Elsevier Editorial System(tm) for Expert Systems With Applications
Manuscript Draft

Manuscript Number:

Title: Margin-based Ensemble Classifier for Protein Fold Recognition

Article Type: Full Length Article

Keywords: Protein fold recognition; Adaptive local hyperplane; Support vector machine; Ensemble classifier; Amino acid sequence

Corresponding Author: Dr Tao Yang,

Corresponding Author's Institution:

First Author: Tao Yang, PhD

Order of Authors: Tao Yang, PhD; Vojislav Kecman, PhD; Longbing Cao, PhD; Chengqi Zhang, DSc

Margin-based Ensemble Classifier for Protein Fold Recognition

Tao Yang [a]*, Vojislav Kecman [b], Longbing Cao [a] and Chengqi Zhang [a]

[a] Faculty of Engineering and Information Technology, University of Technology Sydney, 15 Broadway, Ultimo, NSW 2007, Australia

[b] Department of Computer Science, Virginia Commonwealth University, 401 West Main, Richmond, VA, USA

Abstract

Recognition of protein folding patterns is an important step in protein structure and function predictions. Traditional sequence similarity-based approach fails to yield convincing predictions when proteins have low sequence identities, while the taxonometric approach is a reliable alternative. From a pattern recognition perspective, protein fold recognition involves a large number of classes with only a small number of training samples, and multiple heterogeneous feature groups derived from different propensities of amino acids. This raises the need for a classification method that is able to handle the data complexity with a high prediction accuracy for practical applications. To this end, a novel ensemble classifier, called MarFold, is proposed in this paper which combines three margin-based classifiers for protein fold recognition.

The effectiveness of our method is demonstrated with the benchmark D-B dataset with 27 classes. The overall prediction accuracy obtained by MarFold is 71.7%, which surpasses the existing fold recognition methods by $3.1 - 15.7\%$. Moreover, one component classifier for MarFold, called ALH, has obtained a prediction accuracy of 65.5%, which is $4.7 - 9.5\%$ higher than the prediction accuracies for the published methods using single classifiers. Additionally, the feature set of pairwise frequency information about the amino acids, which is adopted by MarFold, is found to be important for discriminating folding patterns. These results imply that the MarFold method and its operation engine ALH might become useful vehicles for protein fold recognition, as well as other bioinformatics tasks. The MarFold method and the datasets can be obtained from: (http://www-staff.it.uts.edu.au/∼lbcao/publication/MarFold.7z).

Keywords: Protein fold recognition; Adaptive local hyperplane; Support vector machine; Ensemble classifier; Amino acid sequence

*Corresponding author. Email: tyan028@gmail.com. Tel: +61 2 8095 9258. Postal Address: Room 220, Level 4, Building 10, 235 Jones Street, Broadway, NSW 2007, Australia

1 Introduction

The rapid accumulation of human genomic sequence data underscores the urgent need for effective and efficient computational algorithms to transform the sequence datasets into biological knowledge. The discovery of the three-dimensional (3D) structures of proteins is crucial for understanding and predicting their cellular attributes and many other functional properties. Protein fold recognition is the prediction of a protein's 3D structure based on its amino acid sequence information. Proteins are said to have a common fold if they have the same major secondary structure in the same arrangement and with the same topology, whether or not they have a common evolutionary origin (Craven *et al*., 1995).

The traditional approach for predicting protein folding patterns is based on a comparison between the unknown protein sequence and the known protein sequences located in database by computing sequence similarities. These similarity-based methods can achieve encouraging performance when proteins have close evolutionary relationship (Wang & Dunbrack, 2004; Han *et al*., 2005; Soding, 2005), but they fail to identify reliable homologies when there is less than 20% sequence identity available. In contrast, the *taxonometric* approach, which assumes that the number of protein folds is limited, views the protein fold recognition task as a pattern recognition problem without relying on sequence similarities (Chou & Zhang, 1995). In this approach, the relevant features are firstly extracted from the protein sequences and then a classification method can be used on the obtained features. We focus on this approach only in the present study.

The major challenge in the taxonometric approach lies in the data complexity aspect of the underlying problem, which involves a large number of folding classes with only a small number of training samples and multiple heterogeneous feature groups associated with the folding pattern discovery. For such a reason, the modern pattern recognition methods have achieved only modest levels of performance in protein fold recognition applications. Specifically, past work on the taxonometric approach has employed various classifiers including hidden Markov models (Jaakkola *et al*., 1999), artificial neural networks (ANNs) (Ding & Dubchak, 2001; Chung & Huang, 2003), support vector machines (SVMs) (Ding & Dubchak, 2001), Bayesian networks (Raval *et al*., 2002) and K-local hyperplane distance nearest neighbor (HKNN) (Okun, 2004), and their prediction accuracies can hardly reach 60%, which is still below the accuracy level of practical applications.

To deal with such a formidable difficulty, the ensemble classifiers have been extensively developed for the protein fold recognition problem in the recent past (Tan *et al*., 2003; Nanni, 2005, 2006a,b; Shen & Chou, 2006; Chen and Kurgan, 2007; Chen *et al*., 2008; Damoulas & Girolami, 2008; Ghanty & Pal, 2009). The ensemble classifiers are constructed through combining a number of component classifiers on multiple sets of features or attributes derived from different propensities of the amino acids. The rationale is that the sets of proteins misclassified by the individual classifiers would not necessarily overlap and thus the combination of these classifiers can enhance the overall prediction quality. In our observation, only a few types of classifiers are trained with many feature sets. For example, only the OET-KNN (Optimized Evidence-Theoretic K Nearest Neighbors) classifier is modeled on each of the nine different feature groups in (Shen & Chou, 2006). Hence, the performance of component classifiers can greatly influence the final classification accuracy of the resultant ensemble system.

In this paper, we propose a novel *margin*-based ensemble classifier, called MarFold, for the multi-class protein fold recognition task where multiple heterogeneous feature

spaces are available. The MarFold method is built on three component classifiers, namely, adaptive local hyperplane (ALH) (Yang & Kecman, 2008), SVM and ALHK (a variant of ALH developed in this study). The learning algorithms for all the three classifiers are subject to the maximization of a margin between different classes in the training dataset, though the term 'margin' has different implications for the individual classifiers. Specifically, the SVM method is designed to maximize the *global* margin in the kernel space over the entire training dataset, whereas the ALH method maximizes the *local* margin surrounding the query protein in the weighted feature space. Note that the presentation of ALH in this paper is much more detailed than the brief introduction in (Yang & Kecman, 2008). Also, the significance of the feature weighting scheme and the implication of local margin maximization for ALH are explained in this study. The ALHK classifier is the same as ALH except that it uses a different neighborhood selection procedure, which makes it maximize a margin with both the global and local spirits. With such a combination, we explain why the resultant ensemble system is developed with the properties of *superiority*, *suitability*, *consistency* and *complementarity*.

In our proposed ensemble system, the individual classifiers are trained independently on multiple feature sets for amino acid sequences, which consists of the pseudo amino acid compositions (PseAA) (Chou, 2001) (which includes both the basic amino acid compositions (AAC) and the sequence order effect), predicted secondary structure, hydrophobicity, van der Waals volume, polarity, polarizability, as well as attributes derived from pairwise frequency information about the amino acids. The commonly used majority voting scheme severs as the ensemble strategy for fusing the decisions of multiple constituent experts (classifiers) trained on various feature spaces. We perform experiments to compare our proposed method to many taxonometric fold recognition methods in literature on a widely used D-B dataset. Our proposed ensemble method, MarFold, has achieved an overall accuracy of 71.7%; our proposed component classifier, ALH, has obtained an overall accuracy of 65.5%. The experimental results indicate that both the operation engine and the final ensemble system proposed in this work gives comparable results to the traditional component and ensemble classifiers for protein fold classification, respectively.

2 Materials and Methods

2.1 Dataset

To evaluate the proposed method and compare it with existing methods, the D-B dataset established by Ding & Dubchak (2001) has been used in our experiments. There are 313 protein sequences in the training dataset where two proteins have no more than 35% of the sequence identity for aligned subsequences longer than 80 residues. The test dataset consists of 385 SCOP sequences having less than 40% identity with each other (Andreeva *et al*., 2004). In fact, 90% of the proteins of the testing dataset have less than 25% sequence identity with the proteins of the training dataset. The proteins in both the training and test sets belong to 27 different SCOP folds represented by seven or more proteins and corresponding to four major structural classes: $\alpha, \beta, \alpha/\beta$ and $\alpha + \beta$. The fold names and the number of proteins in each fold of Ding and Dubchak's dataset are described in Table 1.

In this study, we compare various methods by the overall accuracy Q, which is defined as the percentage of correctly recognized proteins to all proteins in the test

Table 1: Summary of the 27 protein folds in the D-B dataset

Fold	Index	N_{training}	N_{test}
α			
Globin-like	1	13	6
Cytochrome c	3	7	9
DNA-binding 3-helical bundle	4	12	20
4-helical up-and-down bundle	7	7	8
4-helical cytokines	9	9	9
Alpha; EF-hand	11	7	9
β			
Immunoglobulin-like β-sandwich	20	30	44
Cupredoxins	23	9	12
Viral coat and capsid proteins	26	16	13
ConA-like lectins/glucanases	30	7	6
SH3-like barrel	31	8	8
OB-fold	32	13	19
Trefoil	33	8	4
Trypsin-like serine proteases	35	9	4
Lipocalins	39	9	7
α/β			
(TIM)-barrel	46	29	48
FAD (also NAD)-binding motif	47	11	12
Flavodoxin-like	48	11	13
NAD(P)-binding Rossmann-fold	51	13	27
P-loop containing nucleotide	54	10	12
Thioredoxin-like	57	9	8
Ribonuclease H-like motif	59	10	14
Hydrolases	62	11	7
Periplasmic binding protein-like	69	11	4
$\alpha+\beta$			
β-grasp	72	7	8
Ferredoxin-like	87	13	27
Small inhibitors, toxins, lectins	110	14	27

dataset. Mathematically, the overall accuracy can be expressed as $Q = c/n$, where c is the number of query proteins whose folds have been correctly recognized and n is the total number of proteins in the test dataset. Also, the prediction accuracies in different folds are also employed to test the consistency and complementarity of MarFold. Suppose there are e_k query proteins correctly recognized as belonging to fold k, then $Q_k = c_k/n_k$, $k = 1, \ldots, 27$.

2.2 Our Approach

Our motivation is to develop an ensemble system by fusing the component classifiers with high levels of superiority, suitability, consistency and complementarity for protein

fold recognition. In short, our proposed ensemble classifier for protein fold recognition is developed under the *SSCC* principle. To this end, we combine two ALH-based (ALH and ALHK) classifiers with the SVM classifier on multiple sets of features derived from the amino acid sequences.

The ALH classifier is a local modeling algorithm which takes full advantage of the training samples surrounding the query by constructing linear manifolds in the weighted feature space. It has been shown that the ALH classifier has superior performance than several traditional classifiers including SVM, KNN (K-nearest neighbors), HKNN, linear discriminant analysis (LDA) and decision tree in a variety of pattern recognition problems (Yang & Kecman, 2008, 2009a,b). The SVM classifier, on the other hand, is by far the most popular pattern classifier and it has been successfully applied in many tasks including problems in Bioinformatics (Ding & Dubchak, 2001; Hua & Sun, 2001). Therefore, the adoptions of the ALH-based and SVM classifiers are based on their performance superiorities for general pattern recognition problems.

It has not escaped from our notice that the 'best' pattern recognition algorithm does not necessarily become the most suitable tool for the problem at hand. The ALH-based classifiers (ALH and ALHK) are employed by MarFold since they are especially appealing to the protein fold recognition problem for the following two reasons. First, the ALH-based methods naturally handle the multi-class problems involved in protein fold recognition without resorting to additional multi-class schemes designed for the binary classifiers, such as the all-versus-all procedure used in (Ding & Dubchak, 2001). This property allows the ALH-based methods not to have performance sacrifice and additional running time when the number of classes increases from two to a large number. Second, the feature weighting scheme embedded in the ALH-based methods can inherently give higher weights to those more informative features such that it can resolve the *curse of dimensionality* (Bellman, 1961) caused by the data sparseness nature in protein fold recognition. In fact, the ratios between the number of training samples and the number of features for most protein fold recognition datasets are quite small and this is why component classifiers need to be trained on a subset of features for performance improvement. In parallel, the SVM classifier is also suitable for the high dimensional and sparse datasets since it can map the original input space into a high dimensional feature space by using the renowned 'kernel trick' (Vapnik, 1998).

Although the ALH-based and SVM methods are developed under different paradigms, they are consistent in the optimization objective of maximizing a margin of separation between different classes. On the other side of a coin, the experts in the final committee can give complementary information since the maximizations of margin are implemented in distinct scenarios. The query-specific nature of ALH can make it give outstanding performance when a unique model over the whole dataset is vague, which can be caused by either the complexity of the concepts underlying the data or the insufficiency of the training dataset regarding the establishment of a global model. On the other hand, the ALH model can be trapped into a local solution while the global insights ignored by ALH can be easily recognized by SVM. The ALHK classifier, compared to ALH and SVM, is a classifier with its nature somewhere in between ALH and SVM.

2.3 Feature extraction

Since the classification methods are blind other than the numeric data, various feature extraction methods have been used to convert the protein sequences into numeric features. The most commonly used feature sets are the AAC (C), predicted secondary

structure (S), hydrophobicity (H), normalized van der Waals volume (V), polarity (P) and polarizability (Z) (Ding & Dubchak, 2001). All but the amino acid composition features have dimensionality 21, whereas the amino acid composition has dimensionality 20.

The AAC features are the fundamental attributes for protein function prediction, as they have also been used for many other proteomics problems such as protein subcellular localization (Hua & Sun, 2001). It has also been shown in (Ding & Dubchak, 2001) that AAC is the most effective feature set for protein fold recognition. However, the sequence order information has been neglected by the AAC features. To resolve this drawback, Shen & Chou (2006) proposed the pseudo-amino acid compositions (PseAA) (Chou, 2001) to represent the protein sequence information in a more effective way. We use the PseAA features (A) as a fundamental feature set and they are retained with all combinations of feature sets used in our ensemble and the value of λ for PseAA was set to 5 in our experiments. The feature set CHVPZ has been considered as a fold recognition feature set in (Ghanty & Pal, 2009) as there is about 30% error in the predicted secondary structure itself. We consider using the feature set AHVPZ, its counterpart AS and the combined set ASHVPZ.

We also consider using the pairwise frequency information about the amino acids as the fold discriminative features. In particular, the pairwise frequencies of amino acids separated by exactly one residue ($PF1$) and the pairwise frequencies of adjacent amino acids ($PF2$) (Ghanty & Pal, 2009) have been augmented to form a feature set, and we term this feature set as PF. In this way, we get feature vectors of dimension 800 for PF. Note that the rationale for using $PF1$ lies in the fact that the adjacent residues can result in a complete collapse of a protein chain (Ghanty & Pal, 2009). In our study, we use the PF feature set and combine it with each of the AHVPZ, AS and ASHVPZ feature sets for protein fold recognition. All the feature sets and their dimensions used in our study are listed in Table 3.

Given the derived feature measurements, the protein fold recognition problem becomes a multi-class and multiple feature sets classification problem. We will briefly introduce three modern pattern classifiers, namely, ALH, SVM and ALHK in the following sub-sections.

2.4 ALH classifier

In multi-class classification, we are given a training dataset of N samples, on which we have D feature measurements: $\mathbf{x}_i = (x_{i1}, \ldots, x_{iD})'$[1] $\in \Re^D$. Suppose there are M classes in both the training and test datasets, then each training samples has a known class label $y_i \in \{1, \ldots, M\}$. The goal of a classification method is to predict the class membership of the query $\mathbf{q} = (q_1, \ldots, q_D)'$, and we denote the predicted class label of the query by $f(\mathbf{q})$.

According to *decision theory*, in order to minimize the misclassification rate, the query $\mathbf{q}$ should be assigned to the class with the largest class posterior probability, that is, $f(\mathbf{q}) = \arg\max_m P(m|\mathbf{q})$. The KNN classification rule is derived from the decision theory given above. However, it has a fundamental assumption, which says the class posterior $P(m|\mathbf{q})$ is locally constant, that is, $P(m|(\mathbf{q} + \delta\mathbf{q})) \simeq P(m|\mathbf{q})$ as $\|\delta\mathbf{q}\| \to 0$. This assumption can be violated for datasets with high dimensionality and sparse samples due to the curse of dimensionality. For this reason, the use of a weighted distance metric to select nearest neighbors has been considered to make the

[1] $'$ represents the transpose of a vector or matrix throughout this paper.

class posteriors locally constant. The weight here refers to the discrimination ability of a feature, that is, the relevance of a feature for classifying the different classes in the dataset. Suppose the feature weight for feature j is denoted by w_j, then the weighted Euclidean distance metric between $\mathbf{x}_i$ and $\mathbf{q}$ can be expressed as:

$$L_w(\mathbf{q}, \mathbf{x}_i) = \sqrt{\sum_{j=1}^{D} w_j (q_j - x_{ij})^2}, \quad i = 1, \ldots, N. \tag{1}$$

In the ALH method, the *ratio of between-group to within-group sum-of-squares* has been considered to measure a feature's discrimination ability and hence its feature weight. In particular, the statistical information about the discrepancies between the class centroids is used to measure the feature's class separability. Suppose there are N_m training samples in class m, such that $\sum_{m=1}^{M} N_m = N$, and let us denote $I(.)$ by the indicator function, whose value is 1 if its argument is true and 0 otherwise. Then the class centroid is $\bar{\mathbf{x}}_m = \frac{1}{N_m} \sum_{i=1}^{N} \sum_{m=1}^{M} I(y_i = m)\, x_{ij}$, and the grand class centroid is $\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} x_{ij}$. Now, the ratio of between-group to within-group sum-of-squares for feature j can be defined as follows:

$$r_j = \frac{\sum_{i=1}^{N} \sum_{m=1}^{M} I(y_i = m)(\bar{x}_{mj} - \bar{x}_j)^2}{\sum_{i=1}^{N} \sum_{m=1}^{M} I(y_i = m)(x_{ij} - \bar{x}_{mj})^2}, \quad j = 1, \ldots, D. \tag{2}$$

The use of Equation (2) for weighing features implicitly assumes that the samples in the same class are scattered around one center, and this unimodal assumption also exists in the classical LDA method.

The feature weights can then be determined by the exponential weighting scheme on the normalized r_j:

$$w_j = \frac{\exp(TR_j)}{\sum_{j=1}^{D} \exp(TR_j)}, \tag{3}$$

$$R_j = r_j / \max(r_j), \quad j = 1, \ldots, D, \tag{4}$$

where T is a non-negative temperature parameter that controls the influence of R_j on w_j. If $T = 0$, then $w_j = 1/D$, $\forall j$, implying all features have equal weights. On the other hand, when T is large, a change in R_j will be exponentially reflected in w_j. The exponential weighting procedure forces $w_j \neq 0$, which prevents neighborhoods from extending infinitely in one direction. We illustrate the dependence of w_j on T in Figure 1.

The weighted distance metric defined in Equations (1), (2), (3) and (4) can be used to find the K nearest neighbors. If we use the majority voting rule to classify the query, then the algorithm is a weighted KNN (WKNN) rule. A two-class classification problem is shown in Figure 1, where the WKNN algorithms with various values of T are applied on the same data. The data points are uniformly distributed in a square. The dataset consist of two features denoted by x_1 and x_2 and two classes denoted by squares and circles. Here, the query is represented by a solid dot located in the center of the figures. The vertical line is drawn to separate the two classes. Thus, the correct classification for the query would be the square class. We set $K = 5$ in all four cases. The selected nearest neighbors are labeled by the '+' sign. In this problem, KNN finds three circle nearest neighbors and two square nearest neighbors around the query, and thus it will assign the query to the wrong (circle) class. However, the KNN

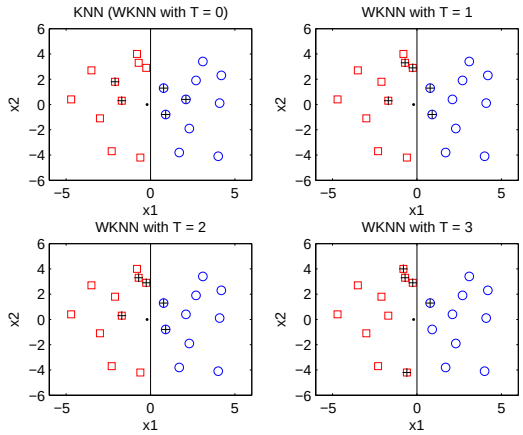

Figure 1: Two-class example by using weighted KNN (WKNN) classification rule with various T

classifier can be improved here by considering the feature weights. Obviously, the two classes can be perfectly separated by considering x_1 only, whereas x_2 can provide no help for classification. All the WKNN algorithms with various T would make the correct decisions for the query. Moreover, the K neighbors show greater variation in x_2 for a larger T. This example shows that the WKNN algorithm implicitly shrinks the neighborhood in directions in which the class centroids differ. In contrast, the KNN algorithm draws a spherical neighborhood around the query.

The decision boundary of SVMs is found by using several training data points (support vectors) only. In contrast, ALH's decision rule is made by using several nearest neighbors as class prototypes only. Thus, how we should utilize these class prototypes in the most efficient way is an important issue. Unlike the KNN or WKNN algorithms, ALH does not classify $\mathbf{q}$ by using the majority voting scheme on its nearest neighbors. Instead, a local linear manifold or hyperplane is constructed for each class to find a virtually enriched training set around $\mathbf{q}$. The local hyperplane for class m around $\mathbf{q}$, $LH_m(\mathbf{q})$, is defined as the hyperplane passing through all the nearest neighbors of $\mathbf{q}$ in class m, and it can be used to fantasize the unseen samples based on a local linear approximation of the manifold of class m.

Suppose $\mathbf{p}_i$ is the ith nearest neighbor (class prototype) in class m, and there are K_m nearest neighbors in class m, such that $\sum_{m=1}^{M} K_m = K$. Then, the local hyperplane for class m can be defined as follows:

$$LH_m(\mathbf{q}) = \{\mathbf{s} \mid \mathbf{s} = \sum_{i=1}^{K_m} \alpha_i \mathbf{p}_i\}, \tag{5}$$

such that

$$\sum_{i=1}^{K_m} \alpha_i = 1. \tag{6}$$

In order to eliminate the constraint in Equation (6), the class centroid of the nearest neighbors around $\mathbf{q}$ ($\bar{\mathbf{p}} = \frac{1}{K_m} \sum_{i=1}^{K_m} \mathbf{p}_i$) can be viewed as the origin or reference point.

Thus, the local hyperplane can also be expressed as:

$$LH_m(\mathbf{q}) = \{\mathbf{s} \mid \mathbf{s} = \sum_{i=1}^{K_m} \alpha_i \mathbf{V}_{.i} + \bar{\mathbf{p}}\}, \tag{7}$$

where $\mathbf{V}$ is the $D \times K_m$ matrix whose ith column is defined as: $\mathbf{V}_{.i} = \mathbf{p}_i - \bar{\mathbf{p}}$. Note that $LH_m(\mathbf{q})$ will define a $K_m - 1$ dimensional hyperplane given that $\mathbf{p}_i$ $(i = 1, \ldots, K_m)$ are independent of one another. If there exist colinearities, it will have a smaller dimensionality.

The undetermined parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{K_m})^T$ are solved by minimizing the (squared) *weighted* distance between $\mathbf{q}$ and $LH_m(\mathbf{q})$ with regularization:

$$\begin{aligned} J_m(\mathbf{q}) &= \min_{\alpha} \Big\{ \big(L_w(\mathbf{q}, LH_m(\mathbf{q}))\big)^2 + \nu \boldsymbol{\alpha}^T \boldsymbol{\alpha} \Big\} && (8) \\ &= \min_{\alpha} \Big\{ \sum_{j=1}^{D} w_j \left(\mathbf{V}_{j.}\boldsymbol{\alpha} + \bar{p}_j - q_j\right)^2 + \nu \boldsymbol{\alpha}^T \boldsymbol{\alpha} \Big\} && (9) \\ &= \min_{\alpha} \Big\{ (\mathbf{s}-\mathbf{q})^T \mathbf{W} (\mathbf{s}-\mathbf{q}) + \nu \boldsymbol{\alpha}^T \boldsymbol{\alpha} \Big\}, && (10) \end{aligned}$$

where $\mathbf{V}_{j.}$ is the jth row of $\mathbf{V}$, $\mathbf{s} \in LH_m(\mathbf{q})$, $\mathbf{W}$ is the diagonal matrix with $W(j,j) = w_j$ and ν is the regularization parameter. It has been proved in (Yang & Kecman, 2008) that the minimization of Equation (10) will be achieved by solving the equation given below for $\boldsymbol{\alpha}$:

$$(\mathbf{U}^T \mathbf{V} + \nu\, \mathbf{I}_{K_m})\boldsymbol{\alpha} = \mathbf{U}^T(\mathbf{q} - \bar{\mathbf{p}}), \tag{11}$$

where $\mathbf{U}^T = \mathbf{V}^T \mathbf{W}$.

Once the minimal $\boldsymbol{\alpha}^*$ is obtained in (11), the corresponding (squared) weighted distance between $\mathbf{q}$ and $LH_m(\mathbf{q})$ can be found as:

$$J_m^*(\mathbf{q}) = \sum_{j=1}^{D} w_j \left(\mathbf{V}_{j.}\boldsymbol{\alpha}^* + \bar{p}_j - q_j\right)^2 + \nu(\boldsymbol{\alpha}^*)^T(\boldsymbol{\alpha}^*). \tag{12}$$

Finally, the class label of the query $\mathbf{q}$ is assigned as:

$$f(\mathbf{q}) = \arg\min_{m}\ J_m^*(\mathbf{q}). \tag{13}$$

The use of local hyperplanes implicitly maximizes the *local margin* surrounding the query. We explain this principle by using an intuitive example illustrated in Figure 2, which shows a two-dimensional example for two-class classification. The feature weights are assumed to be equal for this problem. Consider the spherical neighborhood around the origin, two nearest neighbors (labeled by '+') are selected for each class. A separate dashed line is drawn for each class as the local hyperplane. The local decision boundary that will be produced by the local hyperplanes is represented by the solid line. The ALH algorithm will classify the given point labeled by the black solid as belonging to red class, as it will for all points left of the decision boundary. In this case, we can see that the use of local hyperplanes implicitly maximizes the local margin. Here, the local margin coincides with the global margin found by the SVM classifier. However, the ALH and SVM will draw different decision boundaries for most of the practical problems. For the sake of clarity, the ALH algorithm is summarized in Table 2.

There are three hyper-parameters in the ALH classifier: the number K of nearest neighbors, the temperature parameter T and the regularization parameter ν. It has

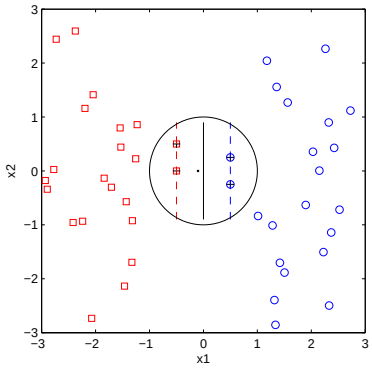

Figure 2: Local maximum margin found by the ALH classifier

Table 2: The ALH classification algorithm

Input: training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, query $\mathbf{q}$, hyper-parameters K, T and ν
1. Weigh the features by using the exponential transformation on the ratio of between-group to within-group sum of squares using Equations (2) - (4).
2. Compute the query - training sample distances in the weighted feature space using Equation (1).
3. Find K nearest neighbors by ordering the distances computed in Step 2.
4. Compute the vector $\boldsymbol{\alpha}$ for each class separately using Equation (11).
5. Compute the query - local hyperplane distances in the weighted feature space using Equation (12).
6. Classify the query using Equation (13).
Output: the class label of the query, $f(\mathbf{q})$

been shown that the performance of ALH is not sensitive to the selection of its hyper-parameters (Yang & Kecman, 2008). Throughout the experiments conducted in this work, we set $\nu = 1$ and the values of K and T were determined via cross-validation over the *training* dataset. In the protein fold recognition problem, the value of K must be large because there are a large number of classes. The initial values for K and T used in ALH are: $K \in \{181, 183, \ldots, 219\}$ and $T \in \{0, 1, \ldots, 5\}$ (a total of 120 combination of hyper-parameter pairs) in our experiments.

2.5 SVM classifier

SVM (Vapnik, 1998) is one of the most popular learning algorithm for pattern recognition applications. SVM is a margin classifier that is designed primarily for two-class classification problems. Intuitively, the objective of SVM is to find a classifier with the largest margin between the samples belonging to two different classes, while minimizing the training error. Here, the principle is that the classifier with the maximum margin is more likely to have a better generalization ability. When the 'kernel trick' is used in SVM, the maximum margin hyperplane is created in the high dimensional feature

space. In the present study, the RBF kernel is used to train the binary SVMs. As with ALH, the hyper-parameters of SVM, which includes the regularization parameter C and spread parameter γ, were determined by cross-validation over the training dataset. We set 120 pairs of initial hyper-parameters with C ranges from 1 to 50 and γ ranges from 5×10^{-5} to 0.5. As a software we used the MATLAB OSU-SVM Toolbox for learning SVM model, which can be obtained from (http://sourceforge.net/projects/svm/).

2.6 ALHK classifier

There are two kinds of K-neighborhood for a given query $\mathbf{q}$: first, the set of K training samples whose distances to $\mathbf{q}$ are smallest; second, the set of K training samples from each class whose distance to $\mathbf{q}$ are smallest. The ALH classifier finds the neighborhood of $\mathbf{q}$ by the first approach, while the ALHK classifier proposed here is the same as ALH except that it finds the K neighbors of $\mathbf{q}$ by the second approach. Accordingly, the local hyperplanes $LH_m(\mathbf{q})$ for ALHK would have $K_m = K, \forall m,$. Therefore, we must have $K_m < \min_m N_m$, where N_m represents the number of training samples in class m, which is seven for the D-B dataset described in Section 2.1. We used $\nu = 0$, but K and T were chosen from a ranges of values through cross-validation over the training dataset. The initial values for K and T for ALHK are: $K \in \{4, 5, 6\}$ and $T \in \{0, 1, \ldots, 5\}$. Here, ALHK only has 18 combination of hyper-parameter pairs, which saves a lot of computational resources compared to ALH and SVM.

There is no explicit answer on which neighborhood selection approach is more rational, but the appropriate choice depends on the specific characteristics of the dataset under consideration. ALH falls strictly into the category of local modeling, while ALHK is closer to a global approach compared to ALH, meaning it is covering all the classes over the whole input space which may not be all relevant to the discrimination of the query.

2.7 Ensemble strategy

In our proposed MarFold method, the three component classifiers (ALH, SVM and ALHK) are trained independently on each of the seven feature sets listed in Table 3, which results in a total of 21 classifiers. The majority voting rule is used to fuse the decisions of the multiple experts. Suppose that the vote v_m for class m $(m = 1, \ldots, M)$ is computed by counting the number of classifiers who has classified $\mathbf{q}$ to class m, the MarFold classifier will classify $\mathbf{q}$ to the class $m^\star$, where $m^\star = \arg\max_m v_m$.

3 Results and Discussions

To demonstrate the effectiveness of our proposed method, we perform experiments on the benchmark D-B dataset. Since the D-B dataset has both the training and test datasets, the tested recognition methods are modeled on the training set only and their recognition accuracies are calculated on the test dataset. In order to provide a fair comparison among the taxonometric protein fold recognition methods, the sequence similarity-based methods are not included in this study (only the method developed by Damoulas & Girolami (2008) uses the similarity features).

The protein fold recognition accuracies for MarFold's three component classifiers (ALH, SVM and ALHK) on seven different feature sets are presented in Table 3. (The bold number represents the best recognition performance for each classifier.) As can

be seen, the ALH classifier achieves the best performance in six out of seven cases, followed by ALHK, which achieves the second best performance in the same six cases. Furthermore, ALH achieves the overall best accuracy (65.5%) and its performance is uniformly good for the various feature sets used.

Table 3: Recognition accuracies ($\%$) for the ALH, SVM and ALHK classifiers on seven feature sets for the D-B dataset

Feature set	Dimension	ALH	SVM	ALHK
AHVPZ	114	56.4	44.7	56.4
AS	51	57.7	40.3	57.1
ASHVPZ	135	61.6	49.4	57.1
PF	800	59.7	**60.8**	55.6
AHVPZ + PF	914	63.9	51.2	59.5
AS + PF	851	64.7	49.4	60.0
ASHVPZ + PF	935	**65.5**	52.7	**61.8**

The best performance of ALH is obtained on the feature set which contains all the possible features in this study (ASHVPZ+PF), which has the largest dimension of 935 among all feature sets. This proves that the ALH classifier is suitable for the high dimensional and sparse dataset for protein fold recognition, where the SVM classifier has been recognized as the benchmark tool. Here, the accuracy for ALH is 12.8% higher than the accuracy for SVM for the (ASHVPZ+PF) feature set. Additionally, the average accuracy of ALH over the nine different feature sets is 11.6% higher than that of SVM, while the average accuracy of ALHK is 8.5% higher than that of SVM.

An interesting phenomenon from Table 3 is that the presence of the PF feature set dramatically improves the recognition accuracies for all of the ALH, SVM and ALHK classifiers. In particular, the SVM classifier has obtained its best accuracy of 60.8% on the PF feature set alone, which is much better than the accuracies obtained on the other feature sets considered in this study as well as the best accuray of 56.0% reported in Ding & Dubchak (2001). Here, the feature set (ASHVPZ + PF) is the most effective set for both ALH and ALHK, though the accuracies for ALH and ALHK on the PF set alone are not outstanding. From these observations, we can see that the pairwise information about amino acids does play an important role for recognizing protein folding patterns.

The comparisons of several traditional protein fold recognition methods using single classifiers are shown in Table 4. (The bold number represents the best recognition performance for the single classifiers.) Ding & Dubchak (2001) tried several versions of the multi-class SVM classifiers and several combinations of amino acid feature spaces, and the best overall accuracy of 56.0% is found by using the SVM classifier with all-versus-all scheme on the CSH feature space. Shamim *et al*. (2007) applied the SVM classifier on the feature sets derived from the information about secondary structural state and solvent accessibility state to further improve the performance to 60.5%. Ghanty & Pal (2009) found that the combination of the SVM classifier and the $PF1$ feature set has the accuracy of 59.1%. In our study, the best performance for the SVM classifier is 60.8%, which is obtained on the PF feature set. The ALH classifier has been combined with the CSHVPZ feature set and it achieved a prediction accuracy of 60.8%. We can see that the ALH classifier combined with the (ASHVPZ + PF) feature set outperforms all the other competing methods, and the recognition accuracy for

Table 4: Recognition accuracy comparisons among the single classifiers on the D-B dataset

Classifier	$Q(\%)$	Source
SVM	56.0	Ding & Dubchak (2001)
RBFN (radial basis function network)	56.4	Huang *et al*. (2003)
HKNN	57.1	Okun (2004)
SVM	60.5	Shamim *et al*. (2007)
MLP (multi-layer perceptron)	57.1	Ghanty & Pal (2009)
RBFN	56.4	Ghanty & Pal (2009)
SVM	59.2	Ghanty & Pal (2009)
ALH	60.8	Kecman & Yang (2009)
SVM	60.8	This paper
ALHK	61.8	This paper
ALH	$\mathbf{65.5}$	This paper

ALH surpasses the existing methods using single classifiers by $4.7 - 9.5\%$. Also, the ALHK classifier combined with the feature set of (ASHVPZ + PF) also outperforms the prior methods by $1.0 - 4.2\%$. Here, the combinations of the ALH classifier with all of the (AHVPZ + PF), (AS + PF) and (ASHVPZ + PF) feature spaces outperform all the methods combining the SVM classifier with various feature spaces.

Table 5: Recognition accuracy comparisons among the ensemble classifiers on the D-B dataset

Classifier	$Q(\%)$	Source
Ensemble of HKNN and LDA	59.2	Nanni (2005)
Ensemble of HKNN and LDA	60.3	Nanni (2006a)
Ensemble of HKNN	61.1	Nanni (2006b)
Ensemble of OET-KNN	62.1	Shen & Chou (2006)
Ensemble of six classifiers	68.4	Chen and Kurgan (2007)
Ensemble of KNN, CNN and PNN[2]	63.1	Chen *et al*. (2008)
Bayesian hierarchical ensemble	68.1	Damoulas & Girolami (2008)
Hierarchical ensemble of GAET-KNN[3]	64.5	Guo & Gao (2008)
Ensemble of SVM, MLP and RBFN	68.6	Ghanty & Pal (2009)
MarFold	$\mathbf{71.7}$	This paper

Table 5 compares the performance of MarFold to nine traditional fold recognition methods using ensemble classifiers. (The bold number represents the best recognition performance for the ensemble classifiers.) It can be seen that the MarFold method achieves the best recognition accuracy of 71.7%, which is $3.1 - 15.7\%$ higher than the prior methods. When we compare MarFold to the best performing competing method by Ghanty & Pal (2009), predicition by MarFold results in a $3.1/31.4 = 10\%$ error rate reduction. We also perform experiments to see the performance of MarFold and ALH with respect to the values of λ in PseAA, and we found that the performance of MarFold is very robust to the selection of λ. In particular, the prediction accuracies of MarFold and ALH for $\lambda = 14$ are 71.4% and 66.5%, respectively.

Table 6 presents the prediction accuracies in the 27 fold for the ALH, SVM, ALHK

and MarFold classifiers. It can be seen that the MarFold method performs equal or better than all the three component classifiers in 22 out of 27 folds (shown in bold), and it performs better than at least one component classifier in all folds. This clearly shows the consistency of the proposed ensemble classifier and the complementarity of its component classifiers.

We also notice that MarFold can recognize three protein folds with 100% accuracy: fold 3 (Cytochrome c), fold 9 (4-helical cytokines), fold 110 (Small inhibitors, toxins, lectins); while some folds are predicted by MarFold with low accuracy: fold 23 (Cupredoxins), fold 32 (OB-fold), fold 69 (Periplasmic binding protein-like). It may be worthwhile to further investigate the biological characteristics of these folds for performance improvement.

Table 6: Comparison of fold accuracies (%) for ALH, SVM, ALHK and MarFold

Fold	ALH	SVM	ALHK	MarFold
1	**83.3**	**83.3**	**83.3**	**83.3**
3	**100.0**	**66.7**	**100.0**	**100.0**
4	**45.0**	**60.0**	**50.0**	**70.0**
7	**62.5**	**50.0**	**87.5**	**87.5**
9	**100.0**	**88.9**	**77.8**	**100.0**
11	**55.6**	**44.4**	**55.6**	**55.6**
20	**90.9**	**86.4**	**75.0**	**95.5**
23	33.3	16.7	50.0	25.0
26	**69.2**	**69.2**	**61.5**	**76.9**
30	**50.0**	**50.0**	**50.0**	**50.0**
31	**75.0**	**37.5**	**75.0**	**75.0**
32	36.8	42.1	42.1	36.8
33	**75.0**	**75.0**	**75.0**	**75.0**
35	**50.0**	**50.0**	**50.0**	**50.0**
39	**71.4**	**71.4**	**57.1**	**71.4**
46	**72.9**	**87.5**	**45.8**	**87.5**
47	**66.7**	**66.7**	**75.0**	**83.3**
48	**46.2**	**38.5**	**53.8**	**61.5**
51	**51.9**	**51.9**	**48.1**	**55.6**
54	41.7	25.0	58.3	50.0
57	**50.0**	**37.5**	**62.5**	**75.0**
59	**57.1**	**42.9**	**57.1**	**64.3**
62	**57.1**	**42.9**	**57.1**	**71.4**
69	25.0	25.0	50.0	25.0
72	**25.0**	**25.0**	**25.0**	**25.0**
87	63.0	29.6	59.3	55.6
110	**100.0**	**100.0**	**100.0**	**100.0**
Overall	65.5	60.8	61.8	71.7

4 Conclusions

We have presented a novel ensemble classifier, called MarFold, for taxonometric protein fold recognition. Our developed method is constructed through fusing the ALH, SVM and ALHK classifiers in seven different feature groups derived from the amino acid sequences. The three component classifiers are all subject to the margin maximization criterion, but their specific mechanisms are complementary to one another. In particular, the naturality to the multi-class problems and inherent feature weighting capabilities allow ALH and ALHK especially appealing to the protein fold classification tasks.

Experimental results on the benchmark D-B dataset show that MarFold achieves a prediction accuracy of 71.7, which is $3.1-15.7\%$ higher than the accuracies for the existing methods. At the same time, its component classifier ALH obtained an accuracy of 65.5%, which is $4.7-9.5\%$ higher than the accuracies for the prior methods using single classifiers. Furthermore, the MarFold method performs equal or better than all of its constitutent classifiers in 22 out of 27 folds, which shows its consistency and the complementarity of its component classifiers. Another major finding in our work is that the pairwise frequency features of the amino acids are important for discriminating folding patterns. Our study demonstates that both the MarFold classifier and the individual expert ALH might become useful vehicles for protein fold recognition, as well as other bioinformatics tasks.

References

Andreeva, A. *et al*. (2004). Scop database in 2004: refinements integrate structure and sequence family data. *Nucleic Acids Res.*, *32*, 226–229.

Bellman, R.E. (1961) *Adaptive Control Processes*. Princeton Univ. Press., Princeton, NJ.

Chen, Y. *et al*. (2008). Ensemble voting systems for multiclass protein fold recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, *22*, 747–763.

Chen, K. & Kurgan, L. (2007). PFRES: protein fold classification by using evolutionary information and predicted secondary structure. *Bioinformatics*, *23*, 2843–2850.

Chou, K.C. (2001). Prediction of protein cellular attributes using pseudo amino acid composition. *Proteins*, *43*, 246–255 (Erratum: ibid., 2001, Vol.44, 60).

Chou, K.C. & Zhang, C.T. (1995). Review: prediction of protein structural classes. *Crit. Rev. Biochem. Mol. Biol.*, *30*, 275–349.

Chung, I.F. & Huang, C.D. (2003). Recognition of structure classification of protein folding by NN and SVM hierarchical learning architecture. *Lecture Notes in Computer Sciences*. Springer, Istanbul, Turkey, Vol. 2714, pp. 1159–1167.

Craven,M.W. *et al*. (1995). Predicting protein folding classes without overly relying on homology. *ISMB*, *3*, 98–106.

Damoulas, T. & Girolami, M.A. (2008). Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection. *Bioinformatics*, *24*, 1264–1270.

Ding, C. & Dubchak, I. (2001). Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, **17**, 349–358.

Ghanty, P. & Pal, N.R. (2009). Prediction of protein folds: extraction of new features, dimensionality reduction, and fusion of heterogeneous classifiers. *IEEE Trans. NanoBioscience*, **8**, 100–110.

Guo, X. & Gao, X. (2008). A novel hierarchical ensemble classifier for protein fold recognition. *Protein Eng. Des. Sel.*, *21*, 659–664.

Han S. *et al*. (2005). Fold recognition by combining profile-profile alignment and support vector machine. *Bioinformatics*, *21*, 2667–2673.

Hua, S. & Sun, Z. (2001). Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, *17*, 721–728.

Huang, C.D. *et al*. (2003). Hierarchical learning architecture with automatic feature selection for multiclass protein fold classification. *IEEE Trans. NanoBioscience*, **4**, 221–232.

Jaakkola, T. *et al*. (1999). Using the Fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Inteligent Systems in Molecular Biology*. AAAI Press.

Kecman V., Yang T.. (2009). Protein fold recognition with adaptive local hyperplane algorithm. In *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, Nashville, TN, USA, pp. 75–78.

Nanni, L. (2005). Fusion of classifiers for protein fold recognition. *Neurocomputing*, *68*, 315–321.

Nanni, L. (2006a). Ensemble of classifiers for protein fold recognition. *Neurocomputing*, *69*, 850–853.

Nanni, L. (2006b). A novel ensemble of classifiers for protein fold recognition. *Neurocomputing*, *69*, 2434–2437.

Okun, O. Protein fold recognition with K-local hyperplane distance nearest neighbor algorithm. In *Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics*, Pisa, Italy, pp. 51–57.

Raval, A. *et al*. (2002). A bayesian network model for protein fold and remote homologue recognition. *Bioinformatics*, *18*, 788–801.

Shamim, M.T. *et al*. (2007). Support vector machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs. *Bioinformatics*, *23*, 3320–3327.

Shen, H.B. & Chou, K.C. (2006). Ensemble classifier for protein fold pattern recognition. *Bioinformatics*, *22*, 1717–1722.

Soding, J. (2005). Protein homology detection by HMM-HMM comparison. *Bioinformatics*, *21*, 951960.

Tan, A.C. *et al*. (2003). Multi-class protein fold classification using a new ensemble machine learning approach. *Genome Inform.*, *16*, 206217.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley, New York.

Wang, G. & Dunbrack, R.L. (2004). Scoring profile–to–profile sequence alignments. *Protein Sci.*, *13*, 1612–1626.

Yang, T. & Kecman, V. (2008). Adaptive local hyperplane classification. *Neurocomputing*, *71*, 3001–3004.

Yang, T. & Kecman, V. (2009a). Face recognition with adaptive local hyperplane algorithm. *Pattern Analysis & Applications*, Theoretical Advances, Springer-Verlag, in press.

Yang, T. & Kecman, V. (2009b). A novel algorithm for learning small medical dataset. *Expert Systems*, *26*, 355–359.