Improvement on Response Performance of Min-Max Modular Classifier by Symmetric Module Selection[*]

Hai Zhao and Baoliang Lu

Department of Computer Science and Engineering, Shanghai Jiao Tong University
1954 Huashan Rd., Shanghai 200030, China
{zhaohai,blu}@cs.sjtu.edu.cn

Abstract. This paper aims to improve the response performance of min-max modular classifier by a module selection policy for two-class classification during recognition. We propose an efficient base classifier selection algorithm. We show that the quadratic complexity of original min-max modular classifier can fall onto the level of linear complexity in the number of base-classifier modules for each input sample under presented selection scheme. The experimental results indicate the effectiveness of our algorithm and verify our theoretical analysis.

1   Introduction

Two-class classification is one kind of basic classification problem. Many essential classification schemes often start from binary classifier and then adapt to multi-class classifiers. Therefore, an improvement on two-class problem processing will earn important value to pattern classification.

In this paper, we consider a decomposition policy of two-class classification task, which is one of two processing stages in min-max modular (M^3) neural network[1]-[2]. The policy is based on partition of training set, which make it possible that a large-scale two-class problem is divided into some smaller scale two-class subproblems for parallel processing.

However, one hand, it is obvious that such partition will yield much many modules, which cause serious performance problems. On the other hand, only partial sub-classifier modules will actually function for each input sample in classification. This activates us that it is possible to optimize the original combination procedure of classifiers only to check those necessary modules, instead of all produced sub-classifier modules for an input sample. This is to say, an additional module selection procedure can be introduced for improving the response performance of the classifier.

In this paper, we will present an efficient module selecting policy named symmetrical selection algorithm to improve the response performance of min-max modular classifier.

2   Task Decomposition and Module Combination

Consider a two-class classification problem, whose output coding of class labels are denoted by C_0 and C_1(as two values, $C_1 > C_0$), or equally, 0 and 1, which will be concise and not lose any generality for our algorithm description. Suppose the training set of class C_0 is decomposed into n subsets and the training set of class C_1 is divided into m subsets. By arranging those m and n training subsets in pairs, we obtain $m \times n$ training set pairs. Each pair is learned by a single binary sub-classifier. Therefore, a larger scale two-class problem can be decomposed into $m \times n$ smaller subproblems. We call the binary sub-classifier as base classifier.

Suppose all produced training set pairs are denoted by X_{ij}, for $i = 0, 1, ...,$ $m - 1$, and $j = 0, 1, ..., n - 1$. Without misunderstanding, we also express X_{ij} as the output of the corresponding base classifier.

Min-max combination defines how the outputs of those $m \times n$ base classifiers are combined into the solution to the original problem. Before combination, a grouping operation on $m \times n$ base classifiers should be done: these base classifiers, X_{ij}, for $j = 0, ..., n - 1$, are defined as one "C_1 group" and i is defined as its group label, and those base classifiers, X_{ij}, for $i = 0, ..., m - 1$, are defined as one "C_0 group" and j is defined as its group label.

Min-max combination of all base classifiers includes two stages: Firstly, the minimization rule is applied to each C_1 group to make the output of the group. Secondly, the outputs of all groups are integrated by the maximization rule to make the final output of the original two-class classification problem.

An outstanding merit for min-max modular classifier is that imbalanced classification can always be avoid from a simple decomposition if needed[4].

3   Symmetrical Module Selection Algorithm

We call a C_1 group whose all member base classifiers hold the class C_1 as a "winning group of class C_1" and a class C_0 group whose all member base classifiers hold the class C_0 as a "winning group of class C_0" . It is easy to demonstrate that in the original min-max combination procedure for one input sample, the fact that there is a winning group of C_1 in all $m \times n$ base classifiers will be sufficient and necessary to make the conclusion that the final combination result must be class C_1 according to the combining rules, vice versa, if any winning group of C_1 does not exist, then the final combination result must be class C_0. Thus, a min-max combination of base classifiers will be redescribed as such a search procedure to find a winning group of class C_1.

Similarly, we can also define one symmetrical *max-min* combination to perform a search procedure for a winning group of class C_0. Naturally, that there exists a winning group of class C_0 in all $m \times n$ base classifiers is a sufficient and necessary criterion for a combination result of class C_0. Symmetrical module selection algorithm, just like its name, is a procedure with consideration of the winning group of both class C_0 and class C_1, instead of one-sided search for the winning group of class C_1 in the original model[1],[2].

Basically, symmetrical module selection is based on the following idea: a) Notice that it is impossible that there exists one winning group of class C_1 and one winning group of class C_0 at the same time. The reason is that any winning group of class C_1, X_{ij}, for $j = 0, ..., n-1$, and any winning group of class C_0, X_{ij}, for $i = 0, ..., m-1$, must share one same base classifier, which can not output two different classification results at the same time. b) The symmetrical module selection algorithm is a procedure to continuously exclude those hopeless class C_1 or C_0 group. The existence of a winning group of class C_0 or class C_1 means final combination classification result is class C_0 or class C_1, respectively. Regarding all outputs of base classifiers as a binary matrix, a row can stand for a class C_1 group, and a column can stand for a class C_0 group. Thus, that a base classifier outputs 1 as classification result means that the column it locates loses the chance to be a winning group of class C_0. Symmetrically, that a base classifier outputs 0 as classification result means that the column it locates loses the chance to be a winning group of class C_1.

The symmetrical module selection algorithm is described below in detail. For convenience, we also say the first subscript of a base classifier is its row, and the second is its column, also a row means a class C_1 group and a column means a class C_0 group.

1. For $i = 0, ..., m-1$, set the tag $RF[i] = 0$, and for $j = 0, ..., n-1$, set the tag $CF[j] = 0$, which means all class one and zero groups are not excluded, respectively.
2. Set two counters, $RCounter = m$ and $CCounter = n$, respectively, which means the number of surviving class one and zero groups without excluded.
3. Set starting row and column, $i = 0$ and $j = 0$.
4. Repeat the following operations:
   (a) If i is an excluded group label, that is, $RF[i] = 1$, then i increases continuously until i is not an excluded label any more or $i = m$.
   (b) If j is an excluded group label, that is, $CF[j] = 1$, then j increases continuously until j is not an excluded label any more or $j = n$.
   (c) If the counter, $RCounter = 0$, or $i = m$, then output the combination classification result class 0 and the algorithm ends here.
   (d) If the counter, $CCounter = 0$, or $j = n$, then output the combination classification result class 1 and the algorithm ends here.
   (e) Check the base classifier X_{ij}.
   (f) If X_{ij} holds class C_1, then
      i. Set the tag, $CF[j] = 1$.
      ii. Subtract 1 from the counter, $CCounter = CCounter - 1$
      iii. $j = j + 1$.
   (g) If X_{ij} holds class C_0, then
      i. Set the tag, $RF[i] = 1$.
      ii. Subtract 1 from the counter, $RCounter = RCounter - 1$
      iii. $i = i + 1$.

4   Performance Estimation

We now show that the number of checked base classifiers in symmetrical module selection algorithm for each input sample will be never larger than $m + n - 1$.

In fact, the algorithm is equally a search procedure in a binary matrix with m rows and n columns, in which the start point is top left corner and the end point is bottom right corner. Each checking for a base classifier is equal to an access for one element in the matrix, which means one row or one column in the matrix must be excluded. In turn, the accessing of the algorithm gets to the next row or column without any backtracking. In summary, such search from top left corner to bottom right corner in this matrix mostly covers $m + n - 1$ elements. Namely, the number of checked base classifier in symmetrical module selection for each input sample will not be larger than $m + n - 1$, which is much better than the case, $m \times n$, under the original model.

5   Experimental Results

Three data sets shown in Table 1 from UCI Repository[3] have been chosen for this study. For a typical realization of min-max modular classifier, support vector machine(SVM) with RBF kernel is selected as base classifier[4]. Two parameters, C and γ, are set to 8 and 0.25 for Internet Ads and Adult data sets, respectively. These two parameters are set to 316.2 and 1 for Banana data set, respectively. The same numbers of samples for class C_1 and C_0 in every subsets are taken.

Table 1. Distributions of samples in three data sets

Data Set	Number of Training Samples			Number of Test Samples		
	Total	Class C_1	Class C_0	Total	Class C_1	Class C_0
Internet Ads	2100	1800	300	1179	1020	159
Adult	32561	24720	7841	16281	12435	3846
Banana	40000	21847	18153	490000	270553	219447

Experimental results are shown in Tables 2 through 4. The comparison between experimental results and theoretical bounds in three data sets are shown in Fig 1. Fig 1(a) shows 20 different experimental results, where n varies from 2 to 21 and m correspondingly varies from 12 to 129 for Internet Ads data set. Fig 1(b) shows 100 different experimental results, where n varies from 2 to 101 and m correspondingly varies from 6 to 317 for Adult data set. Note that Table 2 and Table 3 only show parts of experimental results shown in Fig.1(a) and Fig.1(b) for space limitation.

As expected, the experimental results show that our symmetrical module selection algorithm gives an outstanding performance improvement for testing procedure with respect to the case without any selection, while combining accuracies keep the same. The number of actual checked base classifier is also strictly

Table 2. Experimental results on Internet Ads data set

#Modules	m	n	Without Selection			Symmetrical Selection			
			#Modules	Time(ms)	acc.(%)	#Modules	Bound	Time(ms)	acc.(%)
14	129	21	2709	63656	90.16	29.04	149	26718	90.16
18	100	17	1700	44046	89.40	19.78	116	15484	89.40
23	78	13	1014	29890	89.31	15.07	90	8265	89.31
33	56	9	504	17859	89.57	10.62	64	3375	89.57
60	30	5	150	8421	89.90	5.97	34	1000	89.90

Table 3. Experimental results on Adult data set

#Modules	m	n	Without Selection			Symmetrical Selection			
			#Modules	Time(ms)	acc.(%)	#Modules	Bound	Time(ms)	acc.(%)
320	77	25	1925	7027	76.37	25.45	101	116	76.37
157	157	50	7850	15085	76.36	96.79	206	281	76.36
105	235	75	17625	22613	76.39	183.19	309	430	76.39
77	321	102	32742	37679	76.37	320.97	422	666	76.37

Table 4. Experimental results on Banana data set

#Modules	m	n	Without Selection			Symmetrical Selection			
			#Modules	Time(ms)	acc.(%)	#Modules	Bound	Time(ms)	acc.(%)
90	243	202	49086	154023	90.81	229.56	444	1389	90.81
103	212	176	37312	130945	90.73	199.20	387	1279	90.73
121	181	150	27150	107417	90.99	169.43	330	1096	90.99
145	151	125	18875	88564	90.68	141.66	275	883	90.68
181	121	100	12100	66653	90.61	112.55	220	754	90.61
242	90	75	6750	52709	90.80	84.01	164	679	90.80
363	60	50	3000	32309	90.75	55.87	109	645	90.75

under the theoretical bound. However, the ratio of numbers of checked modules and the ratio of practical response time between two cases are not exactly the same, which may mostly attribute to different number of support vectors obtained by each base classifier.

6 Conclusions

An efficient module selection policy has been presented for improving the response performance of min-max modular classifier. We show that the quadratic

(a)

(b)

(c)
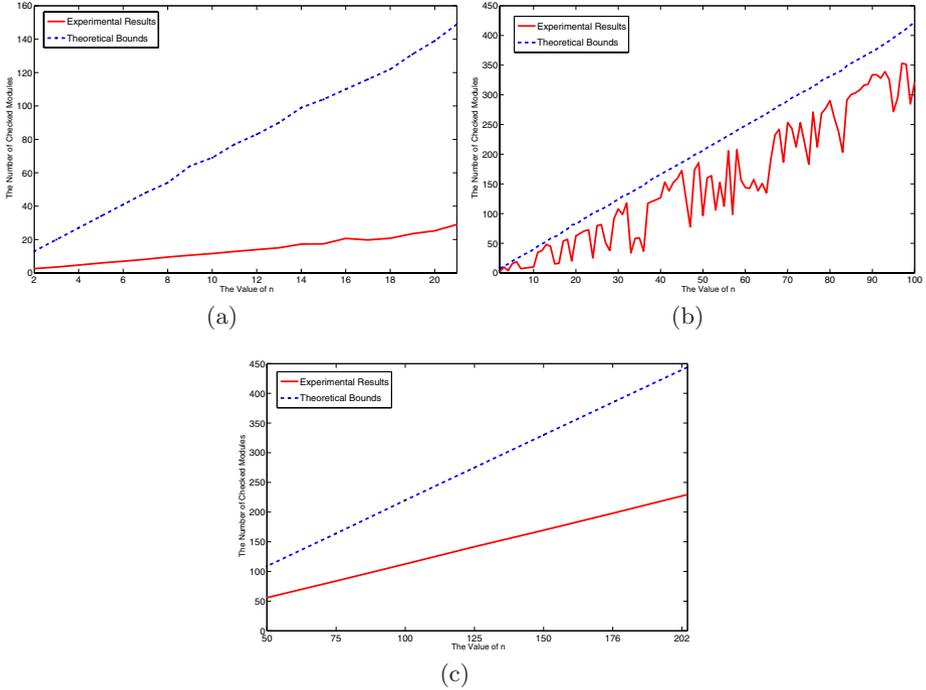
Fig. 1. Comparison of theoretical bound and experimental result on Internet Ads data set(a), Adult data set(b), and Banana data set(c)

complexity of the original min-max modular classifier can be reduced onto linear complexity of the number of base-classifiers for each input sample. The experimental results indicate that an outstanding improvement on the response performance for such modular classifier is obtained.

References

1. Lu, B.L., Ito, M.: Task Decomposition Based on Class Relations: a Modular Neural Network Architecture for Pattern Classification. Lecture Notes in Computer Science. Vol. 1240. Springer (1997) 330-339
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, **10** (1999) 1244-1256
3. 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)
4. Lu, B.L., Wang, K. A., Utiyama, M., Isahara, H.: A Part-versus-part Method for Massively Parallel Training of Support Vector Machines. Proc. of IEEE/INNS Int. Joint Conf. on Neural Networks (IJCNN2004), Budabest, Hungary (2004) 25-29