Multiple and hybrid classifiers*

Jerzy Stefanowski

Institute of Computing Science,
Poznań University of Technology,
3A Piotrowo Street,
60-965 Poznań, Poland,
E-mail: Jerzy.Stefanowski@cs.put.poznan.pl

Abstract

Using multiple and hybrid classifiers results in improving classification accuracy and learning more difficult problems. This is now an active research area of machine learning and statistics. In this paper we have reviewed and discussed the relevant methods for combining classifiers including homogeneous, heterogeneous and hybrid learning algorithm. Then, we have presented two our approaches. The first one is a hybrid system that combines two representations of knowledge: decision rules referring to general trends in data, and single cases corresponding to exceptions or untypical situations. The second proposal is a n^2-classifier which is a specialized multiple classifier for solving multiclass learning problems. This approach is based on training $(n^2 - n)/2$ binary classifiers - one for each pair of classes. The motivations for each system, their architecture, learning, strength and limitations are discussed. Results of their experimental evaluation show that within their speciality these systems outpeforms standard single learning algorithms.

1 Introduction

One of the typical machine learning tasks is to create classification systems on the basis of the set of learning examples. Recently, in the area of machine learning and statistics, there has been observed an increasing interest in combining *multiple learning models* (homogeneous or heterogeneous) into one classification systems. Several motivations for such an integration can be discussed. First of all, one should notice that up to now the majority of research in machine learning, statistics or pattern recognition have been focused on creation and evaluation of the most effective but *single learning algorithms*. As a result a quite large number of different algorithms has been introduced. Many of them use different representations of input examples, classification knowledge and different underlying theories. As it is discussed in the literature and has been checked in empirical comparative studies [23] single algorithms are "specialized" to solve some classes of learning problems, but cannot perform the best in all situations. One of the solution to this limitation is to look for new methodologies allowing to an intelligent integration of different algorithms into one *composed* or *hybrid* system that can be used in diverse situations. Such an integration leads to improving *predictive accuracy*, it means that the composed system could better classify new (or testing) objects than his component models used independently.

The research in this quite new field has been started in the 90's [1, 3, 5, 14, 27]. The integration of multiple classification models has been approached in many ways. First of all, one can decide between using *homogeneous* or *heterogeneous classifiers*. In the other case, there are

*Published in L. Polkowski (ed.), Formal Methods and Intelligent Techniques in Control, Decision Making, Multimedia and Robotics. (Post-Proceedings of 2nd Int. Conference, Warszawa October 2000), Warszawa, 2001, 174-188.

also various proposals. One can use different learning algorithms on the same data set and their predictions could be combined in more or less sophisticated way (e.g. stacked generalization [35] or meta-learning [5]). Further on, due to the selective superiority of single learning algorithms one can create a *hybrid system* using multi-strategic learning principle that fits the data with different representations [22]. Yet another popular approach is to generate and combine multiple, homogeneous models by using different training distributions (e.g. boosting or bagging techniques [3, 9]). Moreover, the multiple classifiers can be trained over different samples or partitions of data sets, even distributed over computation nodes [6, 15].

The main aim of the following paper is to summarize the author and his coworkers proposals of multiple and hybrid systems [16, 17, 18, 32]. It compromises two different systems:

1. Hybrid system integrating representations and strategies of decision rules and case base learning; where decision rules express the typical and general decision situations while cases represent exceptional (untypical or difficult) cases,

2. The n^2-classifier that is a specialized multiple model to solve multi-class learning problem. It consists of several simple homogeneous, binary classifiers that are responsible for distinguishing between two classes.

The motivations for each system, their architecture, learning, speciality, strength and limitations will be discussed. Moreover, results of their experimental evaluation will be given.

The paper is organised as follows. In the next section, a brief review of multiple and hybrid models is given. Then, in section 3, the hybrid decision rule and case based system is discussed. Section 4, presents the idea of the n^2-classifier. Conclusions are drawn in the final section.

2 Multiple and hybrid classification systems

2.1 General issues

By the *combined classification systems* or *multiple classification models* we understand a set of classifiers whose individual predictions (classification decisions) are combined in some way to classify new examples.

The main question is how to create such models and when they may perform better than their components used independently. Some methodological studies have been already started. The first observation is that combining identical classifiers is useless. Thus a necessary condition for effective integration is that member models have a *substantial level of disagreement*, i.e. they make error independently with respect to one another [14]. Some other researchers proved that if single classifiers make errors completely independently then the error rate obtained by a combined system decreases proportionally to the number of classifiers [1, 14]. It is also required that each single model must perform better than a random guess.

Combining classification predictions from single models can be performed in various ways. In general, one can distinguish either *group* or *specialized decision making*. In the first method all base classifiers are consulted to classify a new object while the other method uses only these classifiers whose are expertised for this object. *Voting* is the most common method used to combine single classifiers [20]. Moreover, the vote of each classifier may be weighted, e.g. by the posterior probability referring to its performance on the training data.

2.2 Review of related methods

Several researchers have already proposed various architecture of combined classification systems but there are nearly no review or comparative studies. Below we present own classification of these systems which is inspired by discussion given in [11]. In general, we propose to distinguish combining homogeneous or heterogeneous classifiers. The first option uses the same learning algorithm as the basis for single classifiers. While the other generates classifiers using different learning models. A separate category refers to hybrid systems.

2.2.1 Heterogeneous classifiers

Stacked generalization

Wolpert [35] proposed the *stacked generalization* framework. This is a layered architecture. The classifier at the basic level receives as input the original data and each classifier outputs a prediction. Successive layers receive as input the predictions of the immediately preceding layer and the output is passed to the next layer. A single classifier at the most top level outputs the final prediction. Stacked generalization is an attempt to minimize the generalization error by using the classifiers in higher layers to learn the type of errors made by the classifiers immediately below.

Meta learning

Chan and Stolfo [5] presented two schemes for classifier combination: *arbiter* and *combiner*. Both schemes are based on *meta learning*, where a meta-classifier is generated from meta data on the basis of the predictions of the base classifiers. A combiner is partly similar to stacked generalization but uses more information. An arbiter is also a classifier and it is used to arbitrate among predictions generated by different base classifiers. The training set for the arbiter is obtained from all the available data using a special selection rule that identify examples whose classification cannot be predicted consistently using base classifier. There are also newest and more sophisticated proposals introducing arbiter/combiner trees computed in a bottom-up binary-tree fashion.

2.2.2 Homogeneous classifiers

These methods combine models generated by a single algorithm over diverse data sets. Various strategies were proposed for generation of different classifiers using the same learning algorithm. Most of them manipulate the learning set to generate multiple hypotheses. The learning algorithm runs several times, each time using a different distribution of learning examples. This technique works especially well for *unstable learning algorithms* - algorithms whose output classifier undergoes major changes in response to small changes in the learning data.

Bagging

This technique was introduced by Breiman [3] and produces new training sets by sampling with replacement. It is based on the idea of *bootstrap sampling* - each learning set has the same size but some examples do not appear in it while others may appear more than once. From each replication of the training set a classifier is generated. All classifiers are used to classify each example in the testing set, usually using a uniform vote scheme. Intuitively, the bagging taking a majority of several hypothesis has the effect of *reducing the variability* of individual hypothesis and may lead to more complex decision surface (for more details see also [11]).

Boosting

Freund and Schapire [10, 27] proposed a general method to convert a *weak learner* into the new one that achieves higher classification accuracy. The *boosting algorithm* maintains a weight for each example in the learning set that reflects its importance. Adjusting the weights causes the learning algorithm to focus on different examples leading to different classifiers. Boosting is an iterative algorithm and in each iteration the weights are adjusted in accordance with the performance of the corresponding classifier. The weight of the misclassified examples is increased. The final classifier aggregates the learned classifiers in each iteration by weighted voting. The weight of each classifier is a function of its accuracy. There exist several interesting empirical results of using boosting and bagging (cf. [25]).

Multiple sampling and partitioned data

These methods are based on averaging of solutions over many samples from data. The samples may be either random or result from partitioning data (disjoint or partly replicated). The field of knowledge discovery in large databases often favors these methods when standard approaches cannot be used due to huge amount of data. In such situations it is more effective to partition large data into subsets and apply separate (usually homogeneous) classifiers on them. Several methods were already proposed, see e.g. [6, 15, 20].

Error-correcting output codes

Yet another proposal is the methods of *Error-correcting output codes* - ECOC [7] designed to solve multi-class learning problems by solving two-class problems. ECOC represent classes with a set of output bits, where each bit encodes a binary classification corresponding to unique partition of the classes. Algorithms that use ECOC learn a function corresponding to each bit. All functions are then combined to generate class prediction.

There are also other approaches to creating homogeneous classifiers - see a review in [11].

2.2.3 Hybrid systems

Results of empirical comparisons of various learning algorithms [23] illustrate that each algorithm has its own certain *selective supperiority*. It performs the best for some, but none all problems. This is the motivation to design integrated algorithms that try to fit data with *heterogeneous representations*. It means that, different regions of input data are approximated using different types of models. Comparing to heterogenous classifiers one should notice that here we consider rather *one composed and integrated* system including internal components being different but cooperating models (and also different knowledge representations) while the heterogenous classifier is a *multiple ensemble* of classifiers that are more loosely connected. The hybrid systems are also connected with problematic of *multistartegic learning* [22].

There are several research on this topic, e.g. combinations of decision rules with other learning paradigms inside their nodes. In next section we review some proposals integrating decision rules and instance based learning. Review of some others proposals can be found in [22].

3 Hybrid system integrating decision rules and case-based learning

3.1 General motivations

The general motivation behind designing this system is an observation that each learning model performs better for some learning problems while worse for others. In general, there is no single algorithm the best for all tasks. Thus, an intelligent integration of various algorithms may lead to solving a wider class of learning tasks. Moreover, some problems can be of complex nature where different regions of the input space can be the best approximated using various types of models. So, it is interesting to fit such data with composed algorithm using heterogeneous representations.

In our proposal we focus an interest on an integration of *decision rule representation* and *instance-based representation*. Our idea is also inspired by psychological consideration published by Riesbeck and Shank in [26]. They said:

"*when an activity has been repeated often enough it becomes rule-like in nature. We do not reason from prior cases when well-establish rules are available. (...) When rule fails, the only alternative for its user is to create a case that captures that failure*".

It seems to be quite natural to adopt the above general concept in the following practical classification scheme presented as:

if a new coming case satisfies any rule from the set of decision rules
then apply solution from a rule with a highest priority
else adapt the solution from the most similar example from the set of exceptions.

where set of decision rules contains rules that describe standard or typical examples, and set of exceptions contains examples that represent exceptions or untypical examples

We can shortly summarize that scheme is based on a heuristic: "to solve a new problem, first of all try to use conventional rule-based approach, if it does not work try to remind the most similar situation in the past and adapt the old solution". Rules are used first because standard or typical examples appear more often, so a new example is likely to be a standard one.

This two-level knowledge representation can be obtained in various ways. For instance we could assume that due to certain methodology the set of learning examples is split into *two disjoint subsets*: the set of typical examples and the set of exceptions. The decision rules can be induced from the first set using one of the popular rule learning algorithm.

The split of the example set is quite difficult problem. It can be performed by either an expert, statistical approach (e.g. cluster analysis) or heuristics [36]. The first approach to splitting was considered for example in [33]. Another approach consists in special way of performing rule induction phase. For instance, in RISE system [8] there is a uniform representation of rules and exceptional examples (they are represented as maximally specific rules). There is no splitting algorithm - first all learning examples are turned into maximally specific rules, then the algorithm tries to generalize some of them into rules. The uniform representation of rules and exemplars is also used in the FCLS system [37].

3.2 Architecture and learning of the hybrid system

In our proposal, we follow a new way of specific rule induction. We chose a new rule induction algorithm Explore (originally proposed by Stefanowski and Vanderpooten [30, 21]), which induces from examples the set of all *strong* and *general* decision rules. Such decision rules describe rather examples representing general, we can say, typical cases (or even patterns) for given decision problem. Examples that are not covered by discovered rules can be treated as exceptions and are moved to the set of exceptions. In this approach splitting algorithm and the rule induction algorithm are joined together.

Let us shortly comment that we are not using algorithms already proposed in machine learning literature as they are mainly oriented on induction of, so called, minimal set of rules covering, ideally, all learning examples [12], while in our system we need an induction algorithm generating all other rules covering the most general examples. The algorithm Explore [21, 31] algorithm induces all decision rules which satisfy predefined requirements, regarding:

- the strength (representing the relative number of learning examples supporting the rule),
- the length of condition part (i.e. the number of elementary conditions),
- the level of confidence of the rule (defined as the ratio of all positive examples covered by the rule to all examples covered by the rule),
- as well as requirements to the syntax of conditions.

It may also be adapted to handle inconsistent examples either by using rough set approach [24, 28] or by tuning a proper value of the discrimination level. Induction of the rules is performed using a procedure which is *repeated* for each decision concept to be described. The concept may represent a class of examples or one of its rough approximations in case of inconsistent examples. The main part of the algorithm is based on a breadth-first exploration which amounts to generating rules of increasing size, starting from one-condition rules. Exploration of a specific branch is stopped as soon as a rule satisfying the requirements is obtained or a stopping condition SC, reflecting the impossibility to fulfill the requirements, is met.

A formal description of Explore is presented in [21, 31]. Let us comment that several extra pruning techniques are employed within the search strategy to discard not promising candidates as soon as possible. Stopping conditions are the main control parameters of this algorithm and can be defined according to various requirements. As it was discussed in [31] focusing mainly on the strength of rules will result in obtaining a limited number of rules, which cover certain subset of learning examples, that represent general information patterns in data, while leaving uncovered some more difficult and specific examples. In the hybrid system, the relative strength stopping condition is also chosen, i.e. the smallest percentage of positive examples from a given decision class that a candidate for a rule must cover. In other words the corresponding stopping condition for a complex of conditions $Cond$ currently examined is thus simply:

$$\text{SC: } \textit{relative strength}(Cond) < l$$

Having induced a set of decision rules, the remaining, uncovered learning examples are stored in the base of untypical cases. The use of these integrated representations for classifying new examples follows the scheme presented in section 3.1. In the first step, the new example is matched to the condition parts of the rules. In may results in one of the three situations:

1. single match, i.e. exactly one rule covers an example, the example is classified to the class indicated by the rule,

2. multiple match, i.e. several rules cover an example, according to the conflict resolution strategy, the example is classified to the class pointed by the rule with the largest value of Laplace-corrected accuracy [32],

3. no match, i.e. no rule covers an example - the set of stored examples is used to classify an example.

In the third situation, the underlying concept is to predict the classification of new examples on the basis of its similarity to already stored examples/cases. So, *k nearest* (the most similar) stored examples are considered, where k is a parameter specified by the user. The final classification is made with regard to class membership of considered examples (e.g. a new case can be assigned to the class that is the most frequent among the considered exemplars). The most common Euclidean distance metric is used for calculating the distance between examples.

The crucial issue of designing the hybrid system is determining a *threshold value* for a *stopping condition of relative strength* for Explore algorithm. In general, it should lead to a good compromise between classification accuracy of the integrated system and generality of the model (i.e. the number of rules and examples, average strength of the rule). As it was shown in [31], it is possible to iteratively use Explore with quite simple tuning procedure consisting in systematic varying relative strength observing evaluation measures. As it was examined experimentally, there exists a range of values for strength parameter that leads to a *good compromise* regarding these criteria number of rules, average rule strength, average length of the condition part of the rule, and criterion of classification accuracy.

In the hybrid system, a modified tuning of stopping conditions is used. The threshold values of relative strength are systematically changed in a stepwise way starting from the highest values until deterioration of at least one criterion is met. The important property of this scheme is also looking for different threshold values for each decision class (it may lead to the better classification accuracy than using the same value for all decision classes). Therefore, the procedure starts from some initial values and then iteratively tunes them for each decision class. The general scheme is presented below.

Procedure Tunethresholds
(out $\mathcal{L}$: set of thresholds of relative strength for all classes;
    $l(C)$: threshold of relative strength for class C);
begin
    assign initial values to $\mathcal{L}$;
    generate rules, identify exceptions and
    evaluate hybrid classifier HC;
    best $\leftarrow HC$;
    bestL $\leftarrow \mathcal{L}$;
    for each decision class C **do**
    repeat
        decrease $l(C)$;
        generate rules, identify exceptions and
        evaluate hybrid classifier HC;
        if (classaccuracy(*best*) better classaccuracy(HC)) and not(stoppingconditions) **then**

begin
best $\leftarrow HC$;
bestL $\leftarrow \mathcal{L}$;
end
until (classaccuracy($best$) - classaccuracy(HC) $> \delta$) or stoppingconditions;
$\mathcal{L} \leftarrow$ bestL;
end

Classaccuracy is an evaluation of classification accuracy of the hybrid system. If two structures of systems give a comparable classification accuracy, the more compact system - i.e. with less rules - is chosen as better; δ is an indifference threshold allowing to stop the process of building hybrid classifier when the accuracy decreases to much. *Stoppingconditions* can be understand in a wide sense - here we stop when the procedure generates too much rules for a given class or specificity of the class decreases below a threshold.

The initial values of thresholds $\mathcal{L}$ should be assigned to rather high values such that only few rules (ideally one) are induced per each decision class. Then, the threshold should be systematically reduced, i.e. a certain step should be calculated for each class. Such a way of looking for a proper structure of the hybrid system allows to obtain smaller models earlier what reduces computational costs. This is quite important while analysing large databases. It is also observed in some studies that simple decision rule models can outperform more complex ones.

3.3 Experimental evaluation

The use of the hybrid classifier is illustrated on solving the problem of concerning analysis of business credit applications [32]. This is a real life case study coming from one of Polish commercial banks. Having information about business credit applications described by 35 economical parameters and assigned to 5 groups of banking risk, we tried to built a hybrid system to evaluate the credit risk.

The initial values of thresholds $\mathcal{L}$ for algorithm were determined on the basis of the strongest rules in the set of decision rules induced by typical classification oriented algorithm LEM2 [12]. As a result of using the tuning procedure we were able to create a knowledge base consisting of 70 decision rules and 15 specific cases. The average relative strength of a rule was about 56% (this means that each rule covered 56% examples from the indicated class) and the average rule length was about 2.6 (i.e. rules usually did not have more than two or three conditions in their condition parts). Testing this representation in the standard leaving-one-out way we achieved the estimation of classification accuracy equal to 81%. For this studied problem it is the best classification result, i.e. the component parts of the hybrid system used alone gave smaller values of accuracy. Moreover, the comparative study showed that results obtained by other known machine learning algorithms (C4.5, LEM2, IBL,RISE) are worse, i.e. they provide worse classification accuracy and in case of rule induction algorithms - also weaker and longer rules. Classification results are summarized in Table 1. All estimations have been used by means of leaving-one-out validation technique [34].

Let us also notice that the presented results show that neither approach based exclusively on decision rules (Explore alone) nor based exclusively on examples (IBL) worked better then their hybrid connection. The results clearly illustrate the synergy of these two approaches. Here let us emphasize good results obtained for the stand-alone Explore algorithm, which may be contributed to the dominant role of rules in the classification process.

Similar good results were obtained for two other data sets HSV and Multiple injuries. Let us notice that such hybrid systems allow us not only to increase classification accuracy but also represent the acquired knowledge in a comprehensive way that can be easily inspected by humans. Its classification scheme is also consistent with human way of solving problems and therefore prediction decisions are well justified either by matched rules or nearest examples. The limitation of this proposal are larger computation costs for some data sets than using a single conventional learning algorithm.

Table 1: Classification performance of hybrid system versus other machine learning approaches

System	Classification Accuracy (%)	Size avg. no. of rules and/or examples
Hybrid system	81.1	70/15
Explore alone	76.67	64
IBL1	72.22	89
IBL2	60	38
IBL3	74.44	22
LEM2	73.33	16
c4.5tree	73.33	23 elements
c4.5rules	66.7	9
RISE	72.22	72

4 The n^2-classifier for solving multiclass learning problems

4.1 Motivations, architecture and learning of the n^2-classifier

In this proposal, we focus our attention on using multiple classifiers to solve *multiclass learning problems*. The multiclass learning problem involves finding a classification system that maps descriptions of training examples into a discrete set of n decision classes ($n > 2$). Although the standard way to solve multiclass learning problems includes the direct use of the multiclass learning algorithm such as, e.g. algorithm for inducing decision trees, neural network, or instance-based algorithm, there exist more specialized methods dedicated to this problem. As it is discussed in literature such approaches, e.g., one-per-class method, distributed output codes classification schemes, error-correcting techniques (ECOC) can outperform the direct use of the single multiclass learning algorithms (see, e.g. [5, 7]).

The n^2-classifier belongs to the group of multiple classification models dedicated to solve multiclass learning problems (i.e. with n decision classes where n >2). The n^2-classifier was defined by us in [16]. Below we give its short description.

It is composed of $(n^2 - n)/2$ base binary classifiers. The main principle of the n^2-classifier is the discrimination of each pair of the classes: (i, j), $i, j \in [1..n], i \neq j$, by an independent binary classifier C_{ij}. Each base binary classifier C_{ij} corresponds to a pair of two classes i and j only. Therefore, the specificity of the training of each base classifier C_{ij} consists in presenting to it a subset of the entire data set that contains only examples coming from classes i and j. The classifier C_{ij} yields a binary classification indicating whether a new example, $\mathbf{x}$, belongs to class i or to class j. Let us denote by $C_{ij}(\mathbf{x})$ the classification of an example $\mathbf{x}$ by the base classifier C_{ij}. In following description of the n^2-classifier we assume that $C_{ij}(\mathbf{x}) = 1$ means that example $\mathbf{x}$ is classified by C_{ij} to class i, otherwise ($C_{ij}(\mathbf{x})$= 0) $\mathbf{x}$ is classified to class j.

An algorithm providing final classification assumes that a new example $\mathbf{x}$ is applied to all base classifiers C_{ij}. As a result, their binary predictions $C_{ij}(\mathbf{x})$ are computed. The final classification should be obtained by a proper aggregation of these predictions. The simplest aggregation rule is based on finding a class that wins the most pairwise comparisons. The classification performance of base classifiers could be diversified because they have been trained on different pairs of classes. So, it is necessary to estimate the their credibility. It can be done in several ways. In this study we assume that with each classifier C_{ij} we associate a credibility coefficient P_{ij} defined in following way:

$$P_{ij} = \frac{v_i}{v_i + e_j}$$

where e_j is a number of misclassified examples from class j, and v_i is a number of correctly classified

examples from class i. The computation of the credibility coefficients is performed during the learning phase the n^2-classifier.
The process of classification of an example $\mathbf{x}$ can be described by the following steps:

1. Apply $\mathbf{x}$ to all base classifiers C_{ij}, and obtain their predictions $C_{ij}(\mathbf{x})$.
2. For each class i, $i \in [1..n]$, compute weighted sum:

$$S_i = \sum_{j=1, i \neq j}^{n} P_{ij} \cdot C_{ij}(\mathbf{x})$$

3. Return $argmax_i(S_i)$ as a final classification (break ties arbitrary in favor of the class that comes first in the class order).

The quite similar approach was independently introduced by Friedman [9]. Then it was extended and experimentally studied in [13]. Hastie and Tibshirani called the extended classification model as *classification by pairwise coupling*. As it has been indicated in experiments [9, 13] such integration of binary classifiers performs usually better than the single multiclass classification model.

A potential disadvantage of the n^2-classifier is that most decision concept should be learned $[\binom{n}{2}$ versus $n]$ each with less learning data than in the single multiclass approach. A compensation for this each of pairwise decisions is likely to be (much) simpler function of input attributes. This is especially when each decision class is well separated from most of the others. So pairwise decision boundaries between each pair of classes could be simpler and can be quite often estimated with linear functions while for the standard multiclass approach the decision boundary could be more complicated and more difficult to learn, e.g. with nonlinear approximators.

Friedman argues [9] that all function approximation methods are limited in that for each there are broad classes of (”complex”) target functions with which they have difficulty. Even for universal approximators the learning sample size may place such limits. In cases where the ”simpler” pairwise targets (in the sense of binary classification) more nearly match those that are amenable to the particular approximator being used, one might expect the pairwise polychotomous classification strategy to outperform the standard approach. Moreover, Friedman claims that success of the pairwise approach is connected with a bias reduction strategy. The bias of the learning algorithm reflects its inability to represent the decision concept as averaged over repeated (random) learning samples of the same size. The hope for the new approach is that the less complex pairwise decisions can be learned with less bias than individual ones. However, this property can be reduced because potential increase in variance when smaller samples are used to estimate decision concepts. This effect can be mitigated by the fact that most learning algorithms have smoothing (meta) parameters that can be adjusted to trade increased bias for reduced variance. Therefore, the relative merits of the proposed new approach depends on the specifies of particular problems as the true unknown decision functions, training sample size and the learning algorithm being used. Let us also remind that computation costs of training the n^2-classifier are higher than the standard approach.

4.2 Evaluating different base classifiers

The n^2-classifier is constructed using multiple and homogeneous binary base classifiers. As discussed in previous subsection, the crucial issue while its designing is the choice of learning algorithm to be used by base classifiers. Although several algorithms could be considered it is hypothesized that the expected improvement of classification accuracy may depend on both the particular problem and used proper base classifier.

In [16, 17] and now for this paper we performed series of computation experiments where the influence of the learning algorithm on classification performance was examined. Four different algorithms were used to learn:

Table 2: Performance of n^2-classifier based on decision tree (n^2_{DT}) and decision rules (n^2_{MODLEM})

No.	Name of data set	Accuracy of single DT (%)	Accuracy of n^2_{DT} (%)	Accuracy of MODLEM (%)	Accuracy of n^2_{MODLEM}
1.	Automobile	85.5 $\pm$ 1.9	87.0 $\pm$ 1.9	89.62 $\pm$ 1.3	90.69 $\pm$ 1.5
2.	Cooc	54.0 $\pm$ 2.0	59.0 $\pm$ 1.7	55.57 $\pm$ 0.7	59.51 $\pm$ 1.1
3.	Ecoli	79.7 $\pm$ 0.8	81.0 $\pm$ 1.7	78.63 $\pm$ 0.8	81.7 $\pm$ 1.8
4.	Glass	70.7 $\pm$ 2.1	74.0 $\pm$ 1.1	72.62 $\pm$ 2.2	74.02 $\pm$ 1.5
5.	Hist	71.3 $\pm$ 2.3	73.0 $\pm$ 1.8	–	–
6.	Meta-data	47.2 $\pm$ 1.4	49.8 $\pm$ 1.4	45.27 $\pm$ 0.9	49.96 $\pm$ 1.1
7.	Primary Tumor	40.2 $\pm$ 1.5	45.1 $\pm$ 1.2	40.9 $\pm$ 1.5	45.6 $\pm$ 1.9
8.	Soybean-large	91.9 $\pm$ 0.7	92.4 $\pm$ 0.5	90.61 $\pm$ 0.9	91.75 $\pm$ 0.4
9.	Vowel	81.1 $\pm$ 1.1	83.7 $\pm$ 0.5	78.91 $\pm$ 0.3	83.61 $\pm$ 0.8
10.	Yeast	49.1 $\pm$ 2.1	52.8 $\pm$ 1.8	54.72 $\pm$ 1.2	55.74$\pm$ 0.5

- decision trees (according to the Assistant version [4]),
- decision rules (using MODLEM algorithm [29]),
- neural networks (typical feed forward multi-layer network learned with backpropagation),
- instance based learning (standard approach based on k nearest neighbor principle).

All decision tree classifiers were trained in a unprunned manner. For artificial neural networks, we systematically checked various topologies of networks depending on the particular data, e.g. for data sets with smaller number of input features (ecoli, glass, vowel, yeast) we tested the following number of neurons in input and hidden layers: 8, 10, 12, 14. Moreover with each combination of these topologies we tested various number of epoch: 50, 100, 150, 250. It means that for each learning problem we systematically looked through 64 combinations to find the best learning parameters. The k-NN algorithm was used with k equal to 1 and standard Euclidence distance measure.

All computation experiments have been performed on the typical benchmark data sets. The most of them are coming from the Machine Learning Repository at the University of California at Irvine [2]. The classification accuracy was estimated by stratified version of 10-fold cross-validation technique, i.e. the training examples were partitioned into 10 equal-sized blocks with similar class distributions as in the original set.

For all learning algorithm we evaluated the classification performance of the n^2-classifier and compared to the single learning algorithm used once in a multiclass manner. The obtained results are summarized in Table 2 and Table 3, where each n^2-classifier is compared to use of a single multiclass algorithm.

Let us summarize the results obtained for the particular learning algorithms. In a case of applying the decision tree as a base classifier we can observe that in 8 of all (10) problems the integration of decision trees into the n^2-classifier results in significantly better classification accuracy than the direct use of multiclass single decision tree. For two remaining problems the improvement is indistinguishable. The highest improvement is observed for Cooc data set - 5.0%. Comparable improvements were observed for the case of decision rules. Similarly for neural networks the results show that the n^2-classifier performs generally better than single multiclass approach. The increase of classification accuracy is noticed in 9 of 10 data sets. Moreover, the improvements are relatively higher than for decision trees. Particularly high increase is observed for Vowel data - 26.3%. On contrary using IBL based on K-NN usually does not result in so much impressive improving classification ability of the n^2-classifier.

The obtained results showed clearly that the classification performance of the introduced n^2-classifier is generally better than the accuracy of single classifier approach. Let us also notice that experimental results presented in [9, 13] also indicate that coupling strategy improves the

Table 3: Performance of n^2-classifier based on neural network (n^2_{ANN}) and k-NN algorithm (n^2_{k-NN})

No.	Name of data set	Accuracy of ANN (%)	Accuracy of n^2_{ANN} (%)	Accuracy of k-NN (%)	Accuracy of n^2_{k-NN} (%)
1.	Automobile	52.6 $\pm$ 2.0	58.1 $\pm$ 2.3	77.7 $\pm$ 0.9	76.7 $\pm$ 1.0
2.	Cooc	56.0 $\pm$ 1.9	65.3 $\pm$ 0.7	68.4 $\pm$ 0.6	68.3 $\pm$ 0.6
3.	Ecoli	81.7 $\pm$ 1.7	83.0 $\pm$ 1.6	81.3 $\pm$ 0.5	81.3 $\pm$ 0.4
4.	Glass	62.7 $\pm$ 2.0	62.8 $\pm$ 0.8	68.8 $\pm$ 0.8	68.5 $\pm$ 1.0
5.	Hist	65.7 $\pm$ 3.0	83.3 $\pm$ 1.4	89.3 $\pm$ 0.5	89.3 $\pm$ 0.5
6.	Meta-data	50.5 $\pm$ 1.6	47.2 $\pm$ 1.5	40.6 $\pm$ 1.6	42.1 $\pm$ 1.6
7.	Primary Tumor	38.2 $\pm$ 1.5	43.4 $\pm$ 1.2	33.4 $\pm$ 1.2	36.2 $\pm$ 1.5
8.	Soybean-large	90.1 $\pm$ 0.8	92.9 $\pm$ 0.7	89.9 $\pm$ 0.4	89.9 $\pm$ 0.4
9.	Vowel	59.7 $\pm$ 2.4	86.1 $\pm$ 1.0	98.9 $\pm$ 0.2	98.9 $\pm$ 0.2
10.	Yeast	53.1 $\pm$ 1.4	59.0 $\pm$ 0.9	52.8 $\pm$ 0.7	53.3 $\pm$ 0.7

classification accuracy although the relative performance of different approaches depends on the problem.

Analysis of not so good performance for the instance-based learning leads us to hypothesis that it may be caused by the influence of irrelevant attributes. We think that algorithms with inherent capability of reducing the influence of irrelevant features (e.g. decision trees) could be more appropriate in the n^2-classifier than algorithms in which all features are treated as equally important. The *k-nn* algorithm is known to be sensitive to existence of irrelevant attributes.

This observation led us to a hypothesis that the use of dedicated subsets of features for discriminating each pair of decision classes could lead to higher accuracy than using all features for classifying all classes at once. Therefore, in [18] we extended the n^2-classifier for k-NN by connecting learning phase of its base binary classifiers with the technique of feature selection. We decided to implement it with the wrapper model where the search was performed by forward selection algorithm [19]. Let us stress that practically the wrapper model is now repeated $(n^2 - n)/2$ times in order to get dedicated input feature spaces for all pairs of classes. Then, these sets of features are used to construct the final n^2-classifier.

The results of this experiment [18] performed for k-NN based n^2-classifier with feature selection showed significant improvement of classification accuracy. It somehow confirms our initial hypothesis that creating of feature subspaces dedicated for discrimination of pairs of classes is more promising than using the same set of features for classifying all decision classes.

However, the considered approaches for feature selection are more demanding from the computational and memory point of view than simple approach without selecting features. Feature selection inside wrapper model increases computational costs mainly due to the number of features while the n^2-classifier approach is characterized by higher computational complexity due to the number of classes. Therefore it is worth to perform further research on effectiveness of the n^2-classifier approach for larger data sets with many decision classes.

5 Final remarks

Using multiple and hybrid classifiers results in improving classification accuracy and learning more difficult problems. This is now an active research area of machine learning and statistics. In this paper we have reviewed and discussed the relevant methods for combining classifiers including homogeneous, heterogeneous and hybrid learning algorithm. Then, we presented two our original approaches.

The first one is hybrid system that combines two representations of knowledge: decision rules referring to general trends in data, and the other representation in a form of single cases corresponding to exceptions or untypical situations. This system should be useful to learn problems of complex nature where different regions of the input space can be the best approximated by various types of models using heterogeneous representations. Moreover, the construction and use of this system follows human-like heuristic way of solving new decision problems, where first the human tries to use conventional general rules; if it does not work he looks for a specific similar problem solved in the past and tries to adapt it to the new problem. The important aspect of this system is choosing the algorithm Explore as a tool for inducing general and strong enough rules for a given data. Uncovered examples are used to create exceptional examples. It is also possible to tune parameters of this approach, i.e. thresholds values for requirements to the minimal rule strength that lead to good compromise values of evaluation criteria regarding classification accuracy of the integrated system, number of rules and cases and the average rule strength. We have shown experimentally that the new system outperforms standard approaches for some data sets. Let us also notice that the hybrid systems allow us not only to increase classification accuracy but also represent the acquired knowledge in a comprehensive way that can be easily inspected by humans.

The second proposal is n^2-classifier which is a specialized multiple classifier for solving multiclass learning problems. This approach is based on training $(n^2 - n)/2$ binary classifiers - one for each pair of classes. Final decision is obtained by a weighted probabilistic voting rule. We considered homogeneous classifiers so each binary classifier is based on the same type of learning algorithm. The performed discussion indicates that standard multiclass single algorithms are limited in that for each there are classes of ("complex") decision concepts with which they have difficulty. In cases where exist "simpler" pairwise decision boundaries between each pair of classes that can be easier approximated, one might expect that the n^2-classifier could outperform the standard approach. However, the relative merits of the proposed new approach depends on the specifies of particular problems as training sample size and the learning algorithm being used.

The extensive computation study on several benchmark data sets clearly showed that the classification performance of the n^2-classifiers based on decision trees, rules and neural networks is significantly better than the accuracy of the single classifier approach. The use of simple instance-based learning algorithm is not so encouraging. This can be explained by sensitivity of this algorithm to irrelevant attributes. Introducing feature selection technique inside base classifiers in the structure of the n^2-classifiers solved this limitations and experimental results showed an improvement of classification accuracy. It leads to a hypothesis that creating of feature subspaces dedicated for discrimination of pairs of classes is more promising than using the same set of features for classifying all decision classes.

In general the proposed approach leads to increasing learning accuracy. On the other hand the structure the multiple classifier seems to be much complicated and more difficult to interpret by human than the single model. Let us also remind that computation costs of training the n^2-classifier are higher than the standard approach. In particular it refers to the advanced architecture with selecting features. Feature selection inside wrapper model increases computational costs mainly due to the number of features while the n^2-classifier approach is characterized by higher computational complexity due to the number of classes. Therefore it is worth to perform further research on effectiveness of the n^2-classifier approach for larger data sets with many decision classes.

Acknowledgments
Research on this paper was partly supported from the KBN grant no. 8T11F 006 19.

References

[1] Ali, K., Pazzani, M., Error reduction through learning multiple descriptions. *Machine Learning* 24 (1), 1996.

[2] Blake C., Koegh E., Mertz C.J., Repository of Machine Learning. University of California at Irvine, 1999.

[3] Breiman, L., Bagging predictors. *Machine Learning*, 24 (2), 1996, 123-140.

[4] Cestnik, B., Kononenko, I., Bratko, I., Assistant 86, a knowledge elicitation tool for sophisticated users. In Bratko I., Lavrac N. (eds.) *Progress in Machine Learning*, Sigma Press, Wilmshow, 1987, 31-45.

[5] Chan, P.K., Stolfo, S.J., Experiments on multistrategy learning by meta- learning. In *Proceedings of the Second International Conference on Information and Knowledge Management*, 1993, 314-323.

[6] Chan, P.K., Stolfo, S.J., A comparative evaluation of voting and meta-learning on partitioned data. In *Proceedings of the 12th International Conference on Machine Learning*, San Francisco, 1995, 90-98.

[7] Dietterich, T.G., Bakiri, G., Solving muliclass learning problems via error- correcting output codes. *Journal of Artificial Intelligence Research*, 2, 1995, 263-286.

[8] Domingos, P., Unifying Instance-Based and Rule-Based Induction. *Machine Learning Journal*, 24, 1996, 141-168.

[9] Friedman, J.H., Another approach to polychotomous classification, Technical Report, Stanford University, 1996.

[10] Freund, Y., Boosting a weak learning algorithms by majority. *Information and Computation*, 121 (2), 1995, 256-285.

[11] Gama J., Combining classification algorithms. Ph.D., University of Porto, 1999.

[12] Grzymala-Busse J.W., LERS - a system for learning from examples based on rough sets. In R. Slowinski, (ed.) *Intelligent Decision Support*, Kluwer Academic Publishers, 1992, 3–18.

[13] Hastie, T., Tibshirani R., Classification by pairwise coupling, Proc. NIPS97.

[14] Hansen, L., Salamon, P., Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12 (10), 1990, 993-1001.

[15] Jelonek, J., Generalization capability of homogeneous voting classifier based on partially replicated data. Presented at *Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms. The 13th National Conference on Artificial Intelligence*, 5 August 1996, Portland.

[16] Jelonek, J., Stefanowski J., Using n^2-classifier to solve multiclass learning problems. Technical Report, Poznan University of Techonology 1997.

[17] Jelonek, J., Stefanowski J., Experiments on solving multiclass learning problems by the n^2-classifier. *Proceedings of 10th European Conference on Machine Learning*, Chemnitz, Germany, April 1998, Springer LNAI no. 1398, 1998, 172-177.

[18] Krawiec K., Jelonek, J., Stefanowski J., Comparative study of feature subset selection techniques for machine learning tasks. In *Proceedings of VIIth Intelligent Information Systems* IIS'98 Malbork,15-19 June 1998, IPI PAN Press Warszawa 1998, 68-77.

[19] Kohavi R., Sommerfield D., Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, Montreal, AAAI Press, 1995, 192-197.

[20] Littlestone, N. Warmuth, M.K., The weighted majority algorithm. *Information and Computation*, 108 (2), 1994, 212-261.

[21] Mienko R., Stefanowski J., Vanderpooten J., Discovery-Oriented Induction of Decision Rules. *Cahier du Lamsade* no. 141, Paris, Universite de Paris Dauphine, septembre 1996.

[22] Michalski, R.S., Tecuci, G., *Machine Learning. A multistrategy approach*. Volume IV. Morgan Kaufmann, 1994.

[23] Mitche D., Spiegielhalter C.J., Taylor C., *Machine Learning, Neural and Statistical Classification*, Ellis Horwood 1994.

[24] Pawlak Z., *Rough sets. Theoretical aspects of reasoning about data*. Kluwer Academic Publishers, Dordrecht, 1991.

[25] Quinlan J.R., Bagging, boosting and C4.5. In *Proceedings of the 13th National Conference on Artificial Intelligence*, 1996, 725-730.

[26] Riesbeck, C.K., Schank, R.C., *Inside Case-Based Reasoning*, Lawrence Erlbaum, Hillsdale, 1989.

[27] Schapire, R.E., Using output codes to boost multiclass learning problems. In *Proceedings of the 14th International Machine Learning Conference* 1997.

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

[29] Stefanowski J., The rough set based rule induction technique for classification problems, Proceedings of 6th *Euorpean Conference on Intelligent Techniques and Soft Computing Aaachen* EUFIT 98, 7-10 Sept. 1998, 109-113.

[30] Stefanowski J., Vanderpooten D., A general two stage approach to rule induction from examples. In W. Ziarko (ed.) *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer-Verlag, 1994, 317–325.

[31] Stefanowski J., Vanderpooten D., Induction of decision rules in classification and discovery-oriented perspectives, *International Journal of Intelligent Information Systems*, 1999, (to appear).

[32] Stefanowski J.,Wilk Sz., Minimazing Business Credit Risk by Means of Approach Integrating Decision Rules and Case Based Learning. Raport Badawczy Instytutu Informatyki PP, RA-001/99, February 1999 Also accepted to Journal of Intelligent Systems in Accounting, Finance and Management 2000.

[33] Surma, J., Vanhoof, K., Integrating Rules and Cases for the Classification Task. In*Case Based Reasoning Research and Development. Proceedings of the First International Conference* - ICCBR95, 1995, 325-334.

[34] Weiss S.M. , Kulikowski C.A., *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*, Morgan Kaufmann, 1991.

[35] Wolpert, D., Stacked Generalisation. *Neural Networks*, No. 5, 1992, 241-259.

[36] Zhang, J., Michalski, R.S., An Integration of Rule Induction and Exemplar-Based Learning for Graded Concepts. *Machine Learning*, Number 21, 1995, 235-267.

[37] Zhang, J., Selecting Typical Instances in Instance-Based Learning. In Proceedings of the Ninth International Workshop on Multistrategy Learning, 1992, 470-479.