Feature selection in the n^2-classifier applied for multiclass problems

Jacek Jelonek and Jerzy Stefanowski
Institute of Computing Science, Poznań University of Technology,
ul. Piotrowo 3A, 60-965 Poznań, Poland
e-mails: Jacek.Jelonek@cs.put.poznan.pl, Jerzy.Stefanowski@cs.put.poznan.pl

Abstract

The paper discusses solving multiclass learning problems by a multiple classification system, called the n^2-classifier. Its architecture and work is based on the set of binary classifiers - one for each pair of decision classes. This approach is extended by introducing feature selection for each base classifier. We consider two different approaches to conduct a search for the most relevant features. They are evaluated in an experimental study.

Keywords: machine learning, multiple classifiers, feature selection, genetic algorithms

1 Introduction

Creating systems that automatically learn from provided examples is one of the main problems considered in machine learning and artificial intelligence. In particular, one of common tasks is *supervised learning*, where learning process aims at finding function that assigns learning examples, each described by a fixed set of attributes (features), to known a priori decision classes. Such a function expresses knowledge obtained by an algorithm from examples and it can be successively used to classify new, previously unobserved, instances. In this sense learning process results in creating *classification system* – shortly called *classifier* [17]. Typical measure used to evaluate such systems is *classification accuracy*, i.e. percentage of correctly classified testing examples.

Recently, there has been observed a growing interest in increasing classification accuracy by integrating different classifiers into one composed classification system In proper circumstances such composed system should better classify new (or testing) examples than its component classifiers used independently. The integration of *multiple classifiers* has been approached in many ways, for some review see, e.g. [3, 6, 9, 15]. Experimental evaluations have confirmed that the use of multiple classifiers leads to improving classification accuracy in many problems.

In this paper, we focus our attention on specific multiple classifiers used to solve *multiclass learning problems*. This problem involves finding a classification system that assigns examples into n decision classes, where $n > 2$. Although the standard way to solve it includes the direct use of the multiclass learning algorithm such as, e.g. algorithm for inducing decision trees, neural networks, or instance-based algorithms, there exist however more specialized methods dedicated to this problem. Such approaches as, e.g. one-per-class method, error-correcting techniques (ECOC), or pairwise coupling can outperform the direct use of the single multiclass learning algorithms [4, 5].

Within the framework of multiple classifiers dedicated to solve multiclass learning problems, we introduced the new system, called n^2-*classifier* [11]. It is composed of $(n^2 - n)/2$ *binary base classifiers*. Each base classifier is specialized to discriminate respective pair of decision classes only. In the learning phase each base classifier is constructed by the same learning algorithm but on the subset of learning examples belonging to the given pair of classes. In the classification phase, a new example is classified by applying its description to all base classifiers. Then, their predictions are aggregated to a final classification decision using a majority voting rule.

In our former research, we were interested in the problem of choosing proper learning algorithm to create base classifiers in the n^2-classifier. In [11, 12, 16] we performed series of computation experiments where the influence of the choice learning algorithm on classification performance was examined. Four different algorithms were used to learn: decision trees, decision rules (using MODLEM algorithm [16]), neural networks and instance based learning (based on k nearest neighbor principle – abbreviated as $k-NN$). The obtained results showed that the classification accuracy of the n^2-classifier is significantly better than the accuracy of a respective single multiclass classifier (obtained by the algorithm of the same type as used to create base classifiers) for three algorithms: decision trees, rules and neural networks. On the contrary, using $k-N$ algorithm did not resulted in so encouraging improvement. We have suspected that its the worst performance could result from the fact that this learning algorithm treats all features (attributes) as equally important while other approaches (in particular decision trees and rules) have inherent capability of reducing the *irrelevant features* what may help with defining proper subspaces of features for efficient solving two-class problems.

In this paper, the n^2-classifier is extended by connecting learning phase of its base $k-NN$ classifiers with methods of *feature selection*. We consider two different approaches. In the first approach, the best feature subsets are looked for each pair of classes independently of other pairs. Moreover, these subsets are evaluated locally with the respect to their usefulness for correct discriminating examples from the given pair of classes. In the other approach the feature subsets are searched simultaneously for all pairs of classes and they are evaluated by the global classification accuracy. In the experiments, both these approaches are compared on several benchmark data sets against the single multiclass classifier and the n^2-classifier working with the complete set of features.

2 The n^2-classifier

2.1 *An architecture and learning of the n^2-classifier*

The n^2-classifier is composed of $(n^2 - n)/2$ base binary classifiers. The main idea is to discriminate 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 learning 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.

The complementary classifiers: C_{ij} and C_{ji} (where $i, j \in\ < 1 \ldots n >;\ i \neq j$) solve the same classification problem – they are trained on the same set of examples in order to discriminate between class i-th and j-th. So, they are equivalent ($C_{ij} \equiv C_{ji}$) and inside the architecture of the n^2-classifier it is sufficient to use only (n^2 - n)/2 classifiers $C_{ij}(i < j)$, which correspond to all combination of pairs of n classes.

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 more sophisticated approach may use a *weighted* majority voting rules, where the vote of each classifier is modified by its credibility. It may be calculated as its classification accuracy during learning phase, see e.g. [12].

The quite similar approach was independently introduced by Friedman [5]. Then it was extended and experimentally studied in [8]. Hastie and Tibshirani called the extended classification model as *classification by pairwise coupling.* As it has been indicated in experiments [5, 8] such an integration of binary classifiers performs usually better than the single classification model.

2.2 *Feature selection inside n^2-classifier*

The n^2-classifier may be more accurate than the standard multiclass single algorithms for such multiclass problems, where decision concepts are described by "complex" target functions which could be difficult to learn. In this case, each of pairwise decisions is likely to be (much) simpler function of input attributes (features) and can be easier approximated. We suspect that the success of learning of simpler and more accurate pairwise decision boundaries between each pair of classes is also connected with using the limited number of the most *relevant features* describing examples. The presence of too many irrelevant features may decrease the classification performance. Therefore some algorithms with capability of reducing the influence of irrelevant features (e.g. decision trees) could be more appropriate to use in the framework of the n^2-classifier than algorithms in which all features are treated as equally important. Indeed, the *k-NN* algorithm is known to be sensitive to existence of irrelevant features.

Our previous experimental studies [11, 12, 16] showed that the classification performance of the n^2-classifiers based on decision trees, rules and neural networks was significantly better than the single classifier approach, while the use of simple instance-based learning algorithm was not so encouraging. Therefore, we decide to extend the n^2-classifier based on $k{-}NN$ by connecting learning phase of its base classifiers with techniques of feature selection.

Let us notice that problem of feature selection also occurs for many standard learning algorithms, which may perform poorly when faced with too many irrelevant feature. Several methods have already been proposed, see e.g. review in [2, 13, 14]. In our classifier we employ, the so called, *wrapper model* [13], as a general framework. In this model, the given feature subset selection algorithm conducts a search for the good subset using classifier itself as a part of the *evaluation function.* It means that the subset of features is provided to train the classifier whose classification accuracy is estimated – usually by a cross-validation technique. Notice however, that classification accuracy of the final composed n^2-classifier, containing chosen sets of features, should also be evaluated on the verification examples, which were not used in the learning phase to select particular features [14].

We consider two different approaches to conduct a search for the best subsets of features within above framework.

The first approach is "*local*" in this sense that the feature subsets will be looked for each pair of classes independently of other pairs and these subsets will be evaluated by means of the classification accuracy calculated for examples belonging to this pair of classes only. The search algorithm, which looks through the space of feature subset for the given base classifier, will be a *forward stepwise selection.* It starts search with an empty set of features and successively adds the one with the best classification performance. The process of adding features is stopped when no improvement of the evaluation function is observed. As it is a version of this greedy choice, we extend this technique by introducing a *beam search* strategy. It maintains a fixed-size collection of the best performing subsets of features. This collection is updated in each iteration of the forward selection, for more details see [10]. This algorithm will be shortly called *FBFS.*

On the contrary to the previous approach ("local" one), in which features of all base classifiers are searched independently, in this "*global*" approach feature search is conducted for all base classifiers at the same time. Moreover, the feature subsets are evaluated by means of the classification accuracy calculated for the n^2-classifier and examples belonging to all classes. As even for small instances of feature search problems the number of possible solutions is huge, meta heuristic approach should be applied. In our case we decided to use a *genetic algorithm.*

Let us remind that a genetic algorithm is an iterative procedure that consists of a constant-size population of individuals, each one represented by a finite string of symbols, known as the *genome*, encoding a possible solution in a given problem space [7]. The standard genetic algorithm proceeds as follows: an initial population of individuals is generated at random or heuristically. Every evolutionary step, known as a generation, the individuals in the current population are decoded and evaluated according to some predefined quality criterion, referred to as the *fitness*, or fitness function. To form a new population (the next generation), individuals are selected according to their fitness. Usually individuals are selected with a probability proportional to their relative fitness. This ensures that the expected number of times an individual is chosen is approximately proportional to its relative performance in the population. Selection alone cannot introduce any new individuals into the population, i.e., it cannot find new points in the search space. These are generated by genetically-inspired operators, of which the most well known are *crossover* and *mutation*. Crossover is performed with a given probability between two selected individuals, called parents, by exchanging parts of their genomes to form two new individuals, called offspring. The mutation operator is introduced to prevent premature convergence to local optima by randomly sampling new points in the search space. It is carried out by flipping bits at random. Genetic algorithms are stochastic iterative processes that are not guaranteed to converge; the termination condition may be specified as some fixed, maximal number of generations or as the attainment of an acceptable fitness level.

In order to encode usage of the features for all base classifiers, each genome consists of a set of binary strings. A number of strings included in a genome equals to the number of the base classifiers. So, each string refers to a particular base classifier and indicates which features should be applied for it ('1' in the string means feature should be used and '0' otherwise). Evaluation of the genome is done by estimating of a classification accuracy of the n^2-classifier, which is created taking into account the features defined by the genome. This estimation is done on the basis of "inner" cross-validation, i.e. it uses only samples from the learning set of the main cross-validation loop. The crossover operator is applied for each string of a parents genome and exchange them after a randomly selected crossover point. The mutation operator randomly flips one bit in each string in the genome according to the given probability. The genetic algorithm is terminated after predefined number of generations.

3 Experiments

The aim of the experimental study is to check how much two different techniques of feature selection, discussed in this paper, can increase classification accuracy of the n^2-classifier. The $k-NN$ algorithm was employed to create classifiers in all experiments. We compare performance of:

1. The single classifier used for all features without any selection.

2. The n^2-classifier, where base classifiers are learned with the complete set of features.

3. The n^2-classifier with "local" feature selection performed by the FBFS algorithm.

4. The n^2-classifier with "global" feature selection performed by the genetic algorithm.

All experiments have been performed on the following benchmark data sets (concerning multiclass problems): *Automobile, Ecoli, Glass, Meta-data, Yeast* and *Zoo*. Their characteristics is given in the Table 1. They are coming form Machine Learning Repository at the University of California at Irvine [1]. Two original data sets were slightly modified. For *Meta* data set the continuous decision attribute was discretized using the following cut points: 6, 13, 20 and 50. In case of *Automobile* data set, examples containing missing values were removed. 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. In case of employing the wrapper model inside the n^2-classifier, the additional "inner" cross validation was used on the learning sets to evaluate feature subsets. Classification accuracies for all four compared classifiers are presented in the Table 2.

Table 1. Data sets used in the experiments

Data set	Number of examples	Number of classes	Number of attributes
Automobile	159	6	43
Ecoli	336	8	7
Glass	214	6	9
Meta-data	528	5	43
Yeast	1484	10	8
Zoo	101	7	16

Table 2. Classification accuracy of compared $k-NN$ based classifiers (expressed in %)

Data	Single classifier	n^2 all features	n^2 with FBFS	n^2 with gen. alg.
Automobile	77.9	76.7	85.9	86.1
Ecoli	81.0	81.3	82.0	81.5
Glass	68.8	68.5	71.0	70.5
Meta	40.6	42.1	48.0	47.8
Yeast	52.7	53.3	52.9	54.0
Zoo	96.9	96.9	93.9	97.0

4 Conclusions

Solving multiclass learning problems by the n^2-classifier is discussed. Its architecture and work is based on $(n^2 - n)/2$ binary homogeneous classifiers – one for each pair of classes. In this paper the n^2-classifier has been extended by introducing feature selection for each pair of classes. Two different approaches for selecting features, "local" and "global", have been introduced in the binary classifiers trained by instance-based learning algorithm ($k-NN$). The results of their experimental evaluation are discussed in the following points:

1. In general, the classification performance of the n^2 classifier based on $k-NN$ (with all features) and the single classifier is comparable.

2. The use of features selection performed by the $FBFS$ algorithm improves the classification accuracy, comparing to the single multiclass classifier, on 4 of 6 data sets; The highest improvements are for *Automobile* and *Meta data*.

3. The use of features selection performed by the genetic algorithm improves the classification accuracy, comparing to the single multiclass classifier, on 4 of 6 data sets; The highest improvements are also observed for *Automobile* and *Meta data*.

4. Comparing FBFS and genetic versions of the feature selection we observe the similar classification results.

Let us comment that the improvement of classification accuracy for the $k - NN$ based n^2-classifier was obtained in case of adding feature selection technique only. It confirms our hypothesis that looking for dedicated subsets of features, which discriminate pairs of decision classes, is crucial in the n^2-classifier. Let us remind that in our previous experiments such learning algorithms, which have inherent capability of reducing irrelevant features (e.g. decision trees or rules), also led to an improvement of the classification accuracy for the n^2-classifier [12]. These results could open a new interesting research problem concerning *constructive induction* in the n^2-classifier, i.e. transformation of original features into new more discriminative ones.

Another interesting research issue concerns rather small difference between the use of both feature selection techniques. The intuition could indicate that the "global" approach, focused on more global evaluation function, should lead to much better results than the "local" one. However, the "local" approach is accurate at the similar level (except data set *Zoo*) but it is much simpler and less demanding from the computation points of view.

To conclude, the use of then^2-classifier leads to increasing classification accuracy for multiclass learning problems. However, the structure of the multiple classifier seems to be more complicated and difficult to interpret by human than the single model. Let us also remind that computation costs of training the n^2-classifier with feature selection is higher than the standard approach. On the other hand, one can gain increased accuracy, which would be worth these additional costs for some problems.

Acknowledgment: The authors want to acknowledge support from State Committee for Scientific Research, research grant no. 8T11F 006 19. The software for this work used the GAlib genetic algorithm package written by Matthew Wall at MIT.

References

[1] Blake C., Koegh E., Mertz C.J., Repository of Machine Learning, University of California at Irvine 1999 [URL: http://www.ics.uci.edu/ mlearn/MLRepositoru.html].

[2] Dasf M., Liu H., Feature selection for classification. *Intelligent Data Analysis*, **1**(3) (1997), 131–156.

[3] Dietrich T.G., Ensemble methods in machine learning. *Proc. of 1st Int. Workshop on Multiple Classifier Systems* (2000), 1–15 .

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

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

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

[7] Goldberg D.E., *Genetic algorithms in search, optimization and machine learning*, Addison-Wesely Publishing, Massachusetts, 1989.

[8] Hastie T., Tibshirani R., Classification by pairwise coupling. *Proceedings* NIPS97, 1997.

[9] Jelonek J., Zastosowanie zlozonego systemu klasyfikacyjnego n^2 z mechanizmem konstruktywnej indukcji cech dla wieloklasowych problemow uczenia maszynowego, Ph.D. Thesis, Poznan University of Technology, 2000.

[10] Jelonek J., Stefanowski J., Feature subset selection for classification of histological images. *Artificial Intelligence in Medicine*, **9**, 1997, 227-239.

[11] Jelonek, J., Stefanowski J., Using n^2-classifier to solve multiclass learning problems. Technical Report, Poznan University of Techonology, no RB-01/97, November 1997.

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

[13] John G., Kohavi R., Pfleger K., Irrelevant features and the subset selection problem. *Proceedings of the Eleventh International Machine Learning Conference*, New Brunswick NJ, Morgan Kaufmann, 1994, 121-129.

[14] 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.

[15] Stefanowski J., Multiple and hybrid classifiers. *Formal Methods and Intelligent Techniques in Control, Decision Making, Multimedia and Robotics*, Post-Proceedings of 2nd Int. Conference, Warszawa, Polkowski L. (Ed.) 2001, 174–188.

[16] Stefanowski J., *Algorithims of rule induction for knowledge discovery.* (In Polish), Habilitation Thesis published as Series Rozprawy no. 361, Poznan Univeristy of Technology Press, Poznan, 2001.

[17] 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, San Francisco, 1991.