APIS

www.apis.org

ISSN 1976-7587

Proceedings of the 8th APIS

The 8th International Conference on Applications and Principles of Information Science

Jan. 11-12, 2009

University of the Ryukyus · Okinawa, Japan

Hosted by

Ryukyus University

Sponsored by

Korea Information Processing Society

Co-Sponsored by

SK C&C

Protein Fold Recognition Using Bagging Ensemble of SVM Classifiers

Mahdi Pakdaman Naeini*, Behzad Moshiri**, Kaveh Kavousi***
*Department of Computer Engineering and IT, Islamic Azad University Parand Branch, Tehran, Iran**
*Senior member, IEEE, Control & Intelligent Processing, Center of Excellence. School of ECE, University of Tehran, Tehran, Iran***
*Control & Intelligent Processing, Center of Excellence. School of ECE, University of Tehran, Tehran, Iran****
m.pakdaman@ece.ut.ac.ir;moshiri@ut.ac.ir **;kkavousi@ut.ac.ir****

Abstract

The process of protein fold recognition is a challenging 35 year old problem. Recently, this problem attracted a great deal of interest for many computer scientists to work on the prediction of protein folds using various machine learning methods. However, due to the challenging high-dimension multi-class nature of the problem, the common classification methods do not work very well. This paper presents a bagging classifier ensemble of Gaussian kernel SVMs for predicting the fold of a protein using its composition data. The obtained results in our experiments imply that SVM classifiers hold better Correct Classification Rate (CCR) in comparison to other common classification methods such as MLP and RBF Neural Networks. Moreover, by using classifier ensemble method such as bagging with majority voting mechanism we can improve the generalization power of the SVM classifier for 2%.

Key words: *Protein Fold classification, Bagging, classifier Ensemble, SVM, Neural Networks*

1. Introduction

The protein folding is the process by which the protein assumes its characteristic 3D structure after the translation process in a cell. The amino acid sequence of a protein determines its 3D folds, and this in turn predicts the protein function. This process is pretty deterministic since the amino acid sequence determines the fold and the protein fold determines the function. Therefore, this process is theoretically a predictable deterministic process. However prediction of protein fold using its amino acid sequence is impractical since it is too time consuming and also does not yield good prediction results. As a result, this problem has been a 35 year old challenge for biologist. Recently, by the vast improvements in computers' power, computer scientists become interested in the protein folding problem using the machine learning methods. However the ordinary and common classification methods do not work very well on this problem due to its multi-class and high-dimension challenges.

In this paper, we apply some classification methods such as MLP and RBF networks also the new popular one the SVM classifier on the protein folding problem. Moreover, we use the classifier ensemble method, bagging, to improve the prediction results of SVM classifier. Our results show that the classifier ensemble works better than the other methods, which indeed justifies no free lunch theorem.

In the following in section 2 we describe briefly the classification methods used in our study. Section 3 describes the experimental results and section 4 discusses about the reason of improvement in our result using bagging ensemble method. Finally we bring the conclusion and future works of our study.

2. Classification Methods

The inductive learning methods can be categorized as supervised and unsupervised methods. In this section we introduce a brief introduction to the supervised classification methods used in this study; more details can be found in [13].

2.1 Artificial Neural Networks

The neural network is a very applicable regression and classification tool which has the capability of representing complex relationships among inputs and outputs of a system. Intuitively, neural networks imitate the human brain intelligent behavior using a connectionism approach. In this

approach, neural network's knowledge is stored within inter-neuron connection waits known as synaptic weights. The important advantage of neural networks lies in their ability to be a general function approximator and learn both the linear and non-linear relationships directly from the data. The most common neural network models are the Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) which are used in this paper [3, 9, 12, 13].

2.2 Support Vector Machines (SVM)

In this section we are going to establish a brief background on the theory of SVM. Given a linearly separable set of points $D = \{(x_i, c_i) | x_i \in R^n, c_i \in \{-1,1\}\}_{i=1}^N$, the optimal separating hyper plane, the hyper plane with the largest margin, can be obtained by solving the following optimization problem:

$$\text{Minimizing: } \frac{1}{2} w.w$$

$$\text{Subject to: } c_i(w.x_i + b) \geq 1, i = 1,2,...,N$$

Practically, the margin can be thought of as a measure of robustness of the solution for the classification task. If the set D is not linear separable then the above optimization problem has no solution. In this case we use the idea of soft margin method for the classification purpose. This method choose a hyper plane that splits the examples as cleanly as possible, while still maximizing the distance to the closest cleanly split examples. Theses data points are called support vectors. Actually, these points are sufficient to determine the best classifying hyper plane.

The soft margin method introduces slack variables, ξi, which measure the degree of misclassification of the data point x_i

$$c_i(w.x_i + b) \geq 1 - \xi_i, i = 1,2,...,N$$

The objective function is then increased by a function which penalizes non-zero ξi, and the optimization becomes a trade off between a large margin, and a small error penalty. If the penalty function is linear, then the optimal solution can be obtained by solving the following optimization problem:

$$\text{Minimize: } -\frac{1}{2} w.w + C\sum_{i=1}^{N} \xi_i$$

$$\text{Subject to: } \begin{array}{ll} c_i(wx_i + b) \geq 1 - \xi_i & \\ \xi_i \geq 0 & i = 1,2,...,N \end{array}$$

The Idea of SVMs can be generalized simply to the non-linear discriminative classifier by mapping the input vector into a high dimensional feature space using the trick of kernel functions which is an inner product in the new space. Typical kernel functions are polynomial kernels, radial basis kernels and wavelet kernels [6, 8].

3. Purposed Method and Experimental Results

This section describes the experimental results of predicting the protein fold using just protein composition data set of Chris H.Q. The composition data of a protein consist of the percentages of each amino acid in the protein amino acid sequence. So in this view the input of the model is a 20 dimensional feature vector. In this data set two proteins have no more than 35% of sequence identity for the aligned subsequences longest than 80 residues. Also there are 27 different protein fold in this data set in which each fold have at least seven proteins [2, 4].

Since the problem of classification in this case study is a multi class prediction problem we used 27 different output units in the structure of RBF and MLP networks. Also in our study the MLP network has just one hidden layer. For recognizing the exact class of a protein we use the label of the maximum output unit in the network as the protein class label. Moreover, in the protein fold recognition using the SVM classifier with Gaussian kernel we utilize One-Versus-Others method to predict the final class of protein folds.

For finding the optimum number of neurons in MLP and RBF networks we use the CCR on the validation data set. It means the point in which the neural network has minimum error on validation data set is selected for the optimum number of neurons. Also we use Bagging method to ensemble the SVM classifiers in this paper. In this method, given the training data set P of size n, K new training sets $S_1, S_2, ..., S_k$ are produced with size n by randomly selecting the elements of the original training data set, where the same entity may be selected multiple times. It can be shown that if the probability of selecting entries is equally distributed over all training data we can except that around 63.2% of all training elements are included in each new created training data set S_i on average. Each of new set S_i is used to train one Gaussian kernel SVM classifier then an ensemble of k classifier is obtained using majority voting method [1, 11].

Before using protein composition data we normalized them into the interval [1,-1]. The results by 5-fold cross validation are shown in this section. Generally, in the k-fold cross validation the whole data set is partitioned into k subsets. At each time, one of the partition is used as a test data set and the other k-1 partitions are put together to form the whole training data set. At the end the average error

over all k trails is computed. Also in each of k (k=5 in this study) iteration 20% of training data is used as validation data set. Figure 3 and 4 show the CCR of the examined methods on training data set and test data set respectively. From these figures it is clear that the generalization accuracy of SVM classifier is higher respect to MLP and RBF networks. Also, the results show that using bagging ensemble is remarkably better than using single MLP, RBF or SVM classifiers.

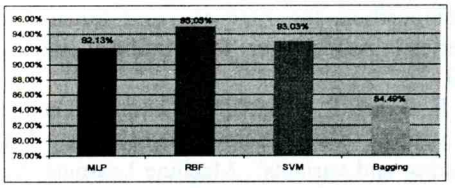

Figure 1- Correct Classification Rate of different methods on training data

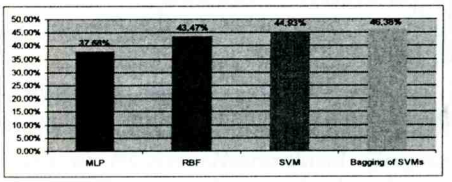

Figure 2- Correct Classification Rate of different methods on test data

4. Discussion

In this section we justify the obtained improvements in protein fold recognition using the bagging ensemble method by describing the bias and variance error of a classifier. This is one of the basic concepts in learning theory which can also justifies the cause of beating a simple learner against a complex one. Moreover, it can be used to prevent a learner being over fitted to the training data.

In a learning problem assume that $S = \{(x_1,t_1),...,(x_n,t_n)\}$ is the set of all training data and the learner L is going to learn the concept y = f(x). In order to evaluate the quality of the learner we usually use a Loss function that can be declared in different ways such as: Zero-One loss, Squared Loss or Absolute Loss function. Generally, a loss function can be written as the summation of bias, variance and the noise [10, 13].

Let assume that we want to learn a quadratic function using a linear learner L and we have many different training sets of the target function. By using different training set of the quadratic function we will obtain different lines e.g. different least squared lines and if we average these lines together we will obtain another line. Since our target function is a quadratic one and we want to learn it by a linear hypothesis, the value of estimated function with the average line for every point will have some error with the true target value of that point. We call this type of estimate error as bias error of our learner.

Bias error of learner exist since hypothesizes do not have the ability of showing entirely the true target concept. The bias error in the linear regression of a quadratic function is shown in Figure1.

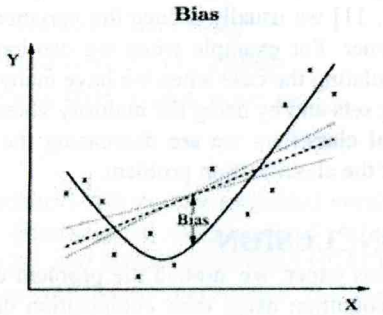

Figure 3- bias error of learning a quadratic function using a linear learner

Consequently, if our learner is a general function approximator by averaging out the result of learning over all different training data we can somewhat cancel out the bias error of our learner. However, the loss function would not be zero because of the variance error of the learner. In Figure 2 we assume that the points labeled by x, o, s are three different training sets of the true quadratic target function showed by the solid black curve. Since we have some noise in our measurement they are not exactly on the curve.

Suppose for each training set we use any general function approximator and the learning process gives us three different dashed curves completely learned on training data. The variance error for each learner on every point is the difference between the estimated target value at that point and the average of the estimates over all learners. This concept is showed in the Figure2.

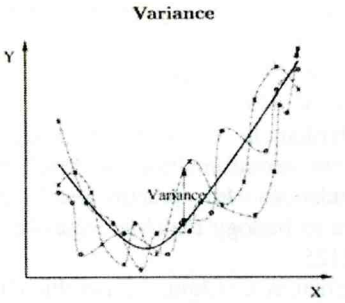

Figure 4- variance error of learning a quadratic function using a general function approximator

We cannot decrease the variance and bias error of a classifier together and there is an optimum point

in between. For example although the bias error of a MLP is less than the bias of a single perceptron, there are some cases in which the variance error of single perceptron beats MLP and sometimes we see that a single perceptron can beat a complex MLP [10, 13].

The interesting point is that when we use classifier ensemble methods such as bagging, boosting or any other classifier ensemble methods [1, 5, 7, 11] we usually reduce the variance error of our learner. For example when we use bagging we are simulating the case when we have many different training sets and by using the majority votes between the final classifiers we are decreasing the variance error of the classification problem.

5. CONCLUSION

In this paper, we studied the problem of protein fold recognition using their composition data in the context of large number of folds. For this purpose, we applied different current classification methods. The results show that the SVM classification method is very promising in the generalization error respect to other classification methods such as RBF and MLP networks. Besides, the experimental results show that by using bagging method on SVM classifiers we can improve the final correct classification rate on the test data around 2%.

In our future work, we are going to apply other classifier fusion methods such as OWA and fuzzy integral to improve the CCR of test data. Also by utilizing other existing features of proteins and using feature transformation techniques such as PCA and ICA, it seems that we can improve CCR of test data much more in our experiences.

References

[1] L. Nanni, A. Lumini, "Ensemblator: An ensemble of classifiers for reliable classification of biological data" journal of Pattern Recognition Letters, Vol 28, 2007, P. 622-630

[2] C. Ding, L. Dubchak, "Multi-class protein fold recognition using support verctor machines and neural networks" Journal of Bioinformatics, vol 17, no. 4, 2001, P. 349-358

[3] H. Bhaskar, D. C. Hoyle, A. Singh, "Machine Learning in bioinformatics: A brief survey and recommendations for practitioners". Journal of computers in biology and Medicine-Vol. 36, 2006, P. 1104-1125

[4] H.B. Shen, K.C. Chou,, "Ensemble Classifier for protein fold pattern recognition" Journal of Bioinformatics, vol 22, no. 14, 2006, P. 1717-1722

[5] S. Cho, J. Ryu , "Classifying Gene Expression Data of Cancer using Classifier Ensemble with Mutually Exclusive Feature" IEEE Proceeding Vol. 90, No. 11, November 2002

[6] W. S. Noble, "Support vector machine applications in computational biology" 2003

[7] L. Breiman, 1996 "Bagging Predictors", Machine Learning 123-140

[8] N. Cristianini, J. Shawe-Taylor, "An introduction to support vector machines and other kernel-based learning methods" Cambridge university press, 2000.

[9] P. Baldi, S. Brunak, 2001, "Bioinformatics: The Machine Learning Approach" adaptive computation and machine learning, second ed. MIT press.

[10] P. Domingos, "A Unified Bias-Variance Decomposition for Zero-one and Squared Loss", American Association for Artificial intelligence, 2000

[11] E. Bauer, R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, Boosting and variants", Machine Learning, Vol. 36, P. 105-142, 1999

[12] G. W. Greenwood, J. Shin, B. Lee, G. B. Fogel, "A Survey of Recent Works on
Evolutionary Approaches to the Protein Folding", IEEE, 1999, P. 488-495

[13] C. M. Bishop, "Pattern Recognition and Machine Learning", second edition: Springer 2006