Equal Clustering Makes Min-Max Modular Support Vector Machine More Efficient

Yimin Wen
Guilin University of Electronic Technology
27 PUBLICATIONS **72** CITATIONS
SEE PROFILE

Bao-Liang Lu
Shanghai Jiao Tong University
259 PUBLICATIONS **3,256** CITATIONS
SEE PROFILE

Hai Zhao
Northeastern University (Shenyang, China)
270 PUBLICATIONS **2,126** CITATIONS
SEE PROFILE

Equal Clustering Makes Min-Max Modular Support Vector Machine More Efficient

Yi-Min Wen[1,2], Bao-Liang Lu[1], and Hai Zhao[1]
[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
1954 Hua Shan Rd., Shanghai 200030, China
{ymwen,blu,zhaohai}@cs.sjtu.edu.cn
[2] Hunan Industry Polytechnic, Changsha 410007, China

Abstract— **Support vector machines has training time at least quadratic in the number of examples, so it is hopeless to use it to solve large-scale problems. In this paper, a novel clustering algorithm that can equally partition training data sets is proposed to improve the performance of the min-max modular support vector machine (M^3-SVM). The simulation results indicate that the proposed clustering method can not only promote the generalization accuracy of the M^3-SVM, but also speed up training and reduce the number of support vectors in comparison with the existing random task decomposition method.**

I. Introduction

Today there are many very large-scale data sets like public-health data, gene expression data, national economics data, and geographic information data. By using these very large data sets, researchers can get higher generalization accuracy, discover infrequent special cases, and avoid over-fitting. However, most of existing machine learning methods such as support vector machine are hard to be used to deal with these very large data sets because both their learning time and space complexity have at least quadratic in the number of examples. Therefore, one of the most challenging problems in machine learning community is to develop new learning model to efficiently handle these large data sets. Many efforts are made to scale SVMs, such as using clustering methods to preprocess large training data [1] [2], using cascade and parallel methods to filter non-support vectors [3] [4] [5].

According to Provost [6], parallelism is a good strategy for dealing with large-scale data sets. In the training phase of parallel learning methods, a large training data set is first decomposed and parallelly processed, then many local learners are obtained. In recognition phase, an unknown sample is presented to all the learners, the outputs of all the learners are integrated to make a final solution to the original problem [7] [8]. Obviously, the performance of parallel method heavily depends on the integration principle of learners and the decomposition strategy of training data set. In our previous work, a modular support vector machine, called min-max modular support vector machine (M^3-SVM) [9], was proposed to overcome the drawback of traditional SVMs. With two module combination principles, namely the minimization principle and the maximization principle, and a random task partition strategy, M^3-SVM has been successfully applied to various fields such as text classification [9], face recognition [10], and gender recognition [11].

However, how to choose a task decomposition strategy according to integration principle to make parallel learning method work efficiently? In essence, the integration principle of M^3-SVM is to dynamically choose a appropriate local learner to classify an input sample. From this point of view, training data should be partitioned according to its distribution in feature space. This law is extremely needed when all samples in training data set are not identically distributed. Random data partition ignores this law and works not very well [12]. In this paper, a novel clustering method is proposed to decompose large training data set into many smaller subsets, which are roughly the same in size. All the experiments show that the proposed clustering method can not only promote the M^3-SVM's generalization accuracy, but also speed up training and reduce the number of support vectors (SVs) in comparison with the traditional random task decomposition method.

II. Min-Max Modular Support Vector Machine

Min-max modular (M^3) neural network [8] is a realization of the principle "divide-and-conquer" in machine learning. By M^3-neural network, a large-scale problem is divided into many smaller and simpler subproblems. In essence, M^3-neural network is a general framework for machine learning.

M^3-neural network can easily address multiclass problems. Here for simplicity of discussion, only two-class classification problems are considered. Given positive training data set $\mathcal{X}^+ = \{(X_i^+, +1)\}_{i=1}^{N^+}$ and negative training data set $\mathcal{X}^- = \{(X_i^-, -1)\}_{i=1}^{N^-}$, where $X_i^+ \in R^n$ and $X_i^- \in R^n$ denote the ith positive and negative training sample respectively, and N^+ and N^- denote the number of positive and negative training samples, respectively. The entire training data set can be defined as $\mathcal{S} = \mathcal{X}^+ \bigcup \mathcal{X}^-$.

In training phase, min-max modular method decomposes $\mathcal{X}^+$ and $\mathcal{X}^-$ into K^+ and K^- roughly equal subsets respectively by using a data partition strategy.

$$\mathcal{X}^+ = \bigcup_{i=1}^{K^+} \mathcal{X}_i^+,\ \mathcal{X}_i^+ = \{(X_m^+, +1)\}_{m=1}^{N_i^+},\ i = 1, 2, ..., K^+$$
$$\mathcal{X}^- = \bigcup_{j=1}^{K^-} \mathcal{X}_j^-,\ \mathcal{X}_j^- = \{(X_n^-, -1)\}_{n=1}^{N_j^-},\ j = 1, 2, ..., K^- \quad (1)$$

where $\bigcap_{i=1}^{K^+}\mathcal{X}_i^+ = \Phi$, $\bigcap_{j=1}^{K^-}\mathcal{X}_j^- = \Phi$; Φ denotes empty set; $N_i^+ = \lfloor N^+/K^+ \rfloor$ for $i = 1, 2, ..., K^+ - 1$, $\lfloor y \rfloor$ denotes the largest integer that is less or equal than y; $N_{K^+}^+ = N^+ - \sum_{i=1}^{K^+-1} N_i^+$; $N_j^- = \lfloor N^-/K^- \rfloor$, for $j = 1, 2, ..., K^- - 1$; and $N_{K^-}^- = N^- - \sum_{j=1}^{K^- -1} N_j^-$. According to (1), every two subsets from $\mathcal{X}^+$ and $\mathcal{X}^-$ respectively are chosen to construct one two-class classification subproblem, so the original classification problem can be divided into $K^+ \times K^-$ smaller classification subproblems as follows:

$$S_{i,j} = \mathcal{X}_i^+ \bigcup \mathcal{X}_j^-,\ i = 1, 2, ..., K^+,\ j = 1, 2, ..., K^- \qquad (2)$$

Because these $K^+ \times K^-$ smaller subproblems need not to communicate with each other in training phase, they can be handled parallelly or sequentially by standard SVM method. Therefore, $K^+ \times K^-$ individual classifiers, $SVM_{i,j}$, $i = 1, 2, ..., K^+, j = 1, 2, ..., K^-$ will be obtained. All subsets in each class with roughly equal size can make load balance among all subproblems $S_{i,j}$, $i = 1, 2, ..., K^+$, $j = 1, 2, ..., K^-$, but it should be pointed out that SVMs' training time does not only rely on problem size.

In recognition phase, a sample X is presented to all classifiers, $SVM_{i,j}$, $i = 1, 2, ..., K^+, j = 1, 2, ..., K^-$ and each classifier outputs a decision value that can be denoted by $SVM_{i,j}(X)$. Then, min-max modular method uses two module combination principles to integrate them. By using "minimization" principle, K^- individual classifiers are integrated as follows:

$$G_i(X) = min_{j=1}^{K^-} SVM_{i,j}(X),\ i = 1, 2, ..., K^+ \qquad (3)$$

where "min" operation chooses the minimum value among K^- decision values of $SVM_{i,j}(X)$, $j = 1, 2, ..., K^-$. After "minimization" principle, "maximization" principle is used to give the final decision value for X:

$$C(X) = max_{i=1}^{K^+} G_i(X) \qquad (4)$$

where "max" operation selects the maximum value among K^+ decision values of $G_i(X)$, $i = 1, 2, ..., K^+$. At last, X can be classified according to the final decision value of $C(X)$. A M^3-SVM for two-class classification problem is shown in Fig. 1.

III. A New Clustering Method for Data Decomposition

The data decomposition problem can be described as follows: given a data set $\mathcal{X} \subset R^n$, m subsets $C_i \subset R^n$ $(i = 1, 2, ..., m)$ are found to satisfy $\bigcup_{i=1}^{m} C_i = \mathcal{X}$ and $\bigcap_{i=1}^{m} C_i = \Phi$. When training data are not identically distributed, the effectiveness of random data partition method will be unstable and sometimes becomes very bad [12]. In order to handle these problems, one needs to generate spatially localized clusters that contain (nearly) equal number of samples to keep load balance. Based on the algorithm "GeoClust" [13], a modified clustering method is proposed in this paper. In our proposed method, the way to generate balanced spatially localized clusters needs to solve the unconstrained nonlinear programming problem as follows:

$$\underset{c_1, c_2, ..., c_m}{\text{Minimize}} : h = max_{i=1}^{m} |W_i - \overline{W}| \qquad (5)$$

where c_i and W_i denote the center vector and number of samples in the ith cluster, respectively, and m denotes the number of partition. $\overline{W}$ is the mean number of samples per cluster, i.e., $\overline{W} = \lfloor \frac{N}{m} \rfloor$. The object function h evaluates the degree of balance among all clusters, the smaller the value of h is, the more balanced all clusters get. The algorithm is shown in Fig.2.

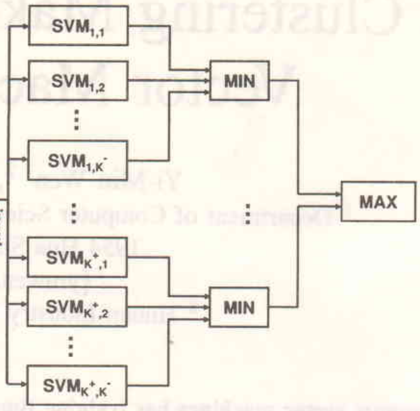

Fig. 1. A M^3-SVM network consisting of $K^+ \times K^-$ individual SVM classifiers, K^+ MIN units, and one MAX unit.

In our proposed equal clustering method, if two cluster contain different number of samples, the center c_i of smaller cluster C_i will move toward to the center c_j of larger cluster C_j while c_j will move toward along the direction of the vector $c_j - c_i$. Two parameters α and δ_i control the movement degree of the centers. In the paper, we set $maxiter = 6000$, $\alpha = 0.01 * 10^{-\lfloor \frac{(m-1)}{10} \rfloor}$, $l = 3$, and $\varepsilon = \lfloor \frac{N}{50m} \rfloor$, respectively.

Apparently, the time complexity of this cluster method is less or equal than $O(m * maxiter)$ and there is no memory thrashing in our method even though data set is very large.

Comparing to the algorithm GeoClust, there are two modifications made to improve it. The first is that, a program break-up condition, "If $h < \varepsilon$, then $break$", is added into the algorithm. This modification can ensure that the algorithm stops at once if the prescribed balance condition is satisfied. If this program break-up condition is cancelled, the partition may become imbalanced again when $maxiter$ is reached, because h does not monotonically decrease in iteration process. The second is that, the proposed method is more proper for partitioning large-scale data set, because the quantity $\frac{W_j}{W_i}$ in GeoClust can take high possibility to become very large and make the two related centers c_i and c_j move too long distance. This will lead to that the new data distribution between these two clusters is still very imbalanced. By modifying $\frac{W_j}{W_i}$ to $\frac{l*W_j}{W_j+(l-1)W_i}$, the movement degree of these two centers is controlled by parameter l. In order to compare the performance

Equal clustering algorithm

```
Input:        Data set: $\mathcal{X} = \{X_i\}_{i=1}^N, X_i \in R^n$
              Prescribed number of partition: $m$
              Maximum number of iterations: $maxiter$
              Learning parameters: $\alpha$ and $l$
              The threshold of the value of object function: $\varepsilon$
Initialize:   randomly choose $m$ vectors in $\mathcal{X}$ , $c_i^0, i = 1, 2, ..., m$, as center vectors of each cluster
              $m$ clusters $C_i = \{c_i^0\},\ i = 1, 2, ..., m$
Algorithm:    For t=1: $maxiter$
                  1. Assign each sample $X_j \in \mathcal{X}$ to the cluster nearest according to the distances between
                     this sample and all center vectors. At the same time, the label indices of cluster that
                     this sample belong to are recoded:
                     $subclusterlabel[j] = arg\ min_{i=1}^m \|X_j - c_i^{t-1}\|,\ j = 1, 2, ..., N$
                  2. Compute the numbers of samples in each cluster: $W_1, W_2, ..., W_m$
                  3. Compute the value of object function $h$
                  4. If $h < \varepsilon$, then $break$
                  5. For each cluster $C_i(\ i = 1, 2, ..., m)$, Update its center as follows:
                      5.1 Compute $\delta_i = \sum_{j=1,j\neq i}^m (\frac{l*W_j}{W_j+(l-1)W_i} - 1)(c_j^{t-1} - c_i^{t-1})$
                      5.2 Update center $c_i^t = c_i^{t-1} + \alpha\delta_i$
                  6. End
              End
```

Fig. 2. Equal clustering algorithm

TABLE I

THE PROBLEM STATISTICS AND THE PARAMETERS USED IN SVMS.

Problems	#attributes	#class	#training data	#test data	c	σ
Banana	2	2	40000	49000	316.2	0.707
Letter recognition	16	2	15000	5000	16	4

of these two cluster methods, a quantity of $h \times iternum$ is introduced, where the variable $iternum$ ($iternum \leq maxiter$) means the number of iteration in each partitioning. Obviously, the smaller the quantity $h \times iternum$ is, the better performance the algorithm gets. The performance comparison between GeoClust and the proposed equal clustering method is shown in Fig. 3. It can be seen that the proposed cluster method performs better than GeoClust.

IV. EXPERIMENTS

A. Experimental Setup

In order to evaluate the performance of the proposed clustering algorithm, simulation experiments are performed to compare the performances of M^3-SVMs using random partition and equal clustering partition. The first data set is Banana data set that comes from [14] and the second one is Letter recognition data that comes from UCI [15]. For banana data, all 100 of its realizations are united to construct a large training data set and one realization of its test data is randomly chosen as test data set. Letter recognition problem is transformed into a two-class classification problem by the method like in [16]. The classes of "E","H","I","M","P","Q","R","X","Y", and "Z" are randomly set to positive class while the rest classes are set to negative class. The training and test data are normalized in the range [0, 1]. From our experience, the data of these two experiments are not identically distributed. The problems statistics and the selection of the parameters for SVMs are shown in Table. I.

In this paper, libSVM [17] whose cache is set to 100M is selected as training tool. All the experiments are performed on a PC that has 3.0GHz CPU with 1GB RAM. The kernel used is the radial-basis function, $\exp(-\frac{1}{2\sigma^2}\|X - X_i\|^2)$. For simplicity, we let $K^+ = K^- = K$ in M^3-SVM method. In order to systematically evaluate the proposed method, the value of K is set to 2,3,...,20 in these experiments. $K = 1$ means that the classifier is trained by the entire training data. The training time of min-max modular support vector machines is counted in two ways. One is sequential training time that is the sum of the training time of all subclassifiers, and the other is parallel training time that is the longest training time among all the

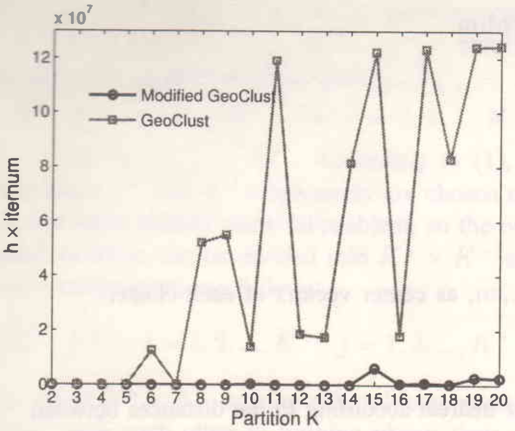

Fig. 3. Performance comparison between the modified GeoClust and the original GeoClust.

classifiers. In order to ensure the credibility of the conclusions, all experiments are repeated three times and the average is taken.

B. Experimental Results and Discussions

In all the figures below, "RP" means random partition method and "CE" means equal clustering method. For convenience, M^3-SVM-RP denotes min-max modular support vector machine based on random partition method, while M^3-SVM-CE denotes min-max modular support vector machine based on equal clustering method.

From Fig. 4, it can be seen that the generalization ability of M^3-SVM is better than standard SVM. The most interesting phenomenon is that M^3-SVM-CE takes higher generalization accuracy than M^3-SVM-RP does in most of time. Fig. 5 illustrates that M^3-SVM-CE generates less support vectors than M^3-SVM-RP does. In Fig. 6, even considering the time used in data partition shown in Table II, M^3-SVM-CE takes less time than M^3-SVM-RP does in sequential mode. From Table II, it can be seen that if the data partition time is ignored, M^3-SVM-CE can run faster than M^3-SVM-RP does in parallel mode. The reason is that CE can make the classification subproblems more separable than RP does. On the other hand, even though the data partitioning costs some time, the fewer support vectors and the higher generalization accuracy can compensate for it. The smaller the number of support vectors is, the less the test time and the cost of realization of M^3-SVM are. In sum, equal clustering method makes min-max modular support vector machine more efficient than random partition does.

TABLE II

THE TIME COST OF TWO DATA PARTITIONING METHODS.

Data	Data partition method	Average time per partitioning (s)
Banana	RP	0.6
	CE	90.1
Letter	RP	1.5
	CE	10.6

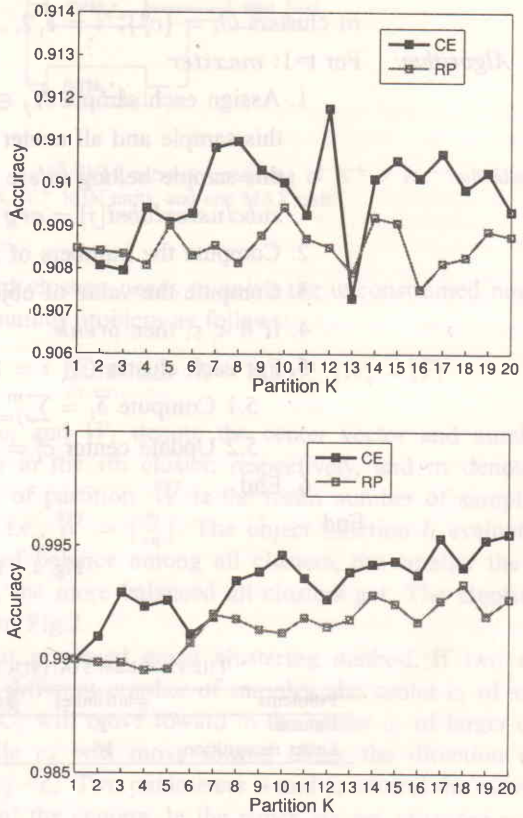

Fig. 4. The upper and lower figures show the accuracy of M^3-SVMs for Banana and Letter recognition problems, respectively.

The reason of equal clustering method performs better than random partition method does can be explained from Figs. 7 and 8. When training data are random partitioned, all the classification subproblems look like the original classification problem as shown in Fig.7. Therefore, all the subclassifiers will look alike and fail to complement each other. In comparison, all the classification subproblems focus on local feature space when training data are partitioned by equal clustering. The subclassifiers are so diverse that the two "min" and "max" integration priciples works effectively. So the generalization accuracy of M^3-SVM-CE is higher than M^3-SVM-RP in most of time. When training data is not identically distributed in feature space, random partition will make the two class samples in each subproblem mix much more while equal clustering method will decrease the mixture degree of two class samples in each subproblem and enhance the separability of two classes in it. Therefore, the number of support vectors can be reduced and the training time is shortened. According to Vapnik [18], the decrease of support vectors will often promote the accuracy of support vector machines.

V. Conclusions

In this paper a new clustering method is proposed to decompose large training data set into a number of smaller subsets. The feature of the proposed equal clustering method

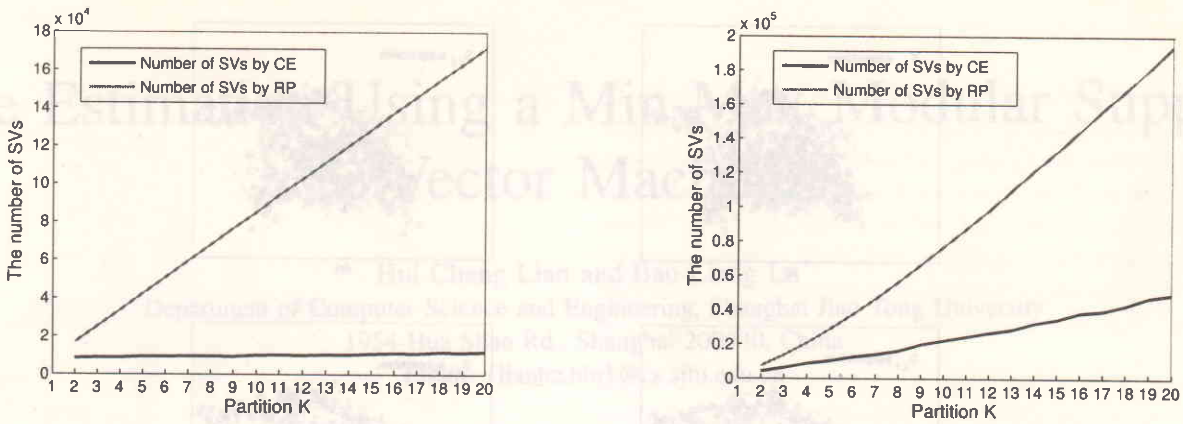

Fig. 5. The left and right figures show the number of SVs produced in Banana and Letter recognition problems, respectively.

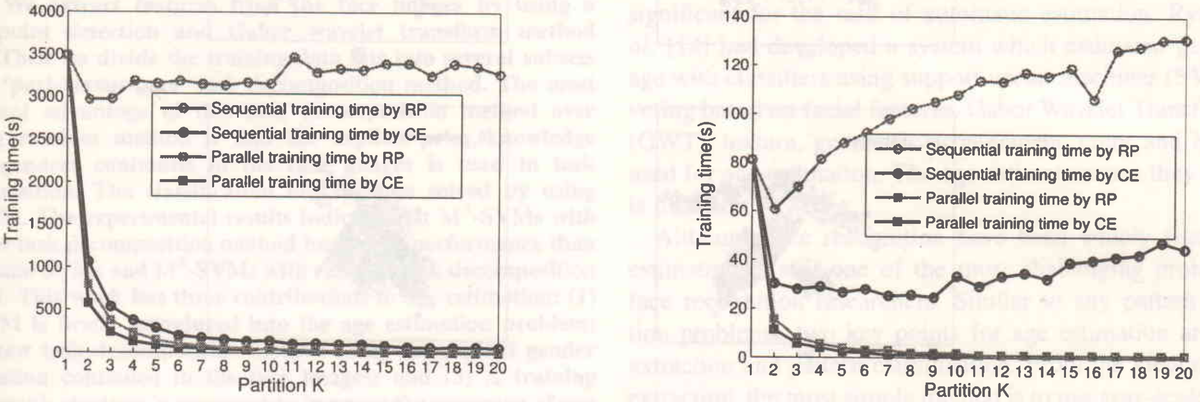

Fig. 6. The left and right figures show both sequential and parallel training time in Banana and Letter recognition problems, respectively.

is that it can partition data set more balancedly than GeoClust does. It has been observed that equal clustering method can catch local probability distribution information in training data, which is ignored by the random task partition method. The simulation results show that when training data are not identically distributed, the proposed clustering method will make min-max modular SVMs more efficient than the random task partition method does.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China via the grants NSFC 60375022 and 60473040. The authors thank Yang Yang for useful advices.

References

[1] D. Boley and D. W. Cao, "Training support vector machine using adaptive clustering," In *Proceedings of SDM'04*. Lake Buena Vista, USA, 2004.

[2] T. Evgeniou and M. Pontil, "Support vector machines with clustering for training with very large datasets," *Lectures Notes in Artificial Intelligence*, vol. 2308, pp. 346-354, 2002.

[3] H. P. Graf, E. Cosatto, L. Bottou, I. Durdanovic, and V. Vapnik, "Parallel support vector machines: the cascade SVM," In *Neural Information Processing Systems*, vol, 17, MIT Press, 2005.

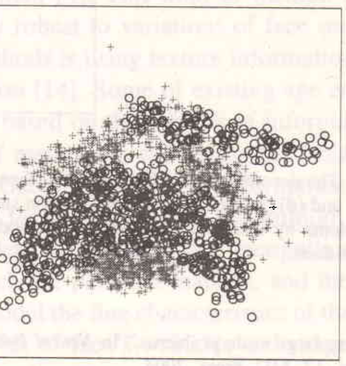

Fig. 7. Distribution of banana training data

[4] Y. M. Wen and B. L. Lu, "A cascade method for reducing training time and the number of support vectors," *Lecture Notes in Computer Science*, vol. 3173, pp. 480-486, 2004.

[5] Y. M. Wen and B. L. Lu, "A hierarchical and parallel method for training support vector machines," *Lecture Notes in Computer Science*, vol. 3496, pp. 881-886, 2005.

[6] F. Provost and J.M. Aronis, "Scaling up inductive learning with massive parallelism," *Machine Learning*, vol. 23, pp. 33-46, 1996.

[7] R. Collobert, Y. Bengio, and S. Bengio, "A parallel mixture of SVMs for

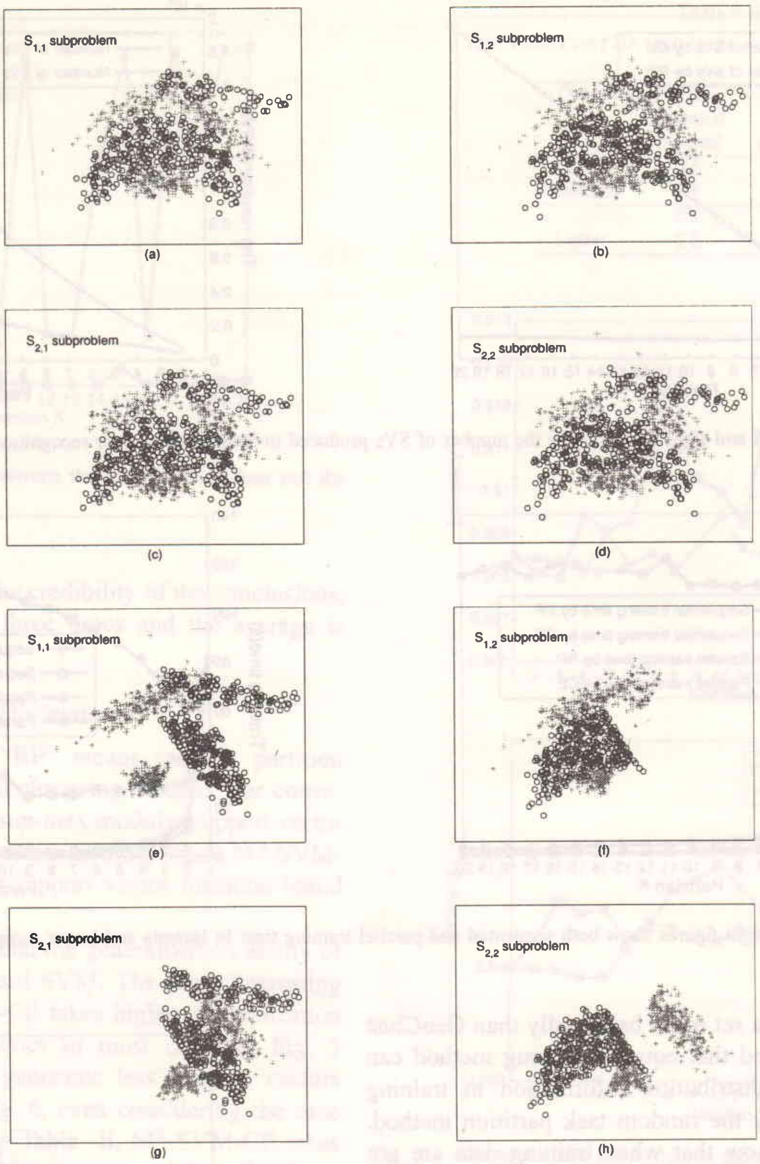

Fig. 8. The decomposition of Banna data when $K = 2$. In all figures, the points labelled by '+' and 'o' denote positive and negative samples respectively. (a), (b), (c), and (d) denote the four classification subproblems by using random task decomposition strategy, and (e), (f), (g), and (h) denote the four classification subproblems by using equal clustering method. It can be seen that equal clustering makes classification subproblems to be more separable than random partition does.

very large scale problems," In *Neural Information Processing Systems*, vol, 17, MIT Press, 2004.

[8] B. L. Lu and M. Ito, "Task decomposition and module combination based on class relations: a modular neural network for pattern classification," *IEEE Trans. Neural Networks*, vol. 10, pp. 1244-1256, 1999.

[9] B. L. Lu, K. A. Wang, M. Utiyama, and H. Isahara, "A part-versus-part method for massively parallel training of support vector machines," In *Proceedings of IJCNN'04*. pp. 735-740, Budapest, Hungary, 2004.

[10] Z. G. Fan and B. L. Lu, "Multi-view face recognition with min-max modular SVMs," *Lecture Notes in Computer Science*, vol. 3611, pp. 396-399, 2005.

[11] F. C. Lian and B. L. Lu, "Gender recognition using a min-max modular support vector machine," *Lecture Notes in Computer Science*, vol. 3611, pp. 438-441, 2005.

[12] K. A. Wang, H. Zhao, and B. L. Lu, "Task decomposition using geometric relation for min-max modular SVMs," *Lecture Notes in Computer Science*, vol. 3496, pp. 887-892, 2005.

[13] A. Choudhury, C. P. Nair, and A.J. Keane, "A data parallel approach for large-Scale gaussian process modeling," In *Proceedings of the second SIAM International Conference on Data Mining*, Arlington, USA, 2002.

[14] G. Rätsch, Available: *http://ida.first.gmd.de/ raetsch/data/benchmarks.htm*

[15] C. L. Blake and C. J. Merz. (1998) UCI Repository of Machine Learning Databases. Univ. California, Dept. Inform. Comput. Sci., Irvine, CA. [online]. Available: *ftp://ftp.ics.uci.edu/pub/machine-learning-databases*

[16] G. Rätsch, T. Onoda, and K. R. Müller, "Soft margins for AdaBoost," *Machine Learning*, vol. 42, pp. 287-320, 2001.

[17] C. W. Hsu and C. J. Lin, "A comparision of methods for multiclass support vector machines," *IEEE Trans. Neural Networks*, vol. 13, pp. 415-425, 2002.

[18] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 1-47, 1998.