Classification Algorithms Based on Linear Combinations of Features

Dominik Ślęzak and Jakub Wróblewski

Institute of Mathematics, Warsaw University,
ul. Banacha 2, 02-097 Warsaw, Poland.
slezak@alfa.mimuw.edu.pl
jakubw@mimuw.edu.pl, http://alfa.mimuw.edu.pl/~jakubw

Abstract. We provide theoretical and algorithmic tools for finding new features which enable better classification of new cases. Such features are proposed to be searched for as linear combinations of continuously valued conditions. Regardless of the choice of classification algorithm itself, such an approach provides the compression of information concerning dependencies between conditional and decision features. Presented results show that properly derived combinations of attributes, treated as new elements of the conditions' set, may significantly improve the performance of well known classification algorithms, such as k-NN and rough set based approaches.

1 Introduction

Classification is the problem of forecasting the *decision* for new cases, basing on their *conditional* features, by comparison with already known instances. An exemplar classification technique is the *nearest neighborhood* approach [3]. Given some arbitrarily fixed distance measure ρ, defined over the Cartesian product of conditional features treated as real valued dimensions, we can find for a new example k ρ-nearest known cases $u_1, ..., u_k$ and classify it as belonging to the same decision class as that most supported by them. The efficiency of this approach depends obviously on the choice of distance type and the choice of conditions over which we define ρ. Namely, it turns out that sometimes it is even better to consider smaller subset of conditions, to obtain better classification results (see e.g. [1]).

Appropriate selection of conditions is the very important task with respect to practical applications, where it is more effective to base on smaller (or easier to be analyzed) groups of features. In the above k-NN approach such a selection is concerned just in view of the classification performance. There are, however, approaches where it is regarded as the main paradigm, enabling to focus not on the classification only, but also on the representation of the dependencies between conditions and decisions. One of them is the decision rules based method, developed within rough sets theory (see Section 2 for details and, e.g., [5] for further references). Although designed originally for discrete data, it can be applied to continuous conditions as well, by using *discretization* (see e.g. [4]) or *tolerance*

J.M. Żytkow and J. Rauch (Eds.): PKDD'99, LNAI 1704, pp. 548–553, 1999.

based techniques (see e.g. [7]), where by considering decision rules built with respect to *similarity* classes of ϵ-*almost* ρ-*same* objects, for a distance measure ρ and a similarity coefficient ϵ, we can obtain classification procedures with efficiency comparable to k-NN, having, however, more possibilities to express what actually happens in data.

It is worth noting that mentioned approaches are based not only on proper usage of already given conditional attributes but try to search for completely new means of expression. In fact, one may claim that the choice of ρ in k-NN actually defines new dimension for expressing the impact on decision in a better way. Also in the rough set based approach, discretization techniques themselves can involve *hyperplanes* (see Section 3 for their correspondence to the task of this work) into descriptors of decision rules or trees. Obviously, it depends on interpretation whether we treat the above examples as producing just classifiers proper for particular techniques or new attributes themselves. In this paper we propose alternative, very simple and intuitive way of automatic extraction of new features from data, as linear combinations of conditions which keep the original meaning of continuously valued attributes for sure. Foundations for searching for such combinations are strictly correlated to the task of classification and decision representation improvement.

Presented algorithms optimize quality measures which have strong theoretical background in the above mentioned hyperplane-based approach (as devoted to linear combinations itself) and indiscernibility characteristics being one of the most expressive tools of classical rough sets theory [6]. Resulting new conditions are possible to be applied not only to rough set based methods. They provide an intelligent preprocessing of data information rather than final classification system, what can be concluded from experiments described in Sections 4 and 5.

2 Rough Set Foundations

The main paradigm of rough sets theory [5] states that a universe of known objects is assumed to be the only source of knowledge used for classification of cases outside the sample. In applications, reasoning is usually stated as a classification problem, concerning distinguished decision attribute to predict under given conditions. By a *decision table* we understand a triple $\mathbf{A} = (U, A, d)$, where each attribute $a \in A \cup \{d\}$ is a function $a : U \to V_a$ from the universe U of objects into the set of all possible values on a. Classification of new objects outside U with respect to their membership to decision classes is performed by analogy with elements of U. In case of symbolic conditional attributes, we consider indiscernibility relation $IND(A) = \{(u_1, u_2) \in U \times U : Inf_A(u_1) = Inf_A(u_2)\}$. Information function $Inf_A(u) = \left(a_1(u), .., a_{|A|}(u)\right)$ yields a one-to-one correspondence between equivalence classes of $IND(A)$ and elements of the set $V_A^U = \left\{w_A \in V_A : Inf_A^{-1}(w_A) \neq \emptyset\right\}$ of all vector values on A supported by objects in U. If for a given $w_A \in V_A^U$ there is inclusion $Inf_A^{-1}(w_A) \subseteq d^{-1}(v_d)$ for some $v_d \in V_d$, we obtain a decision rule of the form $A = w_A \Rightarrow d = v_d$. Then,

given a new object with vector value w_A on A, we classify it as belonging to decision class $d^{-1}(v_d)$.

There are two main reasons for trying to decrease the number of conditional attributes used in rules. First, for less number of descriptors there is higher chance of their *applicability* to new cases and that we need to gather less information about them. The second reason is statistical – decision rules of the form $B = w_B \Rightarrow d = v_d$, for smaller $B \subseteq A$, are expected to classify new cases more properly, because of larger support in data. A lot of algorithms were developed to shorten descriptors $A = w_A$ to $B = w_A^{\downarrow B}$ in a way maximizing support $Inf_B^{-1}\left(w_A^{\downarrow B}\right)$, keeping (approximately) inclusion $Inf_B^{-1}\left(w_A^{\downarrow B}\right) \subseteq d^{-1}(v_d)$ on the other hand (see e.g. [8], [9]).

3 Hyperplanes and Linear Combinations

One of rough set based approaches to continuously valued conditions involves so called *discretization* [4]. Here, we would like to focus on a special technique for decision trees generation, basing on so called *hyperplane* cuts. In binary decision tree representation, the root is supported by the whole universe. Then, to each node we attach two sub-nodes, corresponding to its objects satisfying additionally inequalities $h(u) \geq c$ and $h(u) < c$, respectively. Formula $h(u) = h_1 a_1(u) + ... + h_n a_n(u)$ can be treated as describing a new continuously valued feature being linear combination of attributes $a_1, ..., a_n$. From this point of view, $c \in (\min(h), \max(h)]$, for $\min(h) = \min_{u \in U} h(u), \max(h) = \max_{u \in U} h(u)$ is a real cut generating two-interval discretization over $h = h_1 a_1 + ... + h_n a_n$.

The main aim of algorithms searching for decision trees with such hyperplane based cuts is to provide possibly best discernibility between decision classes with respect to membership to particular nodes. The fundamental discernibility measure evaluating pairs of linear combinations and their cuts is the following:

$$Disc(h, c) = \sum_{v_1 \neq v_2} \|u \in U : d(u) = v_1, h(u) < c\| \cdot \|u \in U : d(u) = v_2, h(u) \geq c\|$$

Trying to focus on optimization of linear combinations parameters "in general", not concerning with any particular cut, one must provide a quality measure reflecting potential ability of using them in various classifier systems. The first idea is thus to search for h corresponding to average $Disc(h, \cdot)$-best discretization cuts. The following measure

$$Q_1(h) = \sum_{u_1, u_2 : d(u_1) \neq d(u_2)} \frac{|h(u_1) - h(u_2)|}{\max(h) - \min(h)}$$

has its own interpretation in the search of combinations putting objects from different decision classes possibly far to each other. Moreover, it turns out to have much in common with average hyperplane cuts quality. Note, that:

$$Q_1(h) = \frac{1}{\max(h) - \min(h)} \int_{\min(h)}^{\max(h)} Disc(h, x)\, dx$$

Let us rewrite U according to increasing ordering $U = \langle u_{h,1}, ..., u_{h,N} \rangle$ induced by h. Assuming that there is no values of h which correspond to objects from different decision classes (otherwise, we delete all objects corresponding to such inconsistencies for a given h), we can easily set the minimal sequence of real values $\min(h) = c_{h,1} < ... < c_{h,k(h)} = \max(h)$ such that for each $i = 1, .., k(h) - 1$ there is inclusion $h([c_{h,i}, c_{h,i+1}]) \subseteq d^{-1}(v_i)$ for some $v_i \in V_d$, where

$$h([c_{h,i}, c_{h,i+1}]) = \{u \in U : c_{h,i} < h(u) \leq c_{h,i+1}\}$$

Intuitively, the number $k(h)$ corresponds to potential difficulty of handling decision rules based on h after discretization. Such coefficient, however, does not enable to search for proper combinations as optimization factor, because it abandons too much information. In our experiments we decided to consider the following measure:

$$Q_2(h) = \sum_{i=1}^{k(h)-1} \|h([c_{h,i}, c_{h,i+1}])\|^2$$

Searching for combinations h which maximize the above formula for $Q_2(h)$ corresponds, actually, to searching for new features which discern minimal number of pairs of objects from different decision classes, after necessary discretization.

4 Algorithmic Foundations

The problem of finding optimal linear combinations of attributes can be divided into two stages: determining which attributes the combination should be concerned with, and determining the proper coefficients of linear combination. In our experiments we solve the first problem by choosing k attributes randomly and finding their optimal linear combination. Quality of particular combinations can be expressed by different modifications of formulas Q_1 and Q_2. Experiments presented in the next subsection were performed for original Q_1 and Q_2 modified as follows, to improve results and take the best from both distance and discernibility based intuitions:

$$Q_{mod}(h) = \left(70 + \ln\left(\frac{\min_{d(u_i) \neq d(u_{i+1})} |h(u_i) - h(u_{i+1})|}{\max |h(u)|}\right)\right) Q_2(h)$$

Given a quality measure, we repeat this algorithm several (20 in our experiments) times and get the best linear combination found. The factor in brackets in formula for Q_{mod} is fixed, concerned with the minimal difference of h values between any two objects from different decision classes, which we should maximize. In fact, it was tuned to obtain possibly best classification performance, with respect to the search procedure described below.

Our task is to create an optimal (in a sense of quality measure) linear combination of the k selected attributes. We used an algorithm based on evolution strategies (see e.g. [2]). Note, that every (normalized) linear combination of k conditional attributes can be defined by $k-1$ angles (concerned with the direction of line representing this combination in k-dimensional space). Thus, the "individual" was composed of $k-1$ angle values. The objective function was based on Q_1 or Q_{mod} quality measure.

5 Experimental Results

Two databases was used for experiments: sat_image database (4435 training and 2000 test objects, 36 attributes) and letter_recognition database (15000 training and 5000 test objects, 16 attributes). Four new attributes was generated for each table: two of them as a linear combination of two selected attributes, two other was created basing on three selected attributes (experiments show, that considering more than three attributes hardly improves results, whereas the computation time grows dramatically). Both the training and test table was extended by four new attributes; only the training tables, however, were used to choose the linear combinations.

Then, the newly created data sets were analyzed using two data mining methods: k-NN (for k from 1 to 10; distances on all dimensions was normalized) and a rough set based analyzer using local reducts (see [9] for details). Table 1 presents results of classification of test tables of the databases extended by new attributes as well as containing only these new ones. In the case of local reducts based method there is a number of decision rules presented in the last column.

Table 1. Classification efficiency on the test data

Table name	Result (k-NN)	Result (local reducts)	No. of rules
sat_image	90.60%	81.30%	5156
extended, Q_1	90.30%	79.50%	3405
extended, Q_{mod}	**91.05%**	82.40%	1867
new attributes, Q_1	81.65%	64.50%	445
new attributes, Q_{mod}	84.30%	76.60%	475
letter_recognition	95.64%	79.64%	21410
extended, Q_1	92.00%	81.64%	17587
extended, Q_{mod}	**95.90%**	79.74%	15506
new attributes, Q_1	50.40%	45.40%	1765
new attributes, Q_{mod}	67.80%	70.84%	4569

Results show that in case of both k-NN and rough sets based method a table extended with four additional attributes can be analyzed more accurately. Moreover, even if only four additional attributes was taken into account, a classification can be done with a pretty good efficiency (e.g. 70.8% of correct answers in case of letter_recognition – this is good result if one take into account that

there is 26 possible answers). Note that in these cases we have 4 attributes instead of 36 or 16 – this is a significant compression of information.

The best results obtained in case of both sat_image and letter_recognition database are better than the best results reported in [3]. However, the result on sat_image is worse than one obtained using k-NN on feature subsets (91.5%, see [1]). The computation time on sat_image (calculation of the best set of four linear combinations): 64 min (Q_1), 31 min (Q_{mod}), on letter_recognition: 3 h (Q_1), 2 h 40 min (Q_{mod}). Calculations was performed on Pentium 200 MHz machine.

6 Conclusions

We provided theoretical and algorithmic framework for finding new features which potentially enable better classification. They were proposed to be searched for as linear combinations of already known continuously valued conditions. Quality measures for optimization of such combinations were shown to have strong intuition based on rough sets theory. Their tuning resulted with interesting experimental outcome, concerning classification task itself as well as representation of dependencies within data.

Acknowledgements This work was supported by KBN Scientific Research Grant 8T11C01011 and ESPRIT Project 20288 CRIT-2.

References

1. Bay, S.D.: Combining Nearest Neighbor Classifiers Through Multiple Feature Subsets. In: Proc. of the International Conference of the Machine Learning. Morgan Kaufmann Publishers, Madison, Wisc. (1998)
2. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer–Verlag (1994)
3. Michie, D., Spiegelhalter, D.J., Taylor, C.C. (eds.): Machine Learning, Neural and Statistical Classification. Ellis Horwood Limited (1994)
4. Nguyen, H.S.: From Optimal Hyperplanes to Optimal Decision Trees. Fundamenta Informaticae. Vol. 34 No. 1-2 (1998) 145–174
5. Pawlak, Z.: Rough sets – Theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht (1991)
6. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: R. Słowiński (ed.), Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory, Kluwer Academic Publishers, Dordrecht (1992) 311–362
7. Stepaniuk, J.: Approximation Spaces, Reducts and Representations. In: Polkowski, L., Skowron, A. (eds.), Rough Sets in Knowledge Discovery 2. Physica–Verlag, Heidelberg (1998) 109–126
8. Wróblewski, J.: Genetic algorithms in decomposition and classification problem. In: Polkowski, L., Skowron, A. (eds.), Rough Sets in Knowledge Discovery 2. Physica–Verlag, Heidelberg (1998) 471–487
9. Wróblewski J.: Covering with reducts – a fast algorithm for rule generation. In: Proc. of RSCTC'98, Warsaw, Poland. Springer–Verlag, Berlin, Heidelberg (1998) 402–407