行政院國家科學委員會專題研究計畫　成果報告

結合輸出錯誤校正碼與支持向量機對蛋白質結構作分類預測

計畫類別：個別型計畫
計畫編號：NSC94-2213-E-164-008-
執行期間：94 年 08 月 01 日至 95 年 07 月 31 日
執行單位：修平技術學院電機工程系

計畫主持人：黃淳德

報告類型：精簡報告
報告附件：出席國際會議研究心得報告及發表論文
處理方式：本計畫可公開查詢

中　華　民　國 95 年 10 月 24 日

行政院國家科學委員會專題研究計畫結案報告書

結合輸出錯誤校正碼與支持向量機對蛋白質結構作分類預測

Protein Structure Classification Prediction by using the combination of Support Vector Machine with Error Correcting Output Coding

計畫編號：NSC 94-2213-E-164-008

修平技術學院 電機工程系
副教授 黃淳德

摘要

利用人工智能的協助提供正確的預測作蛋白質的分類工作,使得生物學家可以縮短實驗的時間。於是，如何提供正確的預測，變成了一個重要的課題。

在這個研究中，我們基於我們先前的研究，並把 SVM 結合 ECOC，嘗試找出一個更有效的分類法則，以便對蛋白質的結構分類作預測。

ECOC SVM 法則將用來解決多類分類器的問題，它可以將問題化為一群二值的問題，並結合這些結果作修正來預測多類的問題。首先 ECOC SVM 將會指定一組長度為 n 的二元碼，並進行訓練，利用梯度的方法決定了最適合的 SVM 核心的參數，並形成了一組獨立的二元 SVM 分類器。接著在測試階段，將利用一組沒有標記的資料,評估每一個二元分類器的輸出以得到 ECOC 矩陣中最接近的向量。

我們利用 ECOCSVM 的方法，結合我們先前提出的階層式學習架構，對 SCOP 中蛋白質分類作預測的驗證。實驗結果顯示可以得到良好的準確度，若要得到更好的準確度，則可以利用不同長度的碼元以得到更好的分類器。

關鍵字：支持向量機、輸出錯誤校正碼、階層式學習架構、SCOP、生物資訊、蛋白質

Abstract

The artificial intelligence aid classification of proteins helps bio-researchers to shorten their experiments, for the prediction can exclusive those wrong answers. But how to separate the right and the wrong properly becomes an important task.

In this work, we based on our prior researches and combine SVM with ECOC to develop an efficient classifier and apply to the classification of proteins structure.

The ECOC SVM is an approach for solving multi-class classification problem by reducing it into a group of binary classification tasks and combines the binary classification results to predict the multi-class target labels. First, the ECOC SVM assigns a unique binary string of length n, called ‘codeword’, for every class to distinguish each other. Then, the n binary classifiers are trained, one for each bit position in the codeword. To improve the performance of each binary classifier, a gradient descent method is used to determine the better penalty parameters and kernel parameters of SVM adaptively. After the training phase, a set of independent binary SVM classifiers, with their parameters, are constructed automatically. In the testing phase, the unlabeled data are predicted by evaluating each output result of each binary classifier and finding the closest vector in the ECOC matrix.

We apply the ECOCSVM to the protein fold classification problem in SCOP with HLA, which has proposed by us. Experimental results show that the proposed scheme can achieve good classification accuracy. To further enhance the overall accuracy, different codeword of different length are also applied to analyze the classification performance and to obtain better results.

Key Words: SVM, ECOC, HLA, SCOP, bioinformatics, protein

1. Introduction

It is known that proteins are formed by 20 kinds of amino acid but the functions are truly decided by the structures of proteins. The sequences of amino acid are called the first structure of proteins, and the basic forms of proteins are called second structure of proteins. There are three basic forms of second structure of proteins, named helix, sheet and plate.

Since the three-dimensional coordinate structures provide insight into the function, mechanism and evolution of protein, there are several famous classification databases existed such as SCOP, CATH, DDBASE, Entrez, and 3Dee, which imbue the structures with context and analysis. These different classification databases of proteins focus on their own characteristics. For example, SCOP provides a detailed description of the structural and evolutionary relationships of the proteins of known structure. Recently, protein classification and protein fold prediction have been solved by the aid of computer with the strong ability of computation [1, 2, 3]. Computational methods have been developed for the assignment of a protein sequence to a folding class in the SCOP also.

In the prior researches, we proposed a hierarchical learning architecture (HLA) to solve the problems of classification. And we have found that researchers have used primary global protein sequence in terms of three descriptors as physical, chemical, and structural properties of the constituent amino acids to code the sequences, Table 1. In addition to the aforementioned traditional global features, other local features describing the chain of amino acids representing proteins called the bi-gram and spaced-bi-gram are also used in feature coding also proposed and used in our experiments. [4]. Machine learning methods such as Neural Network and Support Vector Machine have been induced into this complex classification problem.[1,2,4,6]

Table 1. The descriptors and feature dimension sizes of each of the six protein attributes (protein sequence information --- PSI).

Characteristics	Descriptors			Feature Size
Composition (C)	20 kinds of amino acids			20
Predicted Secondary Structure (S)	Alpha	Beta	Loop	21
Hydrophobicity (H)	Positive	Neural	Negative	21
Volume (V)	Large	Middle	Small	21
Polarity (P)	Positive	Neural	Negative	21
Polarizability (Z)	Strong	Middle	Weak	21
Total Number				125

Support Vector Machine (SVM) is a discriminative method based on the statistical learning theory and has been widely used in many applications including the complex problems of bioinformatics [1,2,5,6]. In this research, a different multi-class support vector machine algorithm, called self-tuning error correcting output coding (ECOC) SVM is proposed to deal with the multi-class protein fold classification problem. The ECOC SVM is an approach for solving multi-class classification problem by reducing it into a group of binary classification tasks and combines the binary classification results to predict the multi-class target labels.

2. Proteins, Protein databank

Structure Classification of Protein (SCOP) is a famous protein databank, which uses the evolution and similarity of proteins to classify the structure of proteins. The data structure of SCOP is found according to the hierarchical structure of proteins. In the SCOP, the main classes are divided into several classes. The main classes, with most numbers of protein, are all alpha(α), all beta(β), alpha/beta (α / β) and alpha and beta ($\alpha + \beta$). These four classes are named by the structure of proteins [7, 8, 9, 10]. The protein classification in SCOP was performed manually or semi-automatically, which takes a great amount of time for such a complex task.

2.1 Training Dataset in Experiments

This training dataset was built for the prediction of protein folds based on the PDB selected sets. The data set was selected by their characteristics so that all proteins in the data set have less than 35% of the sequence identity for the aligned subsequences longer than 80 residues. Following the prior published papers [4, 5, 6, 8], the training data number is 313 and they should be divided into 4 classes with 27 folds according to their structures representing all major structural classes.

2.2 Testing Dataset in Experiments

The testing dataset was based on PDB-40D set developed by the authors of the SCOP database [7,8,9,10]. A total number of 385 proteins with identity less than 40%, same as the prior used, were selected for testing in our research. .Table 2 shows the numbers of proteins in the training and testing datasets for different protein classes used in our experiments. Table 3 shows the numbers of proteins in the training and testing datasets for different folds of each protein class used in our experiments, where there are 27 folds for the 4 classes in total.

Table 2. Pattern numbers of each classes in SCOP which was picked up to be training and testing patterns in this study.

Classes	Pattern Number (Training Data)	Pattern Number (Testing Data)
All Alpha	55	61
All Beta	109	117
Alpha/Beta	115	145
Alpha+Beta	34	62
Total Number	313	385

Table 3. Fold numbers of each class and pattern numbers of each fold in SCOP which was picked up to be training and testing patterns in this study.

Classes	Fold number per class (Training pattern per fold)		Fold number per class (Testing pattern per fold)	
All Alpha	6	13,7,12,7,9,7	6	6,9,20,8,9,9
All Beta	9	30,9,16,7,8,13,8,9,9	9	44,12,13,6,8,19,4,4,7
Alpha/Beta	9	29,11,11,13,10,9,10,11,11	9	48,12,13,27,12,8,14,7,4
Alpha+Beta	3	7,13,14	3	8,27,27
Total Number	27		27	

3. SVM, Support Vector Machines

Support Vector Machine (SVM) is a typical two-class classifier and a kind of universal feedforward network. The SVM will construct a hyperplane in a high-dimensional features space as the decision surface between positive and negative patterns. The structural risk minimization ability makes the SVM a very efficient classifier in various applications including biosequences analysis also [11,12,13,14].

With the further improvements by other researchers recently, the SVM has the ability to do multi-class classification directly, which is the model adopted here in our HLA as the constituent multi-class classifiers. [15]

Given the training set $\mathbf{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$ with explanatory variables $\mathbf{x}_i \in \mathbf{R}^d$ and the corresponding binary class labels $y_i \in \{-1, +1\}$, for all $i = 1, \cdots, l$, where l is the number of data, and d is the dimension of the problem, we wish to find a separating d-dimensional hyperplane described by

$$\mathbf{w} \cdot \mathbf{x} + b_0 = 0, \tag{1}$$

where $\mathbf{w} = [w_1, w_2, \cdots, w_l]$ is the set of linear weights,

$\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l]$ is the input dataset, and

b_0 is the constant.

The separation problem is to determine the hyperplane such that $\mathbf{w} \cdot \mathbf{x} + b_0 \geq +1$ for positive examples and $\mathbf{w} \cdot \mathbf{x} + b_0 \leq -1$ for negative examples. If it is separated without error and the distance between the closest vector to the hyperplane is maximal, this set of vectors is separated by the optimal hyperplane. In this connection, the SVM is used to find a hyperplane to maximize the function M as follow:

$$M = \frac{2}{||W||}. \tag{2}$$

The solution of Eq. (2) must satisfy the following constraints of inequality type

$$y_i\left[\left(\mathbf{x}_i \cdot \mathbf{w}\right)+b_0\right] \geq 1 \qquad i=1,2.....,l\,. \tag{3}$$

This is a classic nonlinear optimization problem with inequality constraints. Such an optimization problem can be solved by the saddle point of the Lagrange function

$$L(\mathbf{w},b,\alpha)=\frac{1}{2}\mathbf{w}^T\mathbf{w}-\sum_{i=1}^{l}\alpha_i\left\{y_i\left[\mathbf{w}^T\mathbf{x}_i+b_0\right]-1\right\}, \tag{4}$$

where α_i's are Lagrange multipliers.

The Lagrangian has to be minimized with respect to **w** and *b* and maximized with respect to α_i>0.

Through mathematics reasoning and calculation, one obtains a standard quadratic optimization problem that can be formulated as follows:

$$\text{Maximize} \quad L_d(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}y_iy_j\alpha_i\alpha_jx_i^Tx_j$$

$$\text{or} \qquad \text{Maximize} \quad L_d(\alpha)=-\frac{1}{2}\alpha^TH\alpha+f^T\alpha$$

$$\text{subject to} \quad \alpha_i \geq 0\,, \quad i = 1, 2,\ldots., l \quad \text{and} \quad \sum_{i=1}^{l}\alpha_iy_i=0\,. \tag{5}$$

Let $\alpha_0=\left(\alpha_1^0,\alpha_2^0,.....\alpha_l^0\right)$ be a solution to this quadratic optimization problem. The separating rule is the following indicator function

$$f(\mathbf{x})=\text{sign}\left(\sum_{i=1}^{N_{sv}}y_i\,\alpha_i^0(\mathbf{x}_i\cdot\mathbf{x})+b_0\right), \tag{6}$$

where N_{sv} denotes the number of support vectors, $\mathbf{x}_i$ are the support vectors, α_i^0 are the corresponding Lagrange coefficients, and b_0 is the constant given by

$$b_0=\frac{1}{2}\left[\left(\mathbf{w}_0\cdot\mathbf{x}^*(1)\right)+\left(\mathbf{w}_0\cdot\mathbf{x}^*(-1)\right)\right], \tag{7}$$

where we denote by $\mathbf{x}^*(1)$ any support vector belonging to the first class and $\mathbf{x}^*(-1)$ a support vector belonging to the second class.

In practical applications for real-life data, the two classes are not completely separable and the separating plane is always a nonlinear function of the data. So the idea of feature space is discovered to solve this problem. The idea in designing a nonlinear SVM is to map an input vector $\mathbf{x} \in \mathbf{R}^d$ into a vector $\mathbf{z}$ of a higher-dimensional feature space F ($\mathbf{z} = \phi(\mathbf{x})$, where ϕ represents a mapping $\mathbf{R}^d \to \mathbf{R}^f$), and to solve a linear classification problem in this feature space:

$$\mathbf{x} \in R^d \to z(\mathbf{x}) = [a_1\phi_1(\mathbf{x}),\ \ a_2\phi_2(\mathbf{x}),\ \ \cdots,\ \ a_n\phi_n(\mathbf{x})]^T \in R^f ,$$

where a_n is constant. Since the computation of the dot products is prohibitive if the dimension of transformed training vectors $\phi(\mathbf{x}_i)$ is very large, and since $\phi(\mathbf{x}_i)$ is not known a priori, the Mercer's theorem for positive definite functions allows to replace $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ by a positive definite symmetric kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$, i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$. So we need to select a kernel function and then solve the following dual quadratic optimization in order to obtain an optimal hyperplane for any linear or nonlinear space:

$$\max_{\alpha} \quad L_d(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to} \quad 0 \le \alpha_i \le C,\ i = 1, 2, ..., l \ \text{ and } \ \sum_{i=1}^{l} y_i \alpha_i = 0 . \qquad (8)$$

The indicator function is

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N_{sv}} y_i \alpha_i^0 k_\Theta(\mathbf{x}_i \cdot \mathbf{x}) + b_0 \right), \qquad (9)$$

where a kernel k_Θ depends on a set of parameters Θ.

4. Error-Correcting Output Coding (ECOC)

Error-correcting output coding (ECOC) is an approach for solving multi-class categorization problems [16]. It reduces the multi-class classification problem to a

group of binary classification tasks and combines the binary classification results to predict multi-class labels.

In supervised classification problems, one is given a training set **S={(x**$_1$, y_1), (**x**$_2$, y_2), $\cdots$, (**x**$_n$, y_n)} $\in$ X, containing n examples. Each sample (**x**, y) consists of an instance $\mathbf{x} \in \mathrm{X}$ and a label $y \in \{1,\ 2,\ \cdots,\ k\}$, where X is the instant space and $k \geq 2$ is the number of classes. A classifier is a mapping $F : X \to \{1,\ 2,\ \cdots,\ K\}$ from instances to labels. In the ECOC method, a $m \times l$ binary code word matrix **M** **(**where $l > \log_2 k$) has one row (code word) for each of k classes, with each column defining one of l sub-problems that use a different labeling. The binary code of error correcting output code is shown in Fig. 1. Specifically, for the jth (j = 1, 2, ..., l) sub-problem, a training pattern with target class $C_i\,(i = 1,\ 2,\ \cdots,\ k)$ is re-labeled as class C_1 if $\mathbf{M}_{ij}$ = b, and as class C_2 if $\mathbf{M}_{ij}$ = $\bar{b}$, where b is a binary variable, typically zero or one. The re-labeling is to consider the k classes as being arranged into two super-classes.

Next, the ECOC classifier will build an individual binary classifier for each column of the code word matrix. To summarize, in the training phase, an ECOC classifier consists of learning a set $\vartheta = \{\vartheta^1,\ \vartheta^2,\ \cdots,\ \vartheta^n\}$ of independent binary classifiers. Then, in the testing phase, the correct class of an unlabeled $\mathbf{x^h}$ **is** hypothesized as follows. First evaluate each independent classifier on x^h, and then generate a n-bit vector $\vartheta(\mathbf{x^h}) = \{\vartheta^1(\mathbf{x^h}),\ \vartheta^2(\mathbf{x^h}),\ \cdots,\ \vartheta^n(\mathbf{x^h})\}$. The generated bit-vector $\vartheta(\mathbf{x^h})$ which is closest to the row of **M** according to some distance will be found. Either Hamming distance Δ or loss-based decoding can be used here.

The performance of the ECOC SVM classifier has been shown to superior to

other multi-class SVM classifiers. However, it will be affected by some factors such as performance of the composing binary classifiers, independence of the binary classifiers, and the loss function. In the next section, an ECOC SVM classifier will be proposed to improve the classification accuracy and design efficiency of the ECOC classifier.

Classes	Code words (each column for one binary)				⋯⋯	
0	-1	-1	-1	-1	⋯⋯	+1
1	-1	+1	-1	+1	⋯⋯	-1
2	+1	-1	+1	-1	⋯⋯	+1
3	-1	-1	+1	-1	⋯⋯	-1
⋮	⋮	⋮	⋮	⋮	⋯⋯	⋮
m	-1	+1	-1	-1	⋯⋯	-1

Fig. 1. Error Correcting Output Code matrix, where *m* is the number of classes (rows) and *l* is the number of classifiers (columns).

5. **The ECOCSVM Classifier**

Since the accuracy of an ECOC SVM classifier is highly affected by the performance of its composing binary classifiers, the performance of a binary SVM classifier is determined by several parameters such as the penalty parameter *C* and kernel parameters. The penalty parameter *C* controls the tradeoff between margin maximization and error minimization. The kernel parameters determine the proper and efficient non-linear mapping from pattern space into feature space. We find out that the best penalty parameter C and kernel parameters to make every binary SVM classifier have the best classification function so that the classification mistakes can be eliminated. Besides, the accuracy of the whole ECOC SVM classifier can be increased globally.

There are many researchers have proposed different methods to improve the problem of kernel model selection [16,17]. We adopt the efficient on-line processing of the gradient descent method to improve the performance of ECOC SVM here, for its efficient. In the SVM methodology, a kernel function which depends on one or several parameters can be encoded into a vector

$$\Theta = (\theta_1, \ \theta_2, \ \cdots, \ \theta_n) \ . \tag{10}$$

Based on the chosen kernel functions, a class of decision functions can be parameterized by α, b, and Θ as follows:

$$f_{\alpha,b,\Theta}(\mathbf{x}) = \text{sign}(\sum_{i=1}^{l} \alpha_i y_i k_\Theta(\mathbf{x}, \ \mathbf{x}_i) + b_0) \, . \tag{11}$$

In addition, there is a penalty parameter C that controls the tradeoff between margin maximization and error minimization as mention above. In our approach, the penalty parameter C as another tunable parameter of kernel functions is considered first. Because the "soft margin" concept proposed [18, 19,20], in which it was shown that the soft-margin SVM with quadratic penalization of errors can be considered as a special case of the hard-margin version with the modified kernel:

$$\mathbf{K} \leftarrow \mathbf{K} + \frac{1}{C}\mathbf{I} \, , \tag{12}$$

where $\mathbf{I}$ is the identity matrix and C a constant for penalizing the training errors. Thus C will be considered as another tunable parameter of a kernel function.

To obtain a proper value of the penalty parameter C and a proper kernel-parameter vector, we first consider the estimation of the generalization error E of the SVM. In pattern classification area, there are several measures of the expected error rate of an SVM. Among these, the single validation error estimate and leave-one-out error estimate are used mostly. Then, we adopt the single validation estimation method to estimate the generalization error of the SVM in our approach.

This estimate is given as follows:

$$E = \frac{1}{N}\sum_{i=1}^{N} \Psi(-y_i f(\mathbf{x}_i)), \tag{13}$$

where Ψ is the step function: $\Psi(x) = 1$ when $x > 0$ and $\Psi(x) = 0$, otherwise; and N is the size of the data set.

The goal is to find the values of the parameters α and Θ such that margin M is maximized and the generalization error E is minimized. With the parameter Θ fixed, we can obtain $\alpha^0 = \arg \max W(\alpha)$ and then choose Θ as

$$\Theta = \arg \min_{\Theta} E(\alpha, \Theta). \tag{14}$$

Because the step function Ψ is not differentiable, we cannot use a gradient descent method to minimize the estimates of generalization errors. To circumvent this problem, Platt proposed the following estimate of the posterior distribution $P(Y = 1 | X = \mathbf{x})$ of an SVM output:

$$P(Y = 1 | X = \mathbf{x}) = \frac{1}{1 + \exp(Af(\mathbf{x}) + B)}, \tag{15}$$

where $f(\mathbf{x})$ is the output of the SVM. The constants A and B in the above equation are found by minimizing the Kullback-Leibler divergence between P and an empirical approximation of P built from a training set $\mathbf{S}=\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)\} \in X$, containing n examples:

$$(A^*, B^*) = \arg \max_{A,B} \sum_{i=1}^{n} \left(\frac{1 + y_i}{2} \times \log(p(\mathbf{x}_i)) + \frac{1 - y_i}{2} \times \log(1 - p(\mathbf{x}_i))\right). \tag{16}$$

The error probability of either target value for a given data example x_i is formulated by

$$E_i = p(y_i \neq z_i) = p_i^{1-t_i}(1 - p_i)^{t_i}, \tag{17}$$

where $z_i = f_i = f(x_i)$ is the corresponding SVM output value, p_i is the estimated posterior probability, and t_i is that $t_i = 1$ if the input vector x_i belongs to class

C_1 and $t_i = 0$ if it belongs to class C_2. For a validation set of size *N*, the average estimate of the error could be written as:

$$E = \frac{1}{N}\sum_{i=1}^{N} E_i = \frac{1}{N}\sum_{i=1}^{N} p_i^{1-t_i}(1-p_i)^{t_i} \ . \tag{18}$$

So the gradient of the generalization error *E* can be computed as follows [26]

$$\frac{\partial E}{\partial \theta_r} = \left.\frac{\partial E}{\partial \theta_r}\right|_{\alpha\text{fixed}} + \frac{\partial E}{\partial \alpha}\frac{\partial \alpha}{\partial \theta_r}, \tag{19}$$

where θ_r is the number in the vector Θ, and *r* = 1, 2, $\cdots$, *n*. The components $\frac{\partial E}{\partial \alpha}$ can be computed as follow

$$\frac{\partial E}{\partial \alpha} = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial E_i}{\partial p_i}\frac{\partial p_i}{\partial f_i}\frac{\partial f_i}{\partial \alpha} \tag{20}$$

and

$$\frac{\partial E_i}{\partial p_i} = -p_i^{1-t_i} t_i (1-p_i)^{t_i - 1} + (1-t_i)(1-p_i)^{t_i} p_i^{-t_i} \tag{21}$$

$$\frac{\partial p_i}{\partial f_i} = -A p_i^2 e^{(Af_i + B)} \tag{22}$$

$$\frac{\partial f_i}{\partial \alpha} = y_i K_\Theta\left(\mathbf{x}_j, \mathbf{x}_i\right), \tag{23}$$

where $y_i = \pm 1$ is the bipolar target of example $\mathbf{x}_i$. Notice that we can include the SVM bias *b* in the vector α as $\alpha = (\alpha_1, \ \alpha_2, \ \cdots, \ \alpha_k, b)$. Then, it can be shown that

$$\frac{\partial \alpha}{\partial \theta_r} = -\mathbf{H}^{-1}\frac{\partial \mathbf{H}}{\partial \theta_r}\alpha^T, \tag{24}$$

where $\mathbf{H} = \begin{pmatrix} \mathbf{K}^Y & \mathbf{Y} \\ \mathbf{Y}^T & 0 \end{pmatrix}$ with the components $\mathbf{K}_{ij}^Y = y_i y_j K\left(\mathbf{x}_i, \mathbf{x}_j\right)$, where vector $\mathbf{Y}$ is the target vector corresponding to the support vectors set, and $\mathbf{Y}^T$ is the transpose matrix of the matrix $\mathbf{Y}$. So we can update the parameters θ_r such that *E* is minimized

by the gradient descent method

$$\Delta\theta_r = -\varepsilon \frac{\partial E(\alpha, \theta_r)}{\partial \theta_r}, \tag{25}$$

where ε is the amplitude of the step along the search direction.

6. **The measurement of Accuracy**

In our HLA classification approach, such confusing conditions will not happen. Therefore, the accuracy measurement in our experiments is quite clear and simple. Let us use a function *A* (accuracy) to indicate the classification correctness of a protein pattern fed into the HLA. Then the total number of correctly classified proteins can be expressed as:

$$\begin{aligned} O &= A(level2 \mid level1) \\ &= A(level2)A(level1 \cap level2) \end{aligned} \tag{26}$$

where *A* is a conditional function whose value is 1 only when a protein pattern is correctly classified by the classifiers in both Level 1 and Level 2 of the HLA, and is 0.

Based on the above concepts, the accuracy measurement of the proposed approach is defined as follows. If the number of testing proteins belonging to the F_i^{th} fold is n_i, but the tested classifier only recognizes o_i proteins as the F_i^{th} fold, then the accuracy rate of this tested classifier is set as $\frac{o_i}{n_i}$ for the F_i^{th} fold. The total classification accuracy can be briefly calculated as follows.

$$N = n_1 + n_2 + n_3 + ... + n_i = \sum_{i=1} n_i \quad \text{(in this case, } i\text{=27, } N\text{=385)} \tag{27}$$

$$O = o_1 + o_2 + o_3 + ... + o_i = \sum_{i=1} o_i \quad \text{(in this case, } i\text{=27)} \tag{28}$$

$$Q = \frac{O}{N} \tag{29}$$

where *N* is the total number of testing proteins data, *O* is the total number of correctly classified proteins in Eq. (26), and *Q* is the classification (prediction) accuracy.

7. Experimental Results

To demonstrate the proposed techniques for multi-class protein fold classification, several experiments are performed and the results are illustrated. The experiment datasets are based on the protein database, SCOP, introduced in above section. The proposed ECOC SVMs with different codeword length are trained and test based on these datasets. All the data are summarized in Table 4 ~ Table 6 and the previous results are also listed in Table7 ~ Table 8.

Table 4. Protein-fold classification accuracy of various single-level classification approaches, where the input PSIs fed into the classifier are C+S+H+P+V+Z.

Classifier / Accuracy	MLP	GRNN	RBFN	SVM
Q(C+S+H+P+V+Z) (%)	48.8	44.2	49.4	51.4

Table 5. Protein-fold classification accuracy comparisons of the proposed HLA and the existing approaches, where "OvO" standing for the one-versus-others method, "uOvO" for the unique one-versus-others method, and "AvA" for the all-versus-all method.

PSI	C (%)	C+S (%)	C+S+H (%)	C+S+H+P (%)	C+S+H+P+V (%)	C+S+H+P+V+Z (%)
OvO (NN)*	20.5	36.8	40.6	41.1	41.2	41.8
OvO (SVM)*	43.5	43.2	45.2	43.2	44.8	44.9
uOvO (SVM)*	49.4	48.6	51.1	49.4	50.9	49.6
AvA (SVM)*	44.9	52.1	56	56.5	55.5	53.9
RBFN (Single-stage)*	40.3	48.6	50.1	52	49.1	49.4
HLA (MLP)	32.7	48.6	47.5	43.2	43.6	44.7
HLA (RBFN)	44.9	53.8	53.3	54.3	55.3	56.4
HLA (GRNN)	-----	----	----	----	----	45.2
HLA (SVM)	-----	----	----	----	----	53.2
HLA (ECOC)	54.81	54.01	55.06	53.25	54.81	56.1

Note: * Data from the paper (Dubchak *et al*., 2001[5]).

** Using RBFN directly to classify the proteins into 27 folds (i.e., single-level approach).

In Table 4, the single-layer approach is used to directly classify each test input vector into 27 folds. It can be seen that the performance of proposed method can have good accuracy when compared to other methods.

Table 6. Classification accuracies of the ECOC-SVM-based HLA with various combinations of global features (C+H+S+P+V+Z) and local features (bi-gram coded feature (B) and spaced bi-gram coded feature (SB)).

ECOC-Based HLA				
Features		Global features(6PSI)	PSIs+B	PSIs+B+SB
No. of Features		125	125+441	125+441+441
Accuracy of Level 1		80.26	83.38	84.94
Accuracy of Level 2	Group 1	70.49	78.69	83.61
	Group2	54.7	66.67	71.79
	Group 3	40.32	50	56.45
	Group 4	38.71	50	56.45
Overall Accuracy(%)		56.1	65.45	68.57

Table 7. Classification accuracies of the RBFN-based HLA with various combinations of global features (C+H+S+P+V+Z) and local features (bi-gram coded feature (B) and spaced bi-gram coded feature (SB)).

RBFN-Based HLA					
Features		Global features (6 PSIs)	Local feature B	PSIs + B	PSIs + B + SB
No. of Features		125	441	125+441	125+441+441
Accuracy of Level 1		81.6	79.2	83.1	83.6
Accuracy of Level 2 (%)	Class 1	67.2	59.0	77.0	73.8
	Class 2	52.1	56.4	62.4	63.2
	Class 3	58.6	60.0	62.8	69.0
	Class 4	48.4	56.5	54.8	53.2
Overall Accuracy (%)		56.4	58.2	63.7	65.5

Table 8. Classification accuracies of the SVM-based HLA with various combinations of global features (C+H+S+P+V+Z) and local features (bi-gram coded feature (B) and spaced bi-gram coded feature (SB)).

SVM-Based HLA					
Features		Global features (6 PSIs)	Local feature B	PSIs + B	PSIs + B + SB
No. of Features		125	441	125+441	125+441+441
Accuracy of Level 1		81.3	77.9	83.4	84.4
Accuracy of Level 2 (%)	Class 1	60.7	57.4	73.8	73.8
	Class 2	49.6	53.8	59.0	60.7
	Class 3	56.6	60.0	64.8	65.5
	Class 4	45.2	59.7	52.6	58.1
Overall Accuracy (%)		53.2	57.7	62.3	64.2

8. Conclusions

In this study, we used algebraic coding theory and gradient descent method to improve the performance of the error-correcting-output-code support vector machine (ECOC SVM) classifier by finding out the better penalty parameter *C* and kernel parameters of SVM. The proposed new multi-class SVM classifier is composed of training phase and testing phases. Several experimental results show the superiority of the proposed scheme over the existing ones. However, there are some factors may be considered to improve the ECOCSVM classifier such as the self-tuning ECOC SVM uses the gradient descent method to tackle the drawback of ECOC SVM and obtain better component binary classifiers; the self-tuning ECOC SVM for multi-class classification resolves the phenomena of unclassifiable regions caused by the one-against-all, one-against-one, or directed acyclic graph SVM methods; and the self-tuning ECOC SVM classifier achieves higher classification accuracy rates than the compared SVM classifiers, and has better generalization ability.

9. References

[1] P. Baldi and S. Brunak, *Bioinformatics: the Machine Learning Approach*, MIT Press, 1998.

[2] C. H. Wu, *Neural networks and genome informatics*, Elsevier, U.K., 2000.

[3] J. Yang, R. Parehk, V. Honavar, and D. Dobbs, "Data driven theory refinement algorithms for bioinformatics," *IJCNN '99 International Joint Conference*, Vol. 6, pp. 4064-4068, July 1999.

[4] C. D. Huang, *Hierarchical Learning Architecture for Multi-class Protein Fold Classification Based on Neural Networks and Support Vector Machine*, Doctoral Dissertation, NCTU, HsinChu, Taiwan, 2004.

[5] I. Dubchak and C. H. Q. Ding, "Multi-class protein fold recognition using support vector machines and neural networks," *Bioinformatics*, Vol. 17, No. 4, pp. 349-358, 2001.

[6] I. Dubchak, I. Muchnik, S. R. Holbrook, and S. H. Kim, "Prediction of protein folding class using global description of amino acid sequence," *Proc. Natl. Acad. Sci.*, USA, Vol. 92, pp. 8700-8704, 1995.

[7] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: A structural classification of proteins database for the investigation of sequence and structures," *Journal of Molecular Biology*, Vol. 247, pp. 536-540, 1995.

[8] I. Dubchak, I. Muchnik, C. Mayor, I. Dralyuk, and S. H. Kim "Recognition of a protein fold in the context of the SCOP classification," *PROTEINS: Structure, Function, and Genetics*, Vol. 35, pp. 401-407, 1999.

[9] L. L. Conte, B. Ailey, T. J. Hubbard, S. E. Brenner, A. G. Murzin, and C. Chothia, "SCOP: a structural classification of proteins database," *Nuclear Acid Research*, Vol. 28, No. 1, pp. 257-259, 2000.

[10] L. L. Conte, S. E. Brenner, T. J. Hubbard, C. Chothia, and A.G. Murzin, “SCOP database in 2002: refinements accommodate structural genomics,” *Nucleic Acids Research*, Vol. 30, No. 1, pp. 264-267, 2002.

[11] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, N.Y., 1995.

[12] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, “Support vector machines,” *Intelligent Systems, IEEE*, Vol. 13, Issue: 4 , pp.18-28, Jul/Aug 1998.

[13] E. Osuna, R. Freund, and F. Girosi, “An improved training algorithm for support vector machines,” *Neural Networks for Signal Processing, Proceedings of the 1997 IEEE Workshok*, pp. 276 –285, Sep. 1997

[14] M. Niranjan, “Support vector machines: a tutorial overview and critical appraisal,” *Applied Statistical Pattern Recognition, IEE Colloquium on*, pp.2/1, 1999

[15] C. J. Lin and C. W. Hsu, “A comparison of methods for multi-class support vector machines,” *IEEE Trans. on Neural Networks*, Vol. 13, pp. 415-425, 2002.

[16] T. Dietterich and G. Bakiri, “Solving multi-class learning problems via error-correcting output codes,” *Journal of Artificial Intelligence Research*, (2): pp. 263-286, 1995.

[17] O. Chapelle and V. Vapnik, “Choosing multiple parameters for support vector machines,” *Advances in Neural Information Processing Systems*, 2001.

[18] C. Cortes and V. Vapnik, “Support vector networks,” *Machine Learning*, Vol. 20, pp. 273-297, 1995.

[19] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods,* Cambridge University Press, 2000.

[20] A. Klautau, N. Jevtic, and A. Orlitsky, “On nearest-neighbor error-correcting output codes with application to all-pairs multi-class support vector machines,” *Journal of Machine Learning Research* 4: pp. 1-15, 2003.