The First International Symposium on Optimization and Systems Biology (OSB'07)
Beijing, China, August 8–10, 2007
, pp. 341–350

SVM-based Automatic Classification for Protein Structural Domain *

Xiao-Jian Shao[1] Ying-Jie Tian[2] Nai-Yang Deng[1,†]

[1]College of Science, China Agricultural University, Beijing, 100083, China.
[2]Chinese Academy of Sciences Research Center on
Data Technology and Knowledge Economy, Beijing, 100080, China.

Abstract The automatic classification for protein structure plays an important role in bioinformatics. Here we present an improved multiclass SVM for the classification based on the features of the protein structure which were extracted from the protein convex hull. Firstly, we modify the gauss radial kernel by adding a positive constant to the kernel function. Secondly, we take weighted SVM to deal with the imbalanced dataset. Experiments demonstrate the superiority of our new strategies. In addition, we design the hierarchical classifier which is more suitable to the CATH database.

Keywords Protein Structure; Support Vector Machines; Hierarchical Classification.

1 Introduction

One of the main tasks of biology is to describe and compare biological structures. And the protein structure prediction is the central problem in computational biology. However, there are few known protein structures that are obtained by the more costly and time- consuming experimental methods, which restrict our insight into the structure and the function of the protein. So it is essential to use theory computation and statistical forecast to predict the structure of the protein.

In the post-genomic era, the accumulated DNA sequence information has grown and it is a challenge for us to predict the structure and function of the protein by using this expanded information. Having a computational method based on machine learning to accurately automatic recognize the protein structure will provide us a new insight into the task. Before designing the classifier we must construct some features to describe the protein, and then to compare the similar structures. In recent years, some discriminative methods of machine learning and statistics have been used to solve the classification or prediction of protein structure. These methods are mainly dependent on the information of protein's amino acid sequence such as the works of Cai. et al.[1], Ding and Dubchak [2], E. Eskin and Stafford [3], F. Markowetz [4], A.C. Tan[5], Nitin et al[6] etc. More and more attention of similarity measure

*This work is supported by the National Natural Science Foundation of China (No. 10631070, 10601064).
†Corresponding author: E-mail:dengnaiyang@vip.163.com

of protein structures have been focused on extracting features from geometric patterns. For example, the author proposed a new method based on Gauss Integrals[14] to construct the features of a protein structure and then to group protein shapes in an unsupervised way. And in the thesis [7] (see also [15, 16]), the author proposed another new method regarding the convex hull of the protein as the 3-dimensional structure of the protein, and constructed the classification features according to the convex hull. Furthermore, the author used NN(Neural Networks) method to construct the classifier [7]; however, the NN classifier didn't demonstrate the good generalization on this problem, especially on the testing precision. In this paper, we will build an improved multiclass SVM classifier based on the above features. As far as we know, Wang is the first people who proposed the new features by using the hull of protein to present its 3-D structures [15, 16]. Here our works are mainly dependent on this method, and we compare our results with that of Wang's only.

In section 2 we outline the protein structure prediction problem on the CATH database. Section 3 introduces the multiclass SVM, while Section 4 presents the experimental results of our improved method on multiclass SVM for the CATH database. And finally Section 5 gives the conclusions and discusses the future work.

2 Protein Structure Prediction on CATH Dataset

It is well known that the protein structure has its own different levels including: protein sequence, secondary structure, tertiary structure and quaternary structure etc. We focus on the protein structure domain level due to its overlap of the domain may infect the classification of the protein, and take the protein structure domain as the basic element of the structure comparisons and the classification problems. There are many databases to deal with the protein structure classification such as SCOP [8], FSSP, MMDB and CATH [9] etc. We use the CATH database to construct our multiclass SVM classifier just because it has a more definite standard of the sequence and the structure of the protein. Now the CATH database uses four main levels to classify a protein: Class (C-level), Architecture (A-level), Topology (Fold family, T-level), and Homologous Superfamily (H-level). Based on the principle of the minimum redundancy and the maximum class cover of the classification dataset, finally a dataset of 2771 samples with 93 classification features[1] is constructed. These samples belong to 4 classes, 36 architectures, 622 folds and 1096 homologous superfamilies according to the CATH classification system. These features are mainly based on the convex hull representation of the protein structure. As described in [16], there are roughly four catalogs of these features: Global shape, Protein surface, Interior of protein structure, and Chemical character of amino acids.

The obtained dataset here is imbalanced on the four levels. E.g. in the C-level the *ab* class has 1384 samples while the few secondary structure class has only 82 samples. This lead to an imbalance in the number of the positive samples and

[1] The detailed features could be found in the materials[7,15,16]

the negative samples that is likely to cause misclassification during the process of constructing the multiclass classifier.

So the final problem is how to construct the more accurately multiclass SVM as well as design the SVM that can deal with the imbalance samples.

3 Multiclass Support Vector Machines

Firstly, we briefly give the outline of the support vector machines (SVMs) [10, 11]. SVM was introduced firstly by Vapnik et al for binary classification, and it is based on the statistical learning theory. Given the training dataset$\chi = \{(x_i, y_i) : x_i \in R^n, y_i \in \{\pm 1\}\}_{i=1}^m$, where each x_i is labeled by y_i. SVM defines a boundary that maximizes the margin between the samples of the two classes. The ultimate aim of the classifier is to predict a class label of the newcome samples. The core parts of the SVM include the optimal margin hyperplane and kernel's mapping. The corresponding dual Quadratic Programming is as follows [10]:

$$\min_{\alpha} \ \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j K(x_i,x_j) - \sum_{i=1}^{m}\alpha_i \tag{1}$$

$$s.t. \ \sum_{i=1}^{m}\alpha_i y_i = 0, 0 \le \alpha_i \le C, i = 1,2,...,m \tag{2}$$

Where C is a constant controlling the trade-off between maximizing the margin and minimizing the errors, $K(x,y)$ is a kernel function. The final decision function is

$$f(x) = \mathrm{sgn}\{\sum \alpha_i^* y_i K(x_i,x) + \gamma^*\}. \tag{3}$$

SVMs have demonstrated better performance than neural networks and other methods in many real-world applications such as classifying microarray data[12], image classification and fold recognition[6] etc.

There are many approaches to extend binary SVM to the multiclass SVM. The traditional methods include so called "one-against-one", "one-against-rest" and "altogether" etc. For$k$ classification problem, the "one-against-one" method construct$k(k$-1)/2 classifiers based on the pairwise classifiers, the "one-against-rest" method perform k classifiers, and the "altogether" strategy perform only one optimal programming to construct the classifier. For the prediction scheme, the "one-against-one" takes the voting rule as follows: for a sample, the corresponding class gain one vote if the sample is assigned to the class according to the two-way classifier and then take the highest votes class as the final class of the sample. For the "one-against-rest" method, it takes the highest output of the k classifier function. Differently, the "altogether" method predicts the sample according to the only one decision function. Note, the training time is not very different for each method and the "one-against-one" method may perform more accurately [17].

Many popular methods are used to deal with the imbalance problems such as treating two class samples with different error cost weights, the undersampling and

oversampling techniques etc. Here we introduce a new approach by modifying the diagonal elements of the kernel matrix during SVM optimization, which has achieved success on the gene expression problem [13]. This method can avoid misclassifying the minor class's samples to the other class. The detail modification is as follows: let $K(x,x)$:=$K(x,x)$ +ls/m, where $K(x,x)$ is the diagonal element of the kernel matrix, s is the number of the positive samples if the x is positive sample, otherwise$s$ is the number of the negative samples if the x is negative sample, m is the total number of training dataset of the corresponding two-way classification, and l is a scale factor which can control the accurate ratio. In addition, our experiments show the adopted method can accelerate computation of the optimization procedure.

In this paper, we adopt the "one-against-one" method to solve the multiclass problems on the four levels of the CATH dataset. And we also introduce the weighted method to deal with the imbalance case of the two-way classification issue.

4 Experiments

Comparing with the traditional nonlinear kernel function, we find the linear kernel performs better on the C-level. But they all don't work well on the A, T and H-levels. Fortunately, we get the acceptable results on the all four levels by modifying gauss radial kernel function. The modified kernel function is obtained by adding a positive constant to the traditional gauss radial kernel function, that is $K(x,y)$ = $exp\{-||x-y||^2/2s^2\}$ +b, where b>0. Obviously, the Gram matrix [10] of the new kernel is also positive semi-defined matrix.

In the experiments present here, we adopt both "Resubstitution test (Self-consistency)" and "Cross-validation test" methods to evaluate the classifiers, and consider the sensitivity as the criterion of the correctness of the classifiers. "Resubstitution test" demonstrates how well the classifier has turned into the internal knowledge, while "Cross-validation" mainly shows the generalization of the classifier, usually including the following methods: k-fold CV, LOO, Sub-sampling etc. Let TP_k be the number of the samples which are classified into the kth class correctly, FN_k be the number of the samples which belong to the kth class but are misclassified to other classes, and $SE_k = \frac{TP_k}{TP_k+FN_k} \times 100\%$ be the sensitivity of the kth class samples. Then the total sensitivity of the multiclass SVM classifiers is defined by

$$SE = \frac{\sum_{k=1}^{l} TP_k}{\sum_{k=1}^{l} (TP_k + FN_k)} \times 100\% \tag{4}$$

where l is the number of the classes.

5 SVM vs. NN

Now we present the results of the multiclass SVM on the CATH dataset. Comparing with the results of the Neural Networks (NN) method given in [15], the proposed multiclass SVM performs better with both "Self-consistency test" and "Cross-validation test" criterions. Table (**1**) describes the comparison results on the Class-

level. Here the parameters of the SVM are set as follows by 5-fold cross-validation: $C = 10$, $\sigma = 0.6$, $b = 5$, $\lambda = 0.1$.

Table 1: Comparison of the NN and multi-class SVM on the Class-level (4 classes)

	SVM			NN		
Percentage	*Tr(%)*	*Te(%)*	*Self_T(%)*	*Tr(%)*	*Te(%)*	*Self_T(%)*
50%	98.196	82.166	90.184	98.99	69.33	84.16
60%	98.196	84.386	92.674	98.62	69.15	86.83
70%	98.918	85.078	94.767	98.56	82.19	93.65
80%	98.827	83.755	95.814	98.42	79.27	94.59
90%	98.717	87.004	97.546	98.27	85.17	96.96
100%	99.747	——	99.747	98.70	——-	98.70

Where the percentage in the first column donates the ratio of the selected training dataset to the whole dataset. E.g. the value of 90% means that the total dataset is random divided into two halves, and training sets is occupying 90% of the whole dataset while the rest 10% of the whole dataset is the testing dataset. And *Tr* donates the classifier's accuracy on the training dataset, *Te* donates the correctness on the testing dataset, and *Self_T* is the Resubstitution correctness on the whole dataset.

From the results above, we find that the features based on the convex hull of the 3-D structure of the protein describe the second structure similarity of the protein appropriately. The comparisons of the SVM and NN on the A-level, H-level, and T-level of the protein are demonstrated on Table (**2**), (**3**) and (**4**) respectively. The results show that our new methods are superior to that of NN on the four levels.

Table 2: Comparison of the NN and multiclass SVM on the A-level (36 classes)

	SVM			NN		
Percentage	*Tr(%)*	*Te(%)*	*Self_T(%)*	*Tr(%)*	*Te(%)*	*Self_T(%)*
50%	95.310	55.572	75.470	92.82	28.58	60.70
60%	93.746	56.159	78.750	92.49	20.97	63.88
70%	92.526	56.265	81.661	92.00	34.70	74.81
80%	93.460	58.696	86.529	90.75	35.5	79.07
90%	92.422	61.733	89.354	89.95	24.05	83.36
100%	93.071	———	93.071	89.90	——-	89.90

As mentioned previously, the positive constant b plays a crucial role in the performance of the SVM. In the following, we demonstrate the variety of the accuracy according to the different value of $b$in order to reveal the importance of b. Figure (**1**) illustrates the variety of the correctness according to the different value of b on the A-level.

It is remarkable that in our experiments, we cannot get the precision greater than 35% whatever the parameter C and s take if the value of the b is zero. And the train-

Table 3: Comparison of the NN and multiclass SVM on the T-level (622 classes)

	SVM			NN		
Percentage	*Tr(%)*	*Te(%)*	*Self_T(%)*	*Tr(%)*	*Te(%)*	*Self_T(%)*
50%	95.310	40.307	70.862	94.44	22.20	58.32
60%	94.108	40.805	73.989	93.96	13.51	61.78
70%	96.907	42.511	82.469	93.85	20.05	71.71
80%	96.370	43.428	85.782	92.69	15.94	77.34
90%	95.309	45.868	90.936	93.78	15.28	85.93
100%	96.350	———	96.350	92.82	——-	92.82

Table 4: Comparison of the NN and multiclass SVM on the H-level (1096 classes)

	SVM			NN		
Percentage	*Tr(%)*	*Te(%)*	*Self_T(%)*	*Tr(%)*	*Te(%)*	*Self_T(%)*
50%	91.270	39.151	66.565	93.82	26.28	60.05
60%	92.742	41.376	72.196	93.46	16.16	62.54
70%	93.814	41.286	78.056	93.33	10.89	68.60
80%	93.189	41.912	82.934	93.44	——	74.16
90%	94.146	47.479	89.479	93.82	——	77.52
100%	94.082	——-	94.082	91.56	——	91.56

ing, testing, and self-consistency correctness increase from 31.636% to 88.292%, 26.087% to 55.596%, and 31.083% to 85.023% respectively when b changes from zero to one. Meanwhile it goes best when b=5 which can be shown from the Figure (**1**).

Table 5: Hierarchy classifiers on the A-level

		50%	60%	70%	80%	90%	100%
α	*Tr(%)*	98.635	98.823	97.384	97.282	97.938	99.747
	Te(%)	77.733	78.593	79.357	82.223	82.95	——
	Self_T(%)	88.19	90.754	92.00	94.28	96.549	99.747
β	*Tr(%)*	97.2918	98.380	97.669	98.090	98.817	99.72
	Te(%)	60.977	65.141	62.854	65.493	65.722	———
	Self_T(%)	79.281	85.196	87.291	91.643	95.945	99.72
$\alpha\beta$	*Tr(%)*	93.84	92.720	91.620	90.914	95.707	97.69
	Te(%)	60.756	60.018	62.128	60.748	60.688	———
	Self_T(%)	77.397	79.674	82.787	84.899	92.215	97.69
*Few**	*Self_T(%)*	———	———	———	———	———	——-
Hierarchy SVM	*Tr(%)*	93.011	92.774	91.736	91.471	94.192	95.80
	Te(%)	62.641	63.535	64.157	64.759	64.942	——-
	Self_T(%)	77.915	81.129	83.492	86.158	91.408	95.80

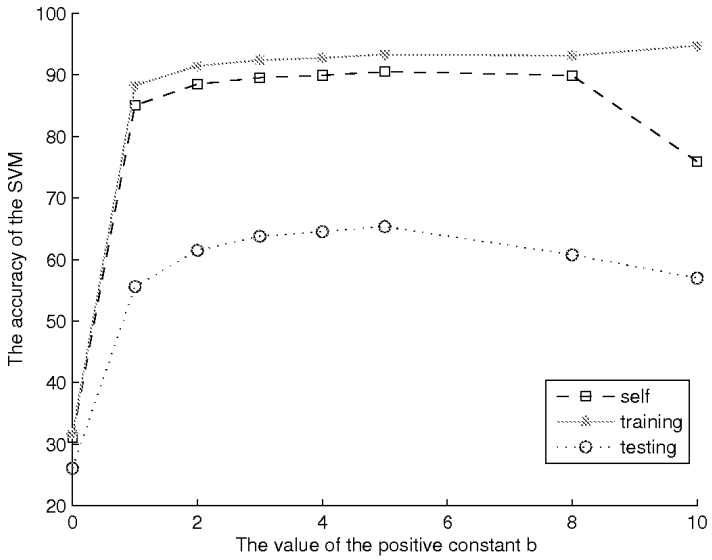

Figure 1: Precision varies according to b

6 Hierarchy Classifiers

In addition, we introduce the hierarchy classifiers. As discussed above, the CATH dataset have four main levels and there are closely hierarchical relationships of the four levels. For example, there are four classes on the Class-level and they have their own subclasses. The mainly *a* class has five subclasses, the mainly *b* class has eighteen subclasses, the *ab* class has twelve subclasses and the few secondary structure class has only one subclass. The total 36 classes constitute the class of our CATH dataset on the Architecture level. And the similarity has happened on the T-level and H-level. So we can construct multiclass SVM classifiers according to the subclass datasets respectively and classify the data from C-level to the lower level hierarchically. Here we denote the previous SVM as "Standard SVM" and the new classifiers as "Hierarchy SVM". Now we design the hierarchy classifiers on the A-level, that is to say, construct the classifier on the subclasses of the four classes on the C-level independently. And then evaluate the A-level classifier by combining the four sub-classifiers. Table (**5**) describes the results of the subclasses of the C-level and the hierarchy results according to the formula (4).

*Remark: The few secondary structure class belongs to only one class on the A-level, so we define the correctness of the subclassifier is 100%

It is reasonable to think that the "Hierarchy SVM" performs better than the "Standard SVM". In the following experiments we compare the performance results of the two SVM classifiers which are constructed with the same training dataset and testing dataset. The results based on the A-level are presented on Figure (**2**) where the

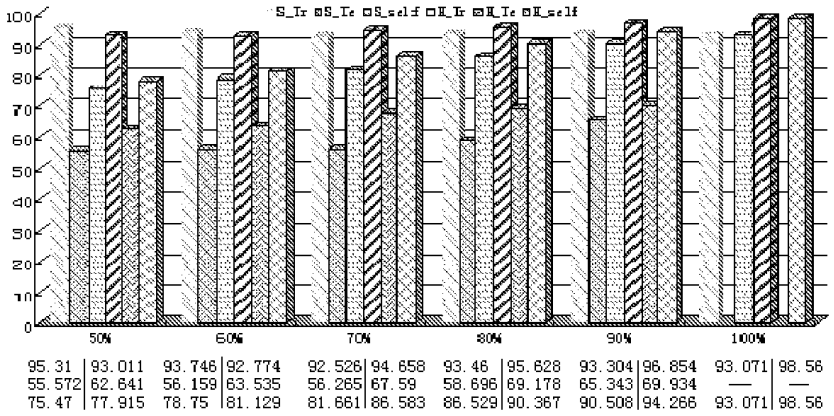

Figure 2: Standard SVM vs. Hierarchy SVM on A-Level

first three series are the correctness of the "Standard SVM", and the other three are that of the "Hierarchy SVM". At the bottom of the figure there are the corresponding accuracy of the two classifiers on the training datasets which occupy the different percentage of the whole dataset. And the results validate our hypothesis that the "Hierarchy SVM" implements better than the "Standard SVM".

Similarly, we construct the "Hierarchy SVM" on the T and H-levels, and compare them with the "Standard SVM". Because there are a large number of the classes, here we don't display the detailed results of the "Hierarchy SVM" on these subclasses. We only show the comparison of the "Standard SVM" and "Hierarchy SVM" on the T, H-levels. The results are displayed on Figure (**3**) & (**4**).

7 Conclusion and Future Work

In this paper, we proposed a new multiclass SVM for the protein structure prediction based on the CATH dataset. The features of the classification are based on the convex hull of the 3-Dimension structure of the protein. The improvements of the new SVM include that modify the traditional gauss kernel by adding a positive constant to the kernel function and take weighted SVM when training datasets of the two-way scenario is imbalanced. It performs batter than NN on all four levels according to both "Resubstitution test" and "Cross-validation test" methods, especially for the cross-validation correction. In the future work, we focus on the reconstruction of the features based on the Gauss integrals[14], as well as combining SVM with other classification methods such as Hidden Markov Models (HMM) to design more efficient classifier for the T and H-levels.

References

[1] Cai, Y. D. Liu, X. J. Xu, X. B. and Zhou, G. P. (**2001**) *BMC Bioinformatics*, 2:3.

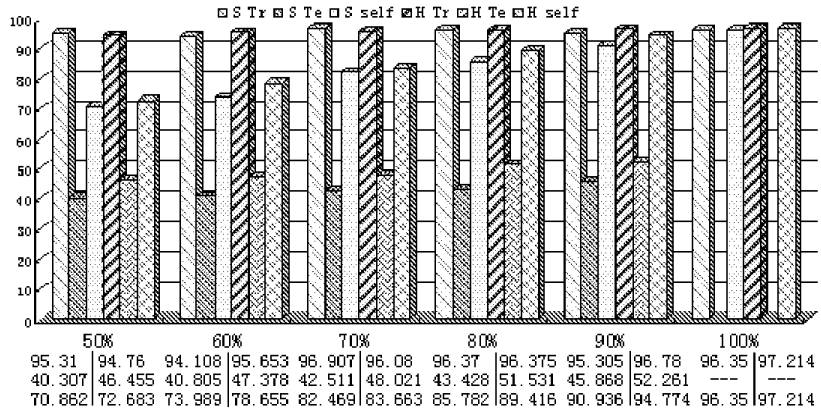

Figure 3: Standard SVM vs. Hierarchy SVM on T-Level

[2] Chris H.Q. Ding, Inna Dubchak (**2001**) *Bioinformatics,* Vol.17, No.2, 349-358.

[3] Eskin, E. Stafford, N. W. (**2002**) *Proc. Pacific Symposium on Biocomputing,* 7, 566–575.

[4] Markowetz, F. Edler, L. and Vingron M. (**2003**) *Biometrical J.*, Vol.45, No.3, 377-389.

[5] Tan, A.C. Gilbert, Deville, D. Y. (**2003**) *Genome Informatics,* 14, 206-217.

[6] Bhardwaj, N., Langlois, R. E. Zhao, G. J. and Lu, H. (**2005**) *Nucleic Acids Research,* 20, 6486-6493.

[7] Wang Y. (**2005**), *Research on protein structure prediction and classification using neutral network,* PhD. Thesis, Institute of Applied Mathematics, Academy of Mathematics and Systems Science, BeiJing.

[8] Conte, L. Ailey, L. Hubbard, B. Brenner, T.J.P. Murzin, S.E. A.G. and Chothia, C. (**2000**) *Nucleic Acids Res.* 28, 260-262.

[9] Orengo, C.A. Michie, A.D. Jones, S. Jones, D.T. Swindells, M.B. and Thornton, J. M. (**1997**) *Structure,* 5, 1093-1108.

[10] Deng, N. Y. Tian, Y. J. (**2004**) *A New Method in DataMining: Support Vector Machine.* Science Press, BeiJing.

[11] Vapnik V. N. (**1995**) *The Nature of Statistical Learning Theory*. Springer, New York.

[12] Langlois, R. E. Diec, A. Dai, Y. Lu, H. (**2004**) *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, 2885-2888.

[13] Brown, M. P. S. Grundy, W. N. Lin, D. Cristianini, N. Sugnet, C. W. Furey, T. S. Manuel, A. Jr., and Haussler, D. (**2000**) *PNAS,* Vol. 97, No. 1, 262-267.

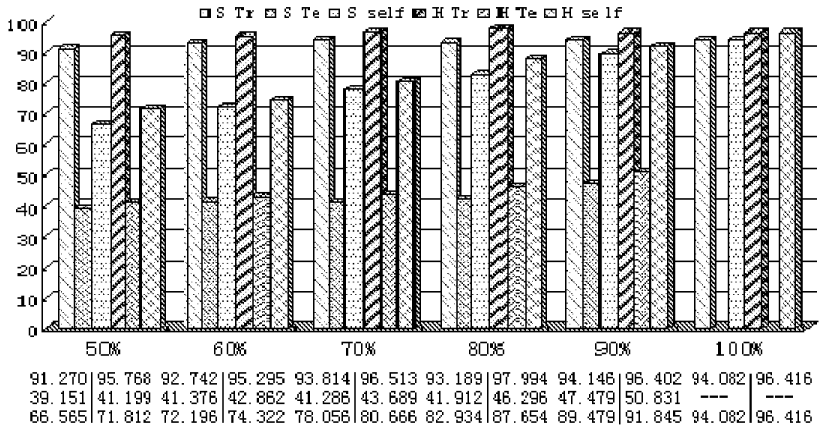

Figure 4: Standard SVM vs. Hierarchy SVM on H-Level

[14] Rôgen, P. and Fain B. (**2003**) *PNAS,* Vol.100, No.1, 119-124.

[15] Wang, Y. Wu, L. Y. Zhang, X. S. and Chen, L.N. (**2006**) *International Journal of Computational Intelligence Research*, Vol.2, No.1, 105-109.

[16] Wang, Y. Wu, L. Y. Chen, L.N. and Zhang, X. S. (**2006**) *International Journal of Bioinformatics Research and Applications*, Vol. 3, No. 2.

[17] Hsu,C.W. Lin, C.J.(**2002**) *IEEE Trans. Neural Networks,* 13,415-425.