11-1-2006

One-Class-at-a-Time Removal Sequence Planning Method for Multiclass Classification Problems

Chieh-Neng Young
National Sun Yat-Sen University

Chen-Wen Yen
National Sun Yat-Sen University

Yi-Hua Pao
ASUS International

Mark L. Nagurka
Marquette University, mark.nagurka@marquette.edu

Accepted version. *IEEE Transactions on Neural Networks*, Vol. 17, No. 6, (November 2006): 1544-1549. DOI.

IEEE Transactions on Neural Networks, Vol. 17, No. 6 (2006): 1544-1549. DOI.

One-Class-at-a-Time Removal Sequence Planning Method for Multiclass Classification Problems

Chieh-neng Young
Department of Mechanical and Electro-Mechanical Engineering, National Sun Yat-Sen University, Kaohsiung 80424, Taiwan, R.O.C.

Chen-wen Yen
Department of Mechanical and Electro-Mechanical Engineering, National Sun Yat-Sen University, Kaohsiung 80424, Taiwan, R.O.C.

Yi-hua Pao
Department of Mechanical and Electro-Mechanical Engineering, National Sun Yat-Sen University, Kaohsiung 80424, Taiwan, R.O.C.

Mark L. Nagurka
Department of Mechanical and Industrial Engineering, Marquette University, Milwaukee, WI

Abstract

Using dynamic programming, this work develops a one-class-at-a-time removal sequence planning method to decompose a multiclass classification problem into a series of two-class problems. Compared with previous decomposition methods, the approach has the following distinct features. First, under the one-class-at-a-time framework, the approach guarantees the optimality of the decomposition. Second, for a Kclass problem, the number of binary classifiers required by the method is only $K-1$. Third, to achieve higher classification accuracy, the approach can easily be adapted to form a committee machine. A drawback of the approach is that its computational burden increases rapidly with the number of classes. To resolve this difficulty, a partial decomposition technique is introduced that reduces the computational cost by generating a suboptimal solution. Experimental results demonstrate that the proposed approach consistently outperforms two conventional decomposition methods

SECTION I. Introduction

A classification problem deals with objects or events to be classified. Such a problem assumes the existence of a known set of K classes

$$\boldsymbol{C} = \{C_1 C_2 \dots C_K\}$$

where C is the set of known classes and the elements C_k of C are called classes. A class can be defined as a pair of variables

$$\text{Pattern} = [\boldsymbol{x}, C_k]$$

where $\boldsymbol{x}$ is the feature vector that characterizes the property of C_k. The goal of classification is to find a decision boundary in the feature space in order to recognize the class $C_k$when a feature vector $\boldsymbol{x}$is present. This mapping can be constructed by a learning-from-example approach where samples with known classes are given. A classifier can then be designed to find the decision boundary in order to infer the class of unknown samples.

A direct approach for multiclass classification problems is to use a single classifier to try to distinguish all classes simultaneously. To adapt neural networks to such problems, an approach is to assign a binary string to each class as the target output. To enhance its performance, these binary strings can be designed with error-correction [1]–[2][3], so that errors by a few of the binary numbers can be recovered. Nevertheless, separating many classes at one time is still a very challenging task since the complexity of the decision boundary often increases with the number of classes. In responding to this difficulty, two decomposition methods have been proposed. The basic idea of these methods is to convert a multiclass problem $(K \geq 3)$ into a number of two-class problems $(K = 2)$. In this work, the two classes associated with a two-class problem are referred to as the true and the false classes, respectively.

The one-against-all (1-a-a) method (e.g., [4]) converts a K-class problem into Ktwo-class problems. In particular, the ith binary classifier used by the 1-a-a method is designed by choosing C_i as the true class, whereas the union of the remaining classes (denoted as $\overline{C_i}$) is the false class. A sample is assigned to $C_i$when the ith binary classifier has the largest true class output. Despite the fact that the 1-a-a method only needs to solve two-class problems one at a time, the training sets of all these two-class problems still come from the union of the same large number of classes associated with the original multiclass problem. As a result, many of the converted two-class problems can still be very difficult to solve.

By trying to distinguish every set of $C_i$from C_j for $i, j = 1 \ldots, K$ and $i > j$, the one-against-one (1-a-1) method (e.g., [5]) splits a Kclass problem into $K(K-1)/2$ two-class problems. In performing a classification, the 1-a-1 method assigns a sample to the class that has won the largest number of true class votes. Compared with the two-class problems of the 1-a-a method, the two-class problems of the 1-a-1 method are often easier to solve since the decision boundary between $C_i$andC_j is expected to be less complex than the decision boundary that separates $C_i$and $\overline{C_i}$ (which containsC_j as well as all the remaining classes). This observation is supported by several experimental results (e.g., [3], [6]). A tradeoff of this improvement is that the number of classifiers increases from $K$to $K(K-1)/2$. Another problem of the 1-a-1 method comes from the ineffective results produced by some of its classifiers. Specifically, a binary classifier trained by samples from $C_i$and C_j can produce unreliable classification results if it is used to determine the membership of C_k samples for $k \neq j$ and $k \neq i$. This "ineffective decision" problem will be addressed again in Section III.

A goal of this work is to develop an alternative decomposition method that requires fewer classifiers than the 1-a-a method and achieves higher classification accuracy than the 1-a-1 method. The paper is organized as follows. The basic idea of the proposed approach is presented in Section II. To reduce the computational cost and improve the classification accuracy, two variations of the approach are introduced in Section III. Section IV presents experimental results that demonstrate the efficiency and accuracy of the methods, and conclusions are given in Section V.

SECTION II. One-Class-at-a-Time Approach

To perform the decomposition, in the proposed approach, a binary classifier is first designed for $C_i$and $\overline{C_i}$ for every C_i. Next, for every $\overline{C_i}$, a binary classifier is developed to classify C_j and $\overline{C_{ij}}$ for every $j \neq i$. Note that $\overline{C_{ij}}$ represents a class obtained by removing $C_i$and C_j from C. This one-class-at-a-time removal procedure is continued until all classes have been classified. This procedure requires only $K-1$ binary classifiers.

The critical issue of this one-class-at-a-time approach is the planning of the removal sequence. It is very likely that a cleverly arranged sequence can simplify the classification problem. To illustrate this possibility, an artificial problem of separating five classes based on two features, as shown in Fig. 1, is considered. To tackle this problem, it is assumed that C_1, C_2, C_3, and C_4 can be removed one-at-a-time from the training set, as illustrated in Fig. 2. For simplicity, this work uses a sequence of 12345→2345→345→45 to represent the removal sequence of Fig. 2. From Fig. 1, it is easy to see that the four two-class problems associated with this removal sequence are all linearly separable and thus easy to solve. In contrast, a nonlinear decision boundary is required in trying to perform an operation of 12345→1245 to separate C_3 and $\overline{C_3}$.

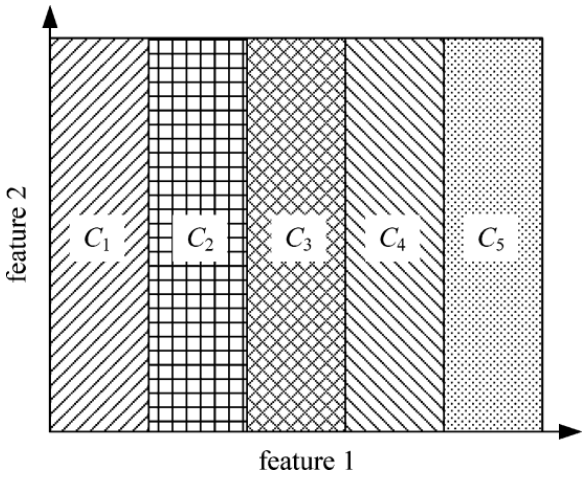

Fig. 1. Artificial five-class problem.

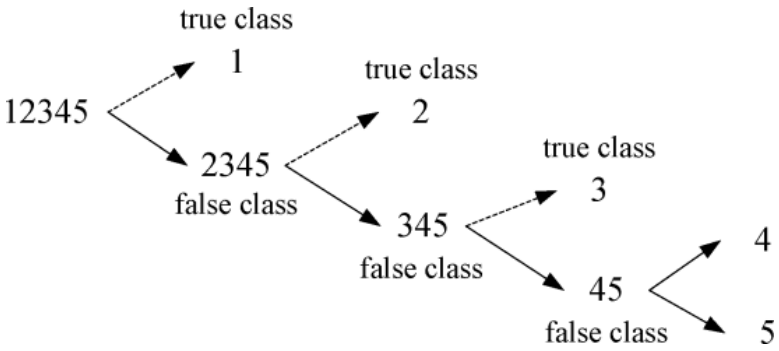

Fig. 2. One-class-at-a-time removal sequence for a five-class problem.

Searching an optimal removal sequence for this problem can be formulated as a multistage decision-making problem (e.g., [7]). In particular, a Kclass problem can be decomposed into a decision-making problem of $K-1$ stages. Fig. 3 depicts such a multistage decision-making problem for a five-class problem. Essentially, each path that connects the starting and ending nodes of Fig. 3 represents a possible one-class-at-a-time solution for the given multiclass problem. In addition, the tree structure (also called version space in machine learning literature) of Fig. 3 contains all such possible solutions. For each stage, the decision that must be made is which class to be removed from the remaining training set so that the cost function can be minimized. Since the goal is to minimize the classification error, this work specifies the cost function as the number of misclassified samples.

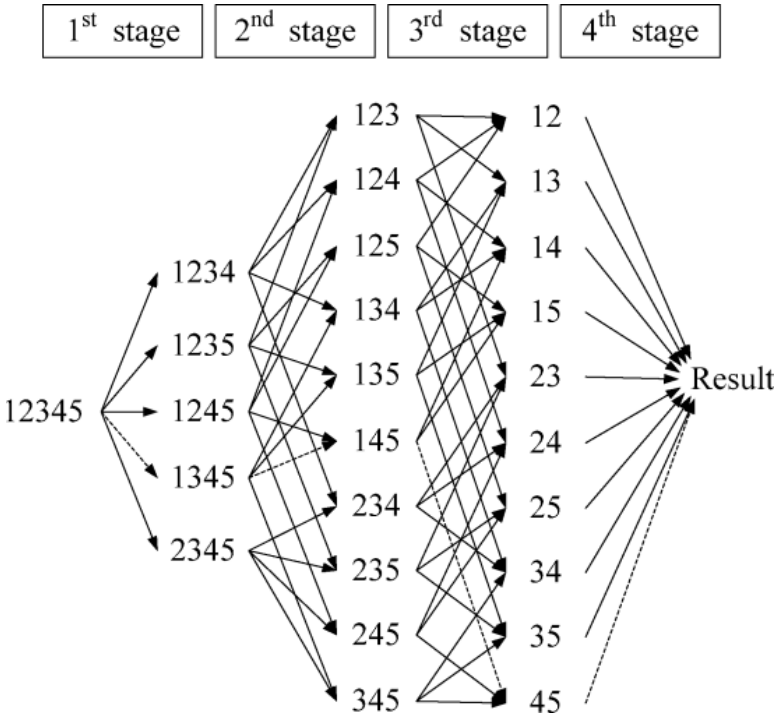

Fig. 3. Tree structure containing all possible one-class-at-a-time solutions for a five-class problem.

Based on the principle of optimality, dynamic programming (DP) can be used to find the global optimal solution for the multistage decision-making problem. For the multiclass classification problem under consideration, the principle of optimality yields the following recurrence relation:

$$J^*(\boldsymbol{C}_p^M) = \min_q [J(\boldsymbol{C}_p^M \rightarrow (\boldsymbol{C}_p^M - \boldsymbol{C}_{pq}^M)) + J^*(\boldsymbol{C}_p^M - \boldsymbol{C}_{pq}^M)](1) \quad (1)$$

for $M = 3,4, \ldots K$. Here, the asterisk indicates that the function has been optimized and thus has the optimal value. In addition, C_p^M represents the pth combination of M classes from the K classes of C, C_{pq}^M is the qth member class of C_p^M, and $J^*(C_p^M)$ and $J^*(C_p^M - C_{pq}^M)$ are the costs associated with the optimal one-class-at-a-time removal sequences for C_p^M and $C_p^M - C_{pq}^M$, respectively. Also, the term $J(C_p^M \rightarrow (C_p^M - C_{pq}^M))$ is the cost associated with the operation of classifying C_p^M into $C_p^M - C_{pq}^M$ and C_{pq}^M. Since the last decision for the proposed one-class-at-a-time approach is to remove a class from a three-class problem, the solution process is initiated by setting M equal to 3. As a result, (1) can be written as

$$J^*(\boldsymbol{C}_p^3) = \min_q [J(\boldsymbol{C}_p^3 \rightarrow (\boldsymbol{C}_p^3 - \boldsymbol{C}_{pq}^3)) + J^*(\boldsymbol{C}_p^3 - \boldsymbol{C}_{pq}^3)]. \quad (2)$$

After removing C_{pq}^3 from C_p^3, $C_p^3 - C_{pq}^3$ contains only two classes. As a result, $J^*(C_p^3 - C_{pq}^3)$ can be determined by separating every possible set of C_i and C_j for $i, j = 1 \ldots, K$ and $i > j$. Note that the 1-a-1 method also solves the same classification problems.

By applying the recurrence relation recursively, the solution procedure is continued until $M = K$. When $M = K$, (1) can be written as

$$J^*(\boldsymbol{C}_p^K) = \min_q [J(\boldsymbol{C}_p^K \rightarrow (\boldsymbol{C}_p^K - \boldsymbol{C}_{pq}^K)) + J^*(\boldsymbol{C}_p^K - \boldsymbol{C}_{pq}^K)]. \quad (3)$$

Note that there is only one possible $\boldsymbol{C}_p^K$, which is the union of all classes, that is, $\boldsymbol{C}$. Therefore, (3) can be rewritten as

$$J^*(\boldsymbol{C}) = \min_q [J(\boldsymbol{C} \rightarrow \overline{\boldsymbol{C}}_q) + J^*(\overline{\boldsymbol{C}}_q)] \quad (4)$$

with $q = 1, \ldots, K$. Note that the binary classifiers associated with $J(\boldsymbol{C} \rightarrow \overline{\boldsymbol{C}_q})$ are identical to those employed by the 1-a-a method.

The results demonstrate that the binary classifiers developed by the conventional 1-a-a and 1-a-1 methods are only a subset of the classifiers used by the proposed approach. For example, for the five-class problem of Fig. 3, the classifiers employed by the 1-a-1 correspond to the rightmost arrows, whereas the leftmost arrows represent the binary classifiers used by the 1-a-a method.

A distinct advantage of DP, compared with other optimization strategies such as a genetic algorithm, is that it guarantees a global optimal solution. Therefore, if the one-class-at-a-time requirement is relaxed, DP can find better solutions. The tradeoff is that the number of possible solutions increases dramatically with the number of classes and the problem then may become computationally too intensive to solve. The proposed approach represents a compromise between computational cost and classification accuracy.

SECTION III. Two Variations of the Proposed Approach

A. A Suboptimal Version of the Proposed Approach

By searching the entire solution space systematically, DP guarantees global optimality for the one-class-at-a-time removal sequence. However, to perform such a search, it can be shown that the number of converted two-class problems is $\sum_{i=0}^{K-3} \binom{K}{K-i} (K-i) + \binom{K}{2}$. This number increases rapidly with the number of classes K hindering the application of the proposed approach to problems with a large number of classes.

To address this "curse of dimensionality" difficulty, this work proposes a suboptimal approach by applying the one-class-at-a-time decomposition only to a subset of C and uses the conventional 1-a-1 method to classify the remaining classes. The specific steps of this suboptimal approach are as follows.

1. Based upon the available computing power, determine the number of classes that is computationally feasible for the proposed approach. Let this number be M.
2. Apply the 1-a-1 method to the given K-class problem.
3. For every M -class subset of the given K classes, determine the number of internal classification errors, which is the number of training samples that are incorrectly assigned to a class that belongs to the same M-class subset.
4. Find the M-class subset that has the largest number of internal classification errors. Denote this set of M classes as $\tilde{C}^M$. The union of the remaining classes is, therefore, $C - \tilde{C}^M$.
5. Apply the proposed one-class-at-a-time approach to decompose $\tilde{C}^M$.
6. To classify a sample, the 1-a-1 method is used first. The classification result is accepted provided that the sample is assigned to one of the classes of $C - \tilde{C}^M$. Otherwise, the sample is classified again by using the one-class-at-a-time removal sequence developed in step 5).

This technique does not guarantee a true optimal solution since it disregards the errors of assigning $\tilde{C}^M$ samples to $C - \tilde{C}^M$ and the errors of misclassifying $C - \tilde{C}^M$ samples to $\tilde{C}^M$. However, by replacing the 1-a-1 method with the proposed decomposition approach for classifying $\tilde{C}^M$ (which has the largest internal classification error among all the M -class subset), it is expected that the number of internal classification errors of $\tilde{C}^M$ can be reduced, thus effectively improving the overall classification accuracy.

B. Building a Committee Machine

Based on the fusion of multiple classifiers, it has been shown that a committee machine can provide higher classification accuracy than an individual classifier (e.g., [8] and [9]). As an example, the 1-a-1 method is essentially a committee machine with $K(K-1)/2$ binary classifier members. Thereafter, the symbol $B(i,j)$ is used to represent such a binary classifier that is trained to classify C_i and C_j.

As described in Section I, one weakness of the 1-a-1 method is the ineffective decision problem, which occurs when trying to use a classifier $B(i,j)$ to classify samples that do not belong to C_i or C_j. The goal of this subsection is to introduce a new committee machine to resolve the ineffective decision problem by using the one-class-at-at-a-time technique.

Let S_{ij} represent the one-class-at-a-time removal sequence that has $B(i,j)$ as its final classifier. The basic idea of this new committee machine is to use the first $K-2$ binary classifiers of S_{ij} to "filter out" samples that do not belong to C_i or C_j. By requiring C_i and C_j to be the last two classes to be processed, the removal sequence S_{ij} can be determined by rewriting the recurrence relation of (1) as

$$J^*(\boldsymbol{C}_p^{M(i,j)}) = \min_q [J(\boldsymbol{C}_p^{M(i,j)} \rightarrow (\boldsymbol{C}_p^{M(i,j)} - \boldsymbol{C}_{pq}^{M(i,j)})) + J^*(\boldsymbol{C}_p^{M(i,j)} - \boldsymbol{C}_{pq}^{M(i,j)})] \quad (5)$$

for $M = 4, \ldots K$. The definition of $C_p^{M(i,j)}$ is similar to $\boldsymbol{C}_p^M$ except that $C_p^{M(i,j)}$ must contain $C_i$and C_j. Similarly, $C_p^{M(i,j)}$ is the qth member class of $C_p^{M(i,j)}$. Note that $C_p^{M(i,j)} \neq C_i$ or C_j since $C_i$and C_j have to be the last two classes to be classified.

To initiate the recurrence relation of (5) for a particular set of i and j, the following results are first set up for $k = 1 \ldots, K, k \neq j$, and $k \neq i$:

$$J^*((\boldsymbol{C}_i \cup \boldsymbol{C}_j \cup \boldsymbol{C}_k)) = J((\boldsymbol{C}_i \cup \boldsymbol{C}_j \cup \boldsymbol{C}_k) \rightarrow (\boldsymbol{C}_i \cup \boldsymbol{C}_j)) + J(\boldsymbol{C}_i \cup \boldsymbol{C}_j). \quad (6)$$

This yields $J^*(C_p^{M(i,j)} - C_{pq}^{M(i,j)})$ for $M = 4$. Relation (5) can then be applied recursively until $M = K$.

The proposed committee machine uses the same set of classifiers $B(i,j)$ as the 1-a-1 method. However, instead of using $B(i,j)$ directly, the committee machine uses S_{ij} to determine the "effectiveness" of $B(i,j)$'s decision. In particular, the first $K-2$ classifiers of S_{ij} can be viewed as a $\overline{C_{ij}}$ sample filter, which can prevent $B(i,j)$ from processing $\overline{C_{ij}}$ samples. Therefore, in this committee machine, the classifier $B(i,j)$ attends voting only when the sample is not assigned to other classes by S_{ij} before reaching classifier $B(i,j)$. Apparently, the reliability of this voting depends on the efficacy of S_{ij} in filtering $\overline{C_{ij}}$ samples. To evaluate its performance, S_{ij} is used to classify the entire training set and count the samples that actually enter the final classification stage of S_{ij}. Among these samples, the ratio of $\overline{C_{ij}}$ samples is computed and denoted as *wij*. Since this ratio characterizes the likelihood of an "ineffective decision", the weighting coefficient for $B(i,j)$'s voting is chosen as 1−*wij*. Finally, the membership of a sample is determined by counting the weighted votes from all the binary classifiers of $B(i,j)$.

In performing a classification, the basic version of the proposed approach uses $K-1$ binary classifiers. In contrast, with $K(K-1)/2$ members, the committee machine version of the proposed approach uses $K(K-1)^2/2$ binary classifiers. As a result, the computational requirement will increase proportionally. This factor should be taken into consideration when implementing the proposed committee machine approach for real-time classification tasks.

SECTION IV. Experimental Results

The first part of this section compares the approach developed in Section II with three conventional methods by using them to solve ten real-world problems obtained from the University of California at Irvine, Irvine, repository of machine learning databases and domain theories [10]. The contents of these ten data sets are summarized in Table I.

Table I Summary of the Tested Data Sets

Dataset	Number of Samples	Number of Classes	Number of Features
Iris	150	3	4
Balance-scale	625	3	4
Lymphography	148	4	18
Hypothyroid	3372	4	28

Vehicle	846	4	18
Car-evaluation	1728	4	6
Satimage	6435	6	36
Glass	214	6	9
Segmentation	2310	7	18
Yeast	1484	10	8
Pendigits	10992	10	16
Vowel	990	11	10
Krk	28056	18	6

Table II Classification Accuracy and Computational Cost of the Single Classifier

Dataset	Classification Accuracy	Computing Time (sec)
Iris	96.67±4.43	2.2
Balance-scale	90.64±3.47	13.5
Lymphography	82.50±8.94	3.5
Hypothyroid	96.50±1.13	188.8
Vehicle	80.52±4.10	40.5
Car-evaluation	92.30±1.77	54.9
Satimage	86.21±1.29	232.2
Glass	64.33±10.05	6.4
Segmentation	93.13±1.73	117.0
Yeast	56.22±3.57	65.2
Pendigits	94.66±0.70	513.9
Vowel	73.18±3.97	128.7
Krk	39.61±0.93	2462.8

In testing these methods, the multilayered perceptron (MLP) is chosen as the base classifier [11]–[12][13]. The numbers of hidden layers and units are chosen as one and five, respectively. No effort has been made to optimize the structure of the MLP. The initial weights are generated randomly from a uniform distribution between 0–0.1. The MLP error measure is chosen as the conventional mean square error with the error defined as the difference between the desired and actual outputs. In this study, the MLP is trained by adaptive boosting (AdaBoost) [14]. The reason for using AdaBoost is that many empirical studies have shown that it can significantly improve the performance of the neural classifiers and is relatively insensitive to overfitting (e.g., [15]).

In comparing the tested methods, the data set is divided into training, validation and testing subsets with an 8:1:1 ratio. The training subset is used to adjust the connection weights of the MLP. The validation subset is used by the early-stop technique to avoid overfitting. The testing subset is used to characterize the generalization accuracy of the MLP. For the sake of reliability, the training process is repeated 100 times by using randomly partitioned training, validation, and testing subsets. This paper reports the average of the testing subset classification accuracy.

The experimental studies were performed using an AMD XP 1700+ based PC. To set up the basis for comparisons, the problems were first solved using a single MLP. Table II presents the means of the classification accuracy and the computing times averaged from 100 trials for each data set. Here, the classification accuracy is defined as the percentage of the correctly classified testing samples. Table III summarizes the means and the standard deviations of the classification accuracy associated with the three decomposition methods. By

comparing these results with the results of the single classifier of Table II, it can be seen that the 1-a-a method achieves higher accuracy in six of the ten tested problems and the 1-a-1 method gives a smaller classification error in seven of the ten problems. In contrast, the proposed approach outperforms the single classifier method in all of the tested problems. In addition, the proposed approach has the smallest classification error in all but the last tested problem.

Table III Summary of Classification Accuracy for the Tested Classification Problem ($K = 10$)

Data set	**Tested Methods**		
	1-a-a	1-a-1	1-at-a-time
Iris	95.81±5.23	95.53±5.59	96.67±4.43
Balance-scale	94.42±3.06	95.21±2.98	95.89±2.76
Lymphography	83.18±8.71	82.54±8.82	83.21±9.05
Hypothyroid	94.93±1.01	96.13±1.39	96.88±0.81
Vehicle	81.02±3.86	81.93±3.71	82.43±4.14
Car-evaluation	97.82±1.11	97.62±1.36	98.05±1.06
Satimage	89.85±1.27	90.81±1.11	90.95±1.17
Glass	45.64±12.29	61.33±10.04	66.00±9.41
Segmentation	96.50±1.13	96.39±1.33	97.01±1.14
Yeast	50.86±1.42	57.41±2.95	56.38±5.78

To compare the computational cost, Table IV summarizes the computing time ratio of the three decomposition methods. Here, the computing time ratio is defined as the ratio of computing time of the tested decomposition method to the computing time of the single MLP. As expected, among the four tested methods, the proposed approach is computationally least efficient. In addition, as shown in Table IV, the computational cost of the proposed approach increases rapidly with the dimension of the classification problem. For example, for the ten-class yeast problem, the computing time of the proposed approach is three orders of magnitude larger than that of the single classifier. In contrast, computationally, the 1-a-a and 1-a-1 methods are only about four times slower than the single classifier.

Table IV Summary of Computing Time Ratio with Respect to Single Classifier Approach ($K = 10$)

Data set	**Tested Methods**		
	1-a-a	1-a-1	1-at-a-time
Iris	1.5	2.9	4.3
Balance-scale	5.2	3.6	8.7
Lymphography	3.1	6.6	16.2
Hypothyroid	2.1	2.6	10.7
Vehicle	2.3	3.6	14.3
Car-evaluation	2.9	7.3	24.6
Satimage	6.0	21.0	197.0
Glass	2.1	2.7	50.1
Segmentation	3.0	7.3	210.5
Yeast	3.7	4.0	1180.9

In responding to this difficulty, the partial decomposition technique is tested on the last four problems of Table I for which the number of classes is ten or greater. In applying the technique, the number of classes to be decomposed M is chosen as 3, 4, 5, and 6. In addition, since the basic version of the proposed approach is less accurate than the 1-a-1 method in dealing with the ten-class yeast problem, the proposed committee machine

method is employed in this part of the experiments. The resulting classification accuracies and computing time ratios are summarized in Tables V and VI, respectively.

As shown in Table V, even with partial decomposition, the proposed suboptimal one-class-at-a-time approach outperforms the 1-a-1 method in all four problems. As M increases, the classification accuracy improves. Also, as shown in Table VI, the computational cost also increases with M. However, with the partial decomposition, this cost has been reduced significantly. For example, for the ten-class yeast problem, the partial decomposition reduces the computing time ratio from 1180.9 to 36.15 (for M=6 or better), demonstrating the effectiveness of the partial decomposition.

SECTION V. Conclusion

Table V Summary of Classification Accuracy for the Tested Classification Problem ($K = 10$)

Dataset			**Method**		
	1-a-1	**One-class-at-a-time**		**Committee Machine**	
		M=3	**M=4**	**M=5**	**M=6**
Yeast	58.09±3.85	58.10±3.82	58.24±3.86	58.42±3.69	58.58±3.72
Pendigit	98.55±0.39	98.57±0.38	98.78±0.34	98.95±0.30	99.07±0.29
Vowel	84.74±4.57	85.08±4.60	87.19±4.17	88.59±3.74	90.23±3.43
Krk	50.50±0.77	50.36±0.76	50.70±0.75	50.84±0.79	51.06±0.80

Table VI Summary of Computing Time Ratio with Respect to Single Classifier Approach ($K = 10$)

Dataset			**Method**		
	1-a-1	**One-class-at-a-time**		**Committee Machine**	
		M=3	**M=4**	**M=5**	**M=6**
Yeast	2.67	3.43	7.45	16.60	36.15
Pendigit	2.44	3.67	6.19	14.73	37.91
Vowel	3.30	3.65	6.4	12.95	38.14
Krk	2.88	3.20	4.32	6.98	13.56

This paper proposes a one-class-at-a-time method to decompose a multiclass problem into a number of two-class problems. In particular, the basic version of the proposed approach splits a Kclass problem into $K - 1$ two-class problems. The planning of the one-class-at-a-time removal sequence is formulated as a multistage decision-making problem, which is then solved using dynamic programming.

To reduce the computational cost of the proposed approach, which increases rapidly with the number of classes, a partial decomposition technique is introduced to determine the suboptimal solution. By using the one-class-at-a-time removal sequence to alleviate the ineffective decision problem, this paper also develops a committee machine framework to improve the classification accuracy.

Experimental results show that the proposed approach consistently provides higher classification accuracy than the conventional single classifier 1-a-a and 1-a-1 methods when the number of classes is less than ten. However, for a tested ten-class problem, the computational cost of the proposed approach is three orders of magnitude larger than that of the single classifier method. By applying the partial decomposition technique, this computational cost can be reduced by a factor of about 30 or better. In addition, with the assistance of the proposed committee machine framework, the suboptimal solution produced by the partial decomposition technique provides better classification accuracy than conventional methods.

References

1. T. G. Dietterich, G. Bakiri, "Solving multiclass learning problems via error-correcting output codes", *J. Artif. Intell. Res.*, vol. 2, pp. 263-286, 1995.
2. V. Guruswami, A. Sahal, "Multiclass learning boosting and error-correcting codes", *Proc. 12th Annu. Conf. Comput. Learning Theory*, pp. 145-155, 1999.
3. J. A. Nossek, R. Eigenmann, G. Papoutsis, W. Utschick, "Classification systems based on neural networks", *Proc. 5th IEEE Int. Workshop Cellular Neural Netw. and Their Applications*, pp. 26-33, 1998.
4. R. Anand, K. Mehrotra, C. K. Mohan, S. Ranka, "Efficient classification for multiclass problems using modular neural networks", *IEEE Trans. Neural Netw.*, vol. 6, no. 1, pp. 117-124, Jan. 1995.
5. S. Knerr, L. Personnaz, G. Dreyfus, "Single-layer learning revisited: A stepwise procedure for building and training a neural network" in Neurocomputing: Algorithms Architectures and Applications, New York:Springer-Verlag, vol. F68, pp. 41-50, 1990.
6. C.-W. Hsu, C.-J. Lin, "A comparison of methods for multiclass support vector machines", *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415-425, Mar. 2002.
7. D. E. Kirk, Optimal Control Theory, NJ, Englewood Cliffs:Prentice-Hall, 1970.

A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, A. Gelzinis, "Soft combination of neural classifiers: A comparative study", *Pattern Recognit. Lett.*, vol. 20, pp. 429-444, Apr. 1999.

8. L. Kuncheva, J. Bezdek, R. Duin, "Decision templates for multiple classifier fusion: an experimental comparison", *Pattern Recognit.*, vol. 34, pp. 299-314, 1999.
9. C. L. Blake, C. J. Merz, *UCI Repository of Machine Learning Databases*, 1998, [online] Available: http://www.ics.uci.edu/~mlearn/ML-Reppository.html.

A. Bortoletti, C. Di Fiore, S. Fanelli, P. Zellini, "A new class of quasi-Newtonian methods for optimal learning in MLP-networks", *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 263-273, Mar. 2003.

B. Xiang, S. Q. Ding, T. H. Lee, "Geometrical interpretation and architecture selection of MLP", *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 84-96, Jan. 2005.

C. Erdogmus, O. Fontenla-Romero, J. C. Principe, A. Alonso-Betanzos, E. Castillo, "Linear-least-squares initialization of multilayer perceptrons through backpropagation of the desired response", *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 325-337, Mar. 2005.

10. Y. Freund, R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", *J. Comp. Syst. Sci.*, vol. 55, no. 1, pp. 119-139, 1997.
11. H. Schwenk, Y. Bengio, "Boosting neural networks", *Neural Comput.*, vol. 12, no. 8, pp. 1869-1887, 2000.

Keywords

IEEE Keywords

Dynamic programming , Computational efficiency , Pattern recognition , Neural networks , Councils , Industrial engineering , Design methodology , Voting

INSPEC: Controlled Indexing

dynamic programming , learning (artificial intelligence) , neural nets , planning (artificial intelligence)

INSPEC: Non-Controlled Indexing

one-class-at-a-time removal sequence planning , multiclass classification problems , dynamic programming , committee machine , partial decomposition technique

Author Keywords

Dynamic programming , multiclass classification , pattern recognition

MeSH Terms

Algorithms , Artificial Intelligence , Cluster Analysis , Information Storage and Retrieval , Neural Networks Computer) , Pattern Recognition, Automated , Signal Processing, Computer-Assisted