# A General Procedure for Combining Binary Classifiers and Its Performance Analysis

Hai Zhao and Bao-Liang Lu*

Departmart of Computer Science and Engineering,
Shanghai Jiao Tong University, 1954 Hua Shan Road,
Shanghai 200030, China
{zhaohai, blu}@cs.sjtu.edu.cn

**Abstract.** A general procedure for combining binary classifiers for multiclass classification problems with one-against-one decomposition policy is presented in this paper. Two existing schemes, namely the min-max combination and the most-winning combination, may be regarded as its two special cases. We show that the accuracy of the combination procedure will increase and time complexity will decrease as its main parameter increases under a proposed selection algorithm. The experiments verify our main results, and our theoretical analysis gives a valuable criterion for choosing different schemes of combining binary classifiers.

## 1 Introduction

The construction of a solution to a multiclass classification problem by combining the outputs of binary classifiers is one of fundamental issues in pattern recognition research. For example, many popular pattern classification algorithms such as support vector machine (SVM) and AdaBoosting are originally designed for binary classification problems and strongly depend on the technologies of multiclass task decomposition and binary classifier combination. Basically, there are two methods for decomposing multiclass problems. One is one-against-rest policy, and the other is one-against-one policy. The former is computationally more expensive, the latter is more popular in practical application and will be concerned in this paper.

There are three main combination policies for one-against-one scheme according to reported studies. a) the most-winning combination (round robin rule ($R^3$) learning [1]); b) the min-max combination that comes from one of two stages in min-max modular ($M^3$) neural network [2]; and c) decision directed acyclic graph (DDAG) [3]. In comparison with one-against-rest scheme, a shortcoming of one-against-one decomposition procedure is that it will yield too many binary classifier modules, precisely the quantity is $K(K-1)/2$, that is, the quadratic

* To whome correspondence should be addressed. This work was supported in part by the National Natural Science Foundation of China via the grants NSFC 60375022 and NSFC 60473040.

L. Wang, K. Chen, and Y.S. Ong (Eds.): ICNC 2005, LNCS 3610, pp. 303–312, 2005.

function of the number of classes, $K$. In the recognition phase, however, it is observed that only a part of binary classifiers will be called to produce a solution to the original multiclass problem.

In order to improve the response performance of this kind of classifiers, it is necessary and meaningful to develop an efficient algorithms for selecting necessary binary classifiers in the recognition phase. Therefore, we focus on binary classifier selection problem under a novel general combination procedure of binary classifiers proposed in this paper. Here, we will only care the module based time complexity, which means our work will be independent of the classification algorithms and then it earns more generality. On the contrary, a related work in [4] focuses on an optimized combining policy for margin-based classification, which strongly depends on classification methods used in binary classifiers.

One of our previous work [5] gives a comparison between DDAG combination and the min-max combination and proves that DDAG can be seen as a partial version of the min-max combination. With ulterior study in this paper, we may obtain a more comprehensive understanding of combination of binary classifiers.

The rest of the paper is organized as follows: In Sections 2 we briefly introduce the min-max combination and the most-winning combination for binary classifiers. In Section 3, a generalized combination procedure is presented and two equal relations are proved. A selection algorithm is presented for the general combination procedure is presented in Section 4. The experimental results and comments on theoretical and experimental results are presented in Section 5. Conclusions of our work and the current line of research are outlined in Section 6.

## 2 Min-Max and Most-Winning Combinations for Binary Classifiers

Suppose a K-class classification problem is divided with one-against-one task decomposition, then $K(K-1)/2$ individual two-class sub-problems will be produced.

We use $M_{ij}$ to denote a binary classifier that learns from training samples of class $i$ and class $j$, while $0 \leq i, j < K$. The output coding of binary classifier $M_{ij}$ in the min-max combination is defined as 0 and 1, where 1 stands for its output of class $i$ and 0 stands for class $j$. $M_{ij}$ will be reused as $M_{ji}$ in the min-max combination, and they output contrary results for the same sample. Thus, though $K(K-1)$ binary classifiers will be concerned in the min-max combination, only one half of them need to be trained.

Before combination, we sort all $K(K-1)$ binary classifier $M_{ij}$ into $K$ groups according to the same first subscript $i$, which is also regarded as the group label. Combination of outputs of all binary classifiers is performed through two steps. Firstly, the minimization combination rule is applied to all binary classifiers of each group to produce the outputs of $K$ groups. Secondly, the maximization combination rule is applied to all groups outputs. If the result of the maximization procedure is 1, then the label of that group which contribute to such result will be the class label of combining output, otherwise, the combining output is

unknown. We name the group which leads to the class label of combining output as "winning group", and the others as " failure groups".

A min-max combination procedure is illustrated in Fig 1.

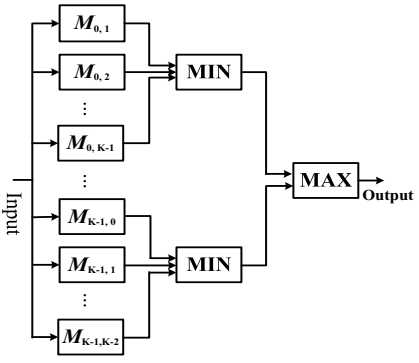



**Fig. 1.** Illustration of K-class min-max combination of $(K-1) \times K$ binary classifiers with $K$ MIN units and one MAX unit

For the most-winning combination of binary classifiers, a direct output coding is applied. The output of each $M_{ij}$ is just $i$ or $j$, instead of 0 or 1. And the combination policy is concise, too. The class label supported by the most binary classifiers is the combining output of $K(K-1)/2$ binary classifiers.

## 3 A General Combining Procedure for Binary Classifiers

For $K(K-1)/2$ binary classifiers produced by one-against-one decomposition procedure, we present a general combination procedure, named N-voting combination, denoted by $V(K,N)$, where $N$ is an additional parameter. A direct class output coding is used in the combination, that is, the output of a binary classifier $M_{ij}$ will just be class $i$ or class $j$. Combination rule is defined as follows. If there are at least $N$ binary classifiers support a class label, e.g. class $i$, and no more binary classifiers support any other class label, then the combining output is just class $i$. Otherwise, the combining output is unknown class.

We will show that N-voting combination $V(K,K-1)$ is equal to the min-max combination. In fact, if there is a class, e.g. class $i$, with consistent support of $K-1$ binary classifiers under $V(K,K-1)$ combination, then this means that only these binary classifiers, $M_{ij}$, $0 \le j \le K-1$, and $i \ne j$, must all support the same class $i$. In other words, their output must all be class 1 under coding method of the min-max combination. These $K-1$ binary classifiers just form a group under the min-max combination. Thus, it must be the group with label $i$ that wins the combination, which means the combining output is class $i$ under the min-max combination. On the contrary, if there is one winning group with a label $i$, under the min-max combination, then these $K-1$ binary classifiers must support the same class $i$. Notice that since the classifier $M_{ij}$ has output

class $i$, then the symmetrical classifier $M_{ji}$ must output the same result class $i$, namely only $K-2$ binary classifiers support class $j$ in the group that are supposed to supported class $j$ as the combining output, which leads to a failure and means that no more binary classifiers support any other class except for class $i$. According to the definition of $V(K, K-1)$ combination, the combining output must be class $i$ under $V(K, K-1)$ combination. So the conclusion that $V(K, K-1)$ and the min-max combination are equal combinations can be drawn. What's more, since the same class label can only be supported by at most $K-1$ binary classifiers, this comes the fact that the upper bound of $N$ must be $K-1$. It is easy to recognize that the supremum of $N$ is $K-1$, too.

We also show that $V(K, [K/2]+1)$ combination is equal to the most-winning combination, where denotation $[K/2]$ means the largest integer below $K/2$. It is induced from the following two facts.

a) For convenient description, we name such combination as $v(K, N)$ combination. If there are just $N$ binary classifiers support a class label, e.g. class $i$, and no more binary classifiers support any other class label, then the combining output is just class $i$. Otherwise, the combining output is unknown class. Suppose the set of combining outputs of all defined class labels by $v(K, N)$ combination is denoted by $s_N$, and the set of combining outputs of all defined class labels by $V(K, N)$ combination is denoted by $S_N$. For the same test sets and trained binary classifiers, there must be $S_N = s_{K-1} \cup s_{K-2} \cup ... \cup s_N$. Then it is obvious that $S_{N_1} \subseteq S_{N_2}$when $N_1 > N_2$, for all $0 \leq N_1, N_2 < K$. That is, for the larger $N$, the corresponding $V(K, N)$ combination will give the less outputs of defined class labels. The reason is that the condition to finish a combining output of defined class label is more and more strict as the value of $N$ increases. Turn to the case of the most-winning combination, such result can be obtain according to its definition:

$$S_{mw} = s_{K-1} \cup s_{K-2} \cup ... \cup s_1, or \quad S_{mw} = S_1. \tag{1}$$

b) To give a combining output of defined class label under $V(K, N)$ or the most-winning combination, such condition must be satisfied: after $N$ binary classifiers are excluded in $K(K-1)/2$ binary classifiers, the remaining classifiers are divided into $K-1$ groups, in which the numbers of binary classifiers all are less than $N$, that is, the following inequality should be satisfied.

$$N > \frac{K(K-1)/2 - N}{K-1}. \tag{2}$$

The solution to the above inequality is $N > (K-1)/2$. Consider $N$ must be an integer, we have $N \geq [(K-1)/2] + 1$, that is, $N \geq [K/2] + 1$. This result suggests

$$s_N = \phi, \forall N, 0 < N < [K/2] + 1. \tag{3}$$

According to (1) and (3), we obtain

$$S_{mw} = s_{K-1} \cup s_{K-2} \cup ... \cup s_{[K/2]+1}, or \\ S_{mw} = S_{[K/2]+1}, \tag{4}$$

and consider all undefined class labels will be output as unknown classes. Therefore, the equality between $V(K, [K/2]+1)$ and the most-winning combination is obvious.

However, the fact that $[K/2]+1$ is a lower bound of $N$ is not necessary to lead to the fact that $[K/2]+1$ is the infimum of $N$ just like the case of upper bound of $N$. Actually, many sets $s_N$ are empty for some $N > [K/2]+1$ in practical classification tasks. To find a larger lower bound of $N$ is still remained as an open problem.

## 4 Selection Algorithm for Combining Binary Classifiers

The original N-voting combination needs $K(K-1)/2$ binary classifiers to be tested for a sample before the mostly supported class label is found. But if we consider the constraint of the value of $N$, then it is possible to reduce the number of binary classifiers for testing, which give an improvement of response performance.

As mentioned in Section 2, $K-1$ binary classifiers with the same first subscript $i$ are regarded as one group with the group label $i$. If there exists more than $K-N$ binary classifiers without supporting the group label in a group for a given value of $N$, then it is meaningless for checking the remained classifiers in the group since this group loses the chance of being a winning one, that is to say, the remained classifiers in the group can be skipped.

The selection algorithm for N-voting combination $V(K, N)$ is described as follows.

1. For a sample, let $i = 0$ and $j = 1$.
2. Set all counters $R[i] = 0$, which stands for the number of binary classifiers rejecting group label $i$, for $0 \leq i < K$.
3. While $i \leq K$, do
   (a) While $j \leq K$ and $R[i] \leq K - N$, do
      i. Check the binary classifier $M_{ij}$.
      ii. If $M_{ij}$ rejects class label $i$, then $R[i] = R[i] + 1$, else $R[j] = R[j] + 1$.
      iii. Let $j = j + 1$, if $j = i$, then let $j = j + 1$ again.
   (b) Let $i = i + 1$ and $j = 1$.
4. Compare each number of binary classifiers rejecting the same class to find the lest-rejected class label as combining output. If all $R[i] > K - N$, for $0 \leq i < K$, then output unknown class as combining classification result.

It is obvious that the chance of a group to be removed by selection algorithm will increase as the value of $N$ increases. This means the efficiency of selection procedure will increase, too. Thus, with the highest value of $N$, $V(K, K-1)$,

or the min-max combination, has the best test performance in the combination series.

Notice that the strictness of voting for a combining output of defined class label will be increase as the value of $N$ increases from $[K/2]+1$ to $K-1$, monotonously. The chance to complete such combination will decrease, simultaneously. This means the accuracy of $V(K,N)$ combination will decrease, monotonously, and the unknown rate will increase, monotonously. Thus, $V(K,[K/2]+1)$ or the most-winning combination is of the highest accuracy in the combination series.

It is hard to directly estimate the performance of N-voting combination selection algorithm. Here we give an experimental estimation. The number of checked binary classifiers under $V(K,K-1)$ or the min-max combination will be

$$n_M = K(\alpha log(K) + \beta), \tag{5}$$

where $\alpha$ and $\beta$ are two constants that depend on features of binary classifier, experimentally, $0 < \alpha \leq 1$ and $-0.5 < \beta < 0.5$. And the number of checked binary classifiers under $V(K,[K/2]+1)$ (or the most-winning policy in some cases) combination will be

$$n_R = \gamma K^2, \tag{6}$$

where $\gamma$ is a constant that depends on features of binary classifier, experimentally, $0 < \gamma < 0.3$. According to above analysis, performance of $V(K,N)$ combination should be between $n_M$ and $n_R$.

According to above performance estimation, our selection algorithm can improve the response performance of one-against-one method from quadratical complexity to logarithmal complexity at the number of binary classifiers in the best case, namely the min-max combination or 1.67 times at least in the worst case, namely the most-winning combination policy.

## 5 Experimental Results

Two data sets shown in Table 1 from UCI Repository[6] are chosen for this study. Two algorithms, k-NN with $k=4$ and SVM with RBF kernel are taken as each binary classifier, respectively. The kernel parameters in SVM training are shown in Table 1, too. The experimental results of N-voting combination with different values of $N$ are shown in Tables 2-5. These tables list the numbers of checked binary classifiers, which show the performance comparison independent of running platform.

It is necessary to access 45 and 325 binary classifiers for two data sets respectively for testing a sample without any module selection. while there is only one half of binary classifiers or less to be checked under presented selection algorithm. This demonstrates an outstanding improvement of test performance. Consider the generality of N-voting combination, the selection algorithm presented has actually included selection procedure of the min-max combination

**Table 1.** Distributions of data sets and corresponding parameters for SVMs

| Data sets | #Class | Number of Samples | | Parameters of SVM | |
|---|---|---|---|---|---|
| | | Train | Test | $\gamma$ | $C$ |
| Optdigits | 10 | 3823 | 1797 | 0.0008 | 8 |
| Letter | 26 | 15000 | 5000 | 0.0125 | 8 |

**Table 2.** Performance of Optdigits data set on N-voting combination: k-NN algorithm

| $N$ | Accuracy | Incorrect rate | Unknown rate | #checked modules |
|---|---|---|---|---|
| 6 | 98.39 | 1.61 | 0.00 | 25.55 |
| 7 | 98.39 | 1.61 | 0.00 | 26.16 |
| 8 | 98.39 | 1.61 | 0.00 | 24.84 |
| 9 | 98.39 | 1.61 | 0.00 | 20.74 |

**Table 3.** Performance of Optdigits data set on N-voting combination: SVM algorithm

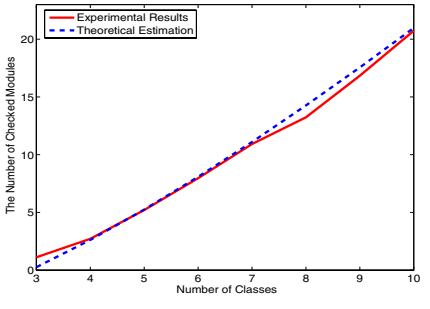
| $N$ | Accuracy | Incorrect rate | Unknown rate | #checked modules |
|---|---|---|---|---|
| 6 | 99.00 | 1.00 | 0.00 | 24.91 |
| 7 | 99.00 | 1.00 | 0.00 | 25.53 |
| 8 | 99.00 | 1.00 | 0.00 | 24.61 |
| 9 | 98.94 | 0.78 | 0.28 | 20.69 |

**Table 4.** Performance of Letter data set on N-voting combination: k-NN algorithm

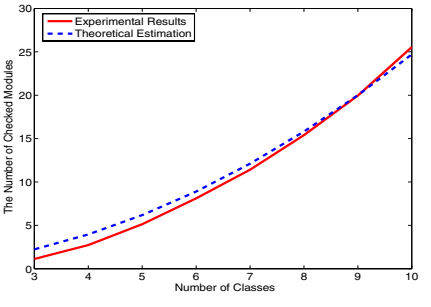
| $N$ | Accuracy | Incorrect rate | Unknown rate | #checked modules |
|---|---|---|---|---|
| 14 | 95.78 | 4.22 | 0.00 | 191.15 |
| 15 | 95.78 | 4.22 | 0.00 | 191.05 |
| 16 | 95.78 | 4.22 | 0.00 | 189.47 |
| 17 | 95.78 | 4.22 | 0.00 | 186.34 |
| 18 | 95.78 | 4.22 | 0.00 | 181.51 |
| 19 | 95.78 | 4.22 | 0.00 | 174.85 |
| 20 | 95.78 | 4.22 | 0.00 | 165.73 |
| 21 | 95.78 | 4.22 | 0.00 | 154.19 |
| 22 | 95.78 | 4.22 | 0.00 | 139.74 |
| 23 | 95.78 | 4.22 | 0.00 | 121.98 |
| 24 | 95.78 | 4.22 | 0.00 | 99.49 |
| 25 | 95.74 | 4.02 | 0.24 | 73.41 |

**Table 5.** Performance of Letter data set on N-voting combination: SVM algorithm

| $N$ | Accuracy | Incorrect rate | Unknown rate | #checked modules |
|---|---|---|---|---|
| 14 | 97.18 | 2.82 | 0.00 | 188.77 |
| 15 | 97.18 | 2.82 | 0.00 | 189.02 |
| 16 | 97.18 | 2.82 | 0.00 | 187.45 |
| 17 | 97.18 | 2.82 | 0.00 | 184.54 |
| 18 | 97.18 | 2.82 | 0.00 | 180.00 |
| 19 | 97.18 | 2.82 | 0.00 | 173.62 |
| 20 | 97.18 | 2.82 | 0.00 | 165.33 |
| 21 | 97.18 | 2.82 | 0.00 | 155.04 |
| 22 | 97.18 | 2.82 | 0.00 | 141.46 |
| 23 | 97.18 | 2.80 | 0.02 | 124.68 |
| 24 | 97.16 | 2.80 | 0.04 | 103.26 |
| 25 | 96.80 | 2.34 | 0.86 | 76.27 |




(a)




(b)

**Fig. 2.** Comparison of theoretical estimation and experimental result of N-voting combination on Optdigits data set under k-NN algorithm, where $\alpha = 1.05, \beta = -0.32$ and $\gamma = 0.247$. (a) $V(K, K-1)$ combination and (b) $V(K, [K/2]+1)$ combination

and the most-winning combination. If we regard selected $V(K, [K/2]+1)$ combination as selected the most-winning combination in the worst case, then there comes nearly 1.7 times improvement at least. If a larger $N$ is taken, then the speeding is much more. In addition, the accuracy and unknown rate do decrease and increase, respectively, while the value of $N$ increases just as expected. However, the decreasing of accuracy or increasing of unknown rate is not outstanding when $N$ is small enough. This suggests that the most-winning combination is equal to $V(K, N)$ combination with a value of $N$ which may be many larger than $[K/2]+1$.

By removing samples of the last class continuously from each data set, we obtain a 3-26 data sets for Letter data and 3-10 data sets for Optdigits data.

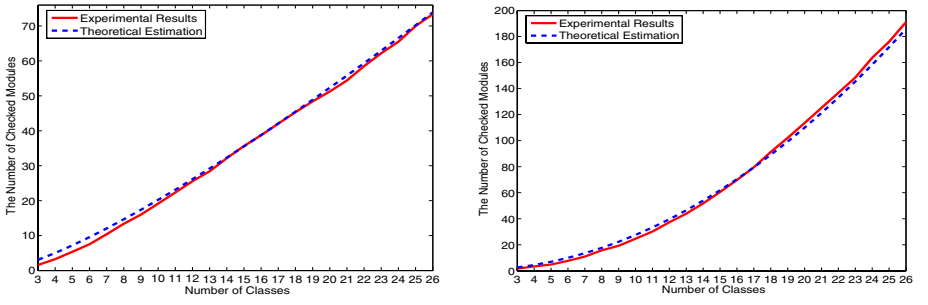



**Fig. 3.** Comparison of theoretical estimation and experimental result of N-voting combination on Letter data set under k-NN algorithm, where $\alpha = 0.87, \beta = 0.0077$ and $\gamma = 0.275$. (a) $V(K, K-1)$ combination and (b) $V(K, [K/2]+1)$ combination

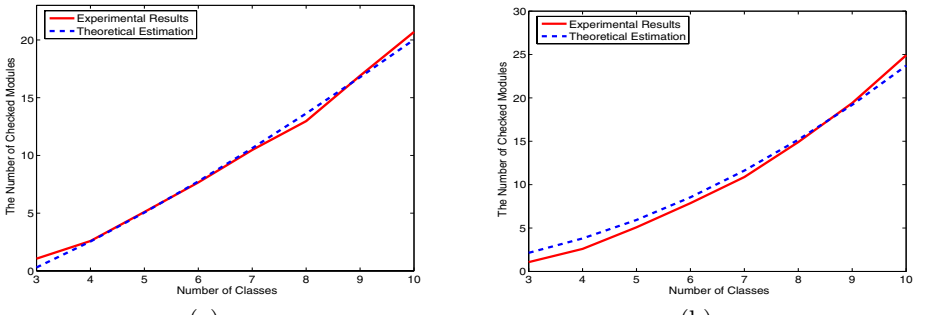



**Fig. 4.** Comparison of theoretical estimation and experimental result of N-voting combination on Optdigits data set under SVM algorithm, where $\alpha = 1, \beta = -0.3$ and $\gamma = 0.237$. (a) $V(K, K-1)$ combination and (b) $V(K, [K/2]+1)$ combination

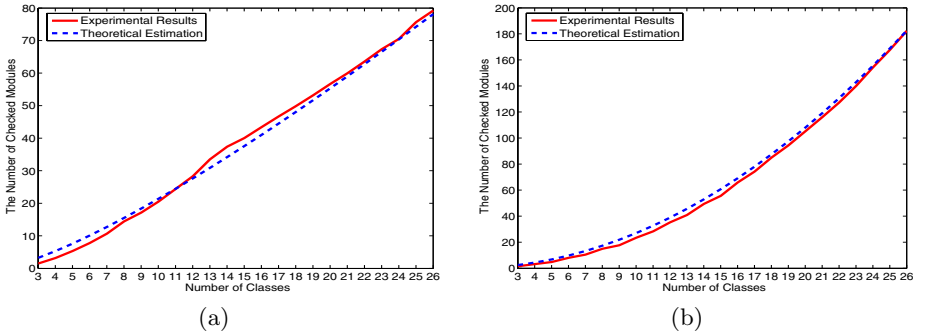



**Fig. 5.** Comparison of theoretical estimation and experimental result of N-voting combination on Letter data set under SVM algorithm, where $\alpha = 0.92, \beta = -0.0385$ and $\gamma = 0.27$. (a) $V(K, K-1)$ combination and (b) $V(K, [K/2]+1)$ combination

Under selection algorithm, the comparison of the numbers of checked binary classifiers between experimental results and theoretical estimation under continuous classes are shown in Figs. 3-4. We see that the experimental estimation value and experimental results are basically identical.

## 6 Conclusions

A general combination procedure of binary classifiers for multi-classification with one-against-one decomposition policy has been presented. Two existing schemes, the min-max combination and the most-winning combination, can be regarded as its two special cases. For such general combination procedure, we ulteriorly propose a selection algorithm. An improvement of response performance to the original combining procedure is demonstrated. The experimental performance estimation of selection algorithm is given, too. The experiments verify the effectiveness of the proposed selection algorithm. Our theoretical analysis gives a valuable criterion for choosing combination policies of binary classifiers. From the generality of our work, the improvement of response performance with presented selection algorithm can also be widely applied, especially for multi-class classification with a large number of classes.

## References

1. Frnkranz, J.: Round Robin Classification. The Journal of Machine Learning Research, Vol. 2 (2002) 721-747
2. Lu, B. L., Ito, M.: Task Decomposition and Module Combination Based on Class Relations: a Modular Neural Network for Pattern Classification. IEEE Transactions on Neural Networks, Vol. 10 (1999) 1244-1256
3. Platt, J., Cristianini, N., Shawe-Taylor, J.: Large Margin DAGS for Multiclass Classification, Advances in Neural Information Processing Systems, 12 ed. S.A. Solla, T.K. Leen and K.-R. Muller, MIT Press (2000)
4. Allwein, E. L., Schapire, R. E., Singer, Y.: Reducing Multiclass to Binary: a Unifying Approach for Margin Classifiers, Journal of Machine Learning Research, Vol. 1 (2000) 113-141
5. Zhao, H., Lu, B. L.: On Efficient Selection of Binary Classifiers for Min-Max Modular Classifier, Accepted by International Joint Conference on Neural Networks 2005-IJCNN2005, Montral, Qubec, Canada, July 31-August 4, (2005)
6. Blake, C. L., Merz, C. J.: UCI Repository of machine learning databases [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science (1998)