ACTA UNIVERSITATIS LODZIENSIS
FOLIA OECONOMICA 235, 2010

Waldemar Wołyński[*]

EFFECTIVENESS OF DECOMPOSITION ALGORITHMS FOR MULTI-CLASS CLASSIFICATION PROBLEMS

Abstract. The problem of predicting of the class (group, population) label is called classification, discrimination, or supervised learning. The set of labels consists of K>2 elements in the case of the multi-class problems and of K=2 elements in the two-class (binary) problems. Since the two-class problems are much easier to solve, than the multi-class problems (furthermore, some classification algorithms may apply in the two-class case only) many authors propose to reduce a multi-class classification problem to a set of binary classification problems.

Orthogonal to the decomposition problem is the problem of subject classifying which is described by the labels assigned by each binary classifiers.

In this paper different decomposition algorithms and both well-known and new methods of combining the information from binary classifiers will be compared. Results of these comparisons are pointing explicitly, that apart from simplifying procedures we are also getting the significant improvement in the quality of classification, especially for the „unstable" classification procedures such as classification trees or neural networks. Hence these methods may be recognized as the techniques of boosting classifiers.

Key words: Binary classifiers, Decomposition algorithms, Aggregation algorithms.

I. INTRODUCTION

In the classification problem one wishes to assign an individual described by the vector $\mathbf{x}=(x_1,x_2,\ldots,x_p)$ of p observations to one of $K\geq 2$ classes (populations, groups) $G_1,G_2,\ldots,G_K$. The classification problem with the number of classes greater than two is called the multi-class problem. In two classes case we will be calling the classification problem as the binary classification problem.

The solution of this problem leads to the classifier $d(\mathbf{x})$ which takes values in the set of the class labels $\{1,2,\ldots,K\}$. The construction of each classifier is based on a learning sample of the form $L_n=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\ldots,(\mathbf{x}_n,y_n)\}$, where $\mathbf{x}_i$ are vectors of observations, and y_i are the class labels $(i=1,2,\ldots,n)$. There are many well-known multi-class classification algorithms. Among them

[*] Ph.D., Faculty of Mathematics and Computer Science, Adam Mickiewicz University of Poznań.

are linear and quadratic Bayes classifiers, classification trees, neural networks and so on. Assume that as a result of learning process we receive the estimators of the class posterior probabilities $p_k(\mathbf{x}) = P(\mathbf{x} \in G_k \mid \mathbf{x})$, where $k = 1,2,\dots,K$. In this case we are defining the (multi-class) classifier $d_M(\mathbf{x}) \equiv d_M(\mathbf{x}; L_n)$ as

$$d_M(\mathbf{x}) = \arg\max_{1 \le k \le K} p_k(\mathbf{x}). \tag{1}$$

As two-class problems are much easier to solve, many authors propose to use two-class classifiers for multi-class classification (see e.g. Price et al. (1995), Dietterich and Bakiri (1995), Friedman (1996), Hastie and Tibshirani (1998), Moreira and Mayoraz (1998), Jelonek and Stefanowski (1998), Krzyśko and Wołyński (2008). In fact, different reasons motivate the decomposition of a large scale problem into smaller subproblems dealing with only two classes. On the one hand, some algorithms do not scale up nicely with the size of the training set. Others are not suited to handle a large number of classes. On the other hand, even when using an approach which can deal with large scale problems, an adequate decomposition of the classification problem into subproblems can be favorable to the overall computational complexity as well as to the generalization ability of global classifier.

The Section 2 contains the details of decomposition and aggregation algorithms. Results of comparing the performance of the multi-class classifiers with algorithms discussed in the Section 2 are described in Section 3.

II. DECOMPOSITION AND AGGREGATION ALGORITHMS

All algorithms discussed here consists with two steps: decomposition and aggregation. We can described the decomposition step by a $K \times L$ code matrix D. The K rows of this matrix represent classes and the L columns represent binary classifiers. In each binary subproblem elements of training sample receive a new label - 0 or 1. In this way we construct two super-classes. Elements which don't take part in construction of specific binary classifier are marked in the code matrix by "-". Three well-known decomposition patterns are: OPC (one-per-class), ECOC (error-correcting output codes) Dietterich and Bakiri (1995) and PWC (pairwise coupling) Hastie and Tibshirani (1998).

In the OPC scheme we train each binary classifier using as one super-class (labeled by 1) the training examples which belong to one class, and as second super-class (labeled by 0) all other training examples. In this case $L = K$. For example, in the 4-classes case, the decomposition matrix has a form:

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Let $p_l(\mathbf{x}) = P(d_l(\mathbf{x}) = 1 \,|\, \mathbf{x})$ be the estimator of the class posterior probability for lth binary classifier $(l = 1,2,\ldots,L)$. Then the aggregated classifier for the OPC scheme has a form

$$d_{OPC}(\mathbf{x}) = \arg\max_{1\le l\le L} p_l(\mathbf{x}). \tag{2}$$

In the ECOC scheme kth row of the matrix D is treated as a *codeword* connected with class G_k. Because in this procedure each binary classifier is trained on all learning samples the *codeword* for the class G_k has a form $w_{k1}w_{k2}\cdots w_{kL}$, where $w_{kl} \in \{1,0\}$. For small number of classes a code matrix may consists of all possible columns with zeros and ones (without complements and the all-zeros or all-ones columns). So, the number of columns (binary classifiers) in this scheme is $L = 2^{K-1} - 1$. For 4-classes case, $L = 7$ and the code matrix D has a form:

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

For large K the number of binary classifiers in very huge (e.g. for $K = 10$ the number of binary classifiers is $L = 511$). In this case we select columns in the code matrix in order to avoid misclassifications. A measure of the quality of the code matrix is the minimum Hamming distance between any pair of codewords. Typically a random generation of the codewords is a reasonably good method for obtained a proper code matrix.

The observation $\mathbf{x}$ has a codeword $p_1(\mathbf{x})p_2(\mathbf{x})\cdots p_L(\mathbf{x})$, where $p_l(\mathbf{x})$ are the estimators of the class posterior probability for lth binary classifier $(l = 1,2,\ldots,L)$. The aggregated classifier for the ECOC minimizing the generalized Hamming distance between the observation $\mathbf{x}$ and class G_k, so

$$d_{ECOC}(\mathbf{x}) = \arg\min_{1\le k\le K} \rho_{GH}(\mathbf{x}, G_k), \tag{3}$$

where

$$\rho_{GH}(\mathbf{x}, G_k) = \sum_{l=1}^{L} | p_l(\mathbf{x}) - w_{kl} |, \quad 1 \le k \le K.$$

In PWC scheme each binary classifier is trained on learning examples from two classes only. So, the number of possible binary classifiers is $L = K(K-1)/2$. For 4-classes case, $L = 6$ and the code matrix D is

$$D = \begin{pmatrix} 1 & 1 & 1 & - & - & - \\ 0 & - & - & 1 & 1 & - \\ - & 0 & - & 0 & - & 1 \\ - & - & 0 & - & 0 & 0 \end{pmatrix}.$$

For this decomposition scheme, many methods of combining the information from binary classifiers were discussed. Hastie and Tibshirani (1998) suggested using technique based on Kullback-Leibler distance. Their method leads to aggregated classifier of the simple form:

$$d_{HT}(\mathbf{x}) = \arg\max_{1\le i\le K} \sum_{j\ne i} p_{ij}(\mathbf{x}). \tag{4}$$

where $p_{ij}(\mathbf{x})$ is an estimator of the probability $P(\mathbf{x} \in G_i \mid \mathbf{x}, \mathbf{x} \in G_i \cup G_j)$.

Another approach was proposed by Price et al. (1995). Their classifier has a form:

$$d_P(\mathbf{x}) = \arg\max_{1\le i\le K} \left(\sum_{j\ne i} p_{ij}^{-1}(\mathbf{x}) - K + 2 \right)^{-1}. \tag{5}$$

The quality of the classifier (4) may be improved. Moreira and Mayoraz (1998) proposed an additional factor separating combined classes G_i and G_j from all other classes.
In this case

$$d_{MM}(\mathbf{x}) = \arg\max_{1\le i\le K} \sum_{j\ne i} q_{ij}(\mathbf{x}) p_{ij}(\mathbf{x}), \tag{6}$$

where $q_{ij}(\mathbf{x})$ is an estimator of the probability $P(\mathbf{x} \in G_i \cup G_j \mid \mathbf{x}, \mathbf{x} \in G_1 \cup G_2 \cup \cdots \cup G_K)$.

Another type of aggregated classifier was proposed by Jelonek and Stefanowski (1998). They have assumed that with each binary classifier there is associated a credibility coefficient $t_{ij} = v_i/(v_i + e_j)$, where e_j is a number of misclassified examples from class G_j, and v_i is a number of correctly classified examples from class G_i. Note that the computation of the credibility coefficient was performed during the learning phase of constructing of this classifier. Finally, classification decision has the form

$$d_{JS}(\mathbf{x}) = \arg\max_{1 \le i \le K} \sum_{j \ne i} t_{ij} p_{ij}(\mathbf{x}). \tag{7}$$

Additional methods of combining the information from binary classifiers were discussed by Krzyśko and Wołyński (2008). They proposed to use product instead of sum, and discussed different types of coefficients separating combined classes. Two of them were used in computational experiments.

$$d_{KW1}(\mathbf{x}) = \arg\max_{1 \le i \le K} \prod_{j \ne i} p_{ij}(\mathbf{x}). \tag{8}$$

$$d_{KW2}(\mathbf{x}) = \arg\max_{1 \le i \le K} \prod_{j \ne i} q_{ij}(\mathbf{x}) p_{ij}(\mathbf{x}). \tag{9}$$

III. COMPUTATIONAL EXPERIMENTS

To compare the performance of the aggregated classifiers with the standard multi-class classifiers we have made several experiments. We performed experiments using the following multi-class data sets from UCI Machine Learning Repository Merz and Murphy (1998): car, vowel. iris, new-thyroid, waveform, and wine, school and crudeoil from Johnson and Wichern (1982), risk from Dilon and Goldstein (1984), threeclass from Hastie and Tibshirani (1998), data-sym from Mojirsheibani (2002) and PlVowel from Jassem (1997). In Table 1 the characteristics of the data sets are given, showing the variety of training set sizes, number of classes and dimensionality.

Table 1. Data sets used in the experiments

No.	Data set	Number of examples	Number of classes	Number of features
1	Car	1728	4	6
2	CrudeOil	56	3	5
3	Data-Sym	150	3	2
4	Vowel	990	11	10
5	Iris	150	3	4
6	PlVowel	1130	6	5
7	Risk	87	3	2
8	School	85	3	2
9	ThreeClass	300	3	2
10	New-Thyroid	215	3	5
11	Waveform	900	3	21
12	Wine	178	3	13

In the experiments we investigated the classification errors obtained by the multi-class classifier, OPC classifier, ECOC classifier and six different standard and modify PWC classifiers. The classification errors were estimated by 10-fold cross-validation technique. The validation technique was repeated 10 times for each data set. We have taken into consideration six different classification procedures:

- ldc - linear Bayes normal classifier.
- qdc - quadratic Bayes normal classifier.
- parzendc - Parzen density based classifier.
- naivebc - naive Bayes classifier.
- treec - binary decision tree classifier.
- rnnc - random neural net classifier.

In computational process we used PRTools 4.0 program (http://www.prtools.org). PRTools is a Matlab based toolbox for pattern recognition van der Heijden et al. (2004). In each procedure we used the default parameters. For example: ldc - without regularization, parzendc - the bandwidth *h* is estimated from data, treec – the impurity of the split is the Gini index; without pruning.

Table 2 describes the means of the classification errors from 12 data sets for all discussed classifiers.

Table 2 shows also that the different classification procedures provide different values of classification errors. For example, the mean classification error for the multi-class linear Bayes method is equal 0.1584 but for the procedure based on the Parzen estimator of the density function this error is equal to 0.0810. Hence, it seems that the better measure of quality of eight aggregating classifiers taking into consideration in this paper will be the relative error:

$$RE = \frac{\text{error for multi - class classifier} - \text{error for aggregated classifier}}{\text{error for multi - class classifier}} 100\%.$$

Table 2. The means of the classification errors from 12 data sets (with standard deviation in brackets)

Classifier	Classification procedure					
	ldc	qdc	parzendc	naivebc	treec	rnnc
d_M	0.1548 (0.0052)	0.1005 (0.0063)	0.0810 (0.0061)	0.1748 (0.0085)	0.1519 (0.0154)	0.1794 (0.0150)
d_{OPC}	0.1834 (0.0058)	0.1014 (0.0062)	0.0813 (0.0058)	0.1751 (0.0107)	0.1463 (0.0127)	0.1668 (0.0102)
d_{ECOC}	0.1823 (0.0051)	0.1040 (0.0062)	0.0820 (0.0059)	0.1773 (0.0080)	0.1512 (0.0108)	0.1612 (0.0124)
d_{HT}	0.1283 (0.0061)	0.0986 (0.0068)	0.0787 (0.0055)	0.1573 (0.0072)	0.1482 (0.0112)	0.1336 (0.0121)
d_P	0.1260 (0.0054)	0.0998 (0.0065)	0.0780 (0.0052)	0.1618 (0.0086)	0.1470 (0.0103)	0.1332 (0.0135)
d_{MM}	0.1449 (0.0051)	0.1029 (0.0067)	0.0799 (0.0049)	0.1581 (0.0071)	0.1384 (0.0114)	0.1359 (0.0107)
d_{JS}	0.1277 (0.0048)	0.0988 (0.0051)	0.0771 (0.0065)	0.1644 (0.0085)	0.1472 (0.0103)	0.1361 (0.0147)
d_{KW1}	0.1273 (0.0054)	0.1000 (0.0080)	0.0784 (0.0054)	0.1587 (0.0101)	0.1474 (0.0125)	0.1323 (0.0128)
d_{KW2}	0.1411 (0.0050)	0.1025 (0.0057)	0.0815 (0.0050)	0.1566 (0.0071)	0.1351 (0.0119)	0.1288 (0.0113)

The means values of this ratio are given in Table 3. The best aggregating classifier is that for which the positive value of the ratio is maximal. The negative value of the ratio shows that the error for multi-class classifier is less than the error for aggregating classifier. We were also testing the null-hypothesis that the aggregating classifier perform better than multi-class classifier (the relative error is significantly greater than zero). We used the non-parametric Wilcoxon signed rank test.

Table 3. The means of the relative errors from 12 data sets (stars denoted the statistical significant differences at 0.05 level)

Classifier	Classification procedure					
	ldc	qdc	parzendc	naivebc	treec	rnnc
d_{OPC}	–54.137	–1.091	–13.032	–10.869	3.956	11.937*
d_{ECOC}	–55.239	–3.509	–15.368	–15.204	3.296	12.348*
d_{HT}	8.382	1.911	0.366	–5.463	–4.661	21.796*
d_P	12.666*	–0.581	4.348*	–6.909	–3.799	20.584*
d_{MM}	2.965	–2.029	–5.134	–1.369	3.239	24.092*
d_{JS}	9.728*	–2.255	0.847	–6.609	–3.742	16.752*
d_{KW1}	9.677*	0.029	0.977	–5.504	–3.400	19.354*
d_{KW2}	5.011*	–1.208	–11.785	–1.735	5.250	24.617*

From Table 3 we see that in order to receive good results we should combined a classification procedure with suitable aggregated classifier. For example, the OPC and ECOC decomposition methods improve the quality such “unstable” classification procedures as classification trees and random neural net. For remaining procedures using these decomposition method is worthless. The multi-class linear Bayes classifier is less effective than almost all aggregated classifiers. This multi-class procedure assumed that all classes have the same covariance matrix. In the procedures based on decomposition, this assumption is used only for each pair of classes. For quadratic Bayes procedure, Parzen density based procedure and naïve Bayes procedure the decomposition algorithms do not improve the quality of these procedure (except Price aggregated classifier in parzendc procedure).

We are aware that the classification performance of the discussed decomposition methods need to be further evaluated considering additional real-world data and multi-class procedures.

REFERENCES

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

Dillon W.R., Goldstein, M. (1984), *Multivariate Analysis Methods and Applications*, Wiley.

Friedman J.H. (1996), Another approach to polychotomous classification. Technical report, Stanford Univ.

Hastie T., Tibshirani R. (1998), Classification by pairwise coupling, *The Annals of Statistics*, 26, 451–471.

van der Heijden F., Duin R.P.W., de Ridder D., Tax D.M.J. (2004), *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using Matlab*, Wiley.

Jassem W. (1997), Polish phonetical balanced and frequency-weghted word list, *Speech and Language Technology*, 1, 71–99.

Jelonek J., Stefanowski J. (1998), Experiments on solving multiclass learning problems by n^2-classifier, In *Proceedings of the 10th European Conference on Machine Learning (ECML-98)*, Chemnitz, Germany, 172–177.

Johnson R.A., and Wichern D.W. (1982), *Applied Multivariate Statistical Analysis*, Prentice-Hall, Inc.

Krzyśko M., Wołyński W. (2008), New variants of pairwise classification, *European Journal of Operational Research*, doi: 10.1016/j.ejor.2008.11.009.

Merz C.J., Murphy P.M. (1998), UCI repository of machine learning databases. Machine-readable data repository http://www.ics.uci.edu/~mlearn/mlrepository.html, Irvine, CA: University of California, Department of Information and Computer Science.

Mojirsheibani M. (2002), A comparison study of some combined classifiers, *Commun. Statist. - Simula.*, 31, 245–260.

Moreira M., Mayoraz E. (1998), Improved pairwise coupling classification with correcting classifiers, In *Proceedings of the 10th European Conference on Machine Learning (ECML-98)*, Chemnitz, Germany, 160–171.

Price D., Knerr S., Personnaz L., Dreyfus G. (1995), Pairwise neural network classifiers with probabilistic outputs, In *Advances in Neural Information Processing Systems 7 (NIPS-94)*, The MIT Press, 1109–1116.

Waldemar Wołyński

EFEKTYWNOŚĆ ALGORYTMÓW DEKOMPOZYCYJNYCH WIELOKLASOWYCH ZAGADNIEŃ KLASYFIKACYJNYCH

Problem predykcji etykiety klasy (grupy, populacji) na podstawie obserwacji wektora cech jest nazywany klasyfikacją, analizą dyskryminacyjną lub uczeniem się pod nadzorem. Zbiór etykiet składa się z $K>2$ elementów w przypadku zagadnień wieloklasowych oraz z $K=2$ elementów w przypadku zagadnień dwuklasowych (binarnych). Ponieważ zagadnienia dwuklasowe są z reguły o wiele prostsze od zagadnień wieloklasowych (co więcej, niektóre algorytmy klasyfikacyjne dają się zastosować jedynie w przypadku dwuklasowym) wielu autorów proponuje dekompozycje zagadnień wieloklasowych do zagadnień binarnych. Do szczególnie znanych algorytmów tego typu należą: one-per-class (OPC), pairwise coupling (PWC) oraz error-correcting output codes (ECOC).

Dualnym do zagadnienia dekompozycji jest zagadnienie łączenia informacji uzyskanych z klasyfikatorów binarnych. Klasyczne algorytmy bazują na minimalizacji odległości Hamminga, technice głosowania lub sumowania prawdopodobieństw a posteriori.

W pracy porównano różne algorytmy dekompozycyjne oraz zarówno klasyczne jak i nowe metody łączenia informacji z klasyfikatorów binarnych. Wyniki tych porównań wskazują jednoznacznie, że zwłaszcza w przypadku „niestabilnych" procedur klasyfikacyjnych takich jak drzewa klasyfikacyjne czy sieci neuronowe, poza uproszczeniem samych procedur uzyskujemy również znaczną poprawę jakości klasyfikacji. Stąd metody te zaliczyć można do technik wzmacniania klasyfikatorów.