEISTOTNYCH NA BŁĄD KLASYFIKACJI W INDUKCJI REGUŁ

Typowym zadaniem *data mining* jest wykrycie niespodziewanych i systematycznych relacji w danych, gdy nie ma wcześniejszych oczekiwań co do natury tych relacji. W dużych zbiorach, które nie były zgromadzone w celu prowadzonej przez badacza analizy, zwykle występuje wiele zmiennych nieistotnych, co może obniżyć jakość modelu dyskryminacyjnego. W takich sytuacjach stosowane są metody selekcji zmiennych. W nieparametrycznych i adaptacyjnych metodach dyskryminacji (drzewa klasyfikacyjne, indukcja reguł) selekcja zmiennych jest częścią algorytmu uczącego. Za pomocą symulacji badany jest wpływ zmiennych nieistotnych na błąd klasyfikacji w tych metodach.