Using Bagging Classifier to Predict Protein Domain Structural Class

http://www.jbsdonline.com

Abstract

Classification and prediction of protein domain structural class is one of the important topics in the molecular biology. We introduce the Bagging (Bootstrap aggregating), one of the bootstrap methods, for classifying and predicting protein structural classes. By a bootstrap aggregating procedure, the Bagging can improve a weak classifier, for instance the random tree method, to a significant step towards optimality. In this research, it is demonstrated that the Bagging performed at least as well as LogitBoost and Support vector machines in predicting the structural classes for a given protein domain dataset by 10 cross-validation test, which indicate that the Bagging method is promising and anticipated that it could be potentially further improved on predicting protein structural classes as well as other biomacromolecular attributes, if the bagging method and other existing methods can be effectively complemented with each other.

Key words: Protein structure classification; Bagging; Support vector machines; LogitBoost; and Amino acid composition.

Introduction

Classification and prediction of protein structure is one of the important topics in the protein science. According to Nikashima work (1), it implied that it is correlated between protein structural information and amino acid composition. Klein (2) and Chou (3) also showed that by the amino acid composition, scientists could discover some folding patterns information of protein structure, while avoiding to the complicated and irregular three-dimensional structure. Usually Proteins are classified into four folding classes is all-\(\alpha\), all-\(\beta\), \(\alpha/\beta\), and \(\alpha+\beta\). The \(\alpha\), \(\beta\) folding information of protein secondary structure becomes important for characterizing the overall structure of a protein or its domain. A lot of effort is made to predict the protein structural classes based on the amino acid composition, [see, e.g., (4-7)]. Recently with the developing of computer science, mathematics and statistics, novel methods, such as support vector machines (SVMs) and LogitBoost classifiers, were introduced to this field (8, 9).

In this paper, we would like to present a approach named “Bagging” (10), which is the abbreviation of “Bootstrap aggregating”, to predict the protein structural classes. We will give a detailed description of Bagging in the following.

How Bagging Works

The Bagging predictor was firstly introduced by Breiman (1996), which is similar to the boosting algorithm. It is a method for using independent bootstrap procedure on the replicate learning set and then to get an aggregated predictor with plurality voting. Tests and theoretical analysis of Breiman’s work shows that the Bagging

Liuhuan Dong
Yuan Yuan
Yudong Cai

1Dept of Combinatorics and Geometry
CAS-MPG Partner Institute for Computational Biology
Shanghai Institutes for Biological Sciences
Chinese Academy of Sciences
China 200031

2Department of Statistics
Jiaxing College
Zhejiang, China 314000
predictors can push an unstable weak classifier to a significant step towards optimality (10). The bagging method has been used to various classification and prediction problems in biology and social sciences [see, e.g., (11-14)].

For Learning set:

\[ L := \{(y_n, x_n), n = 1, 2, \ldots, N\}, \]  
where \( x_n \in X \) is the input and \( y_n \in Y \) is the response from a predictor denoted by \( \phi(X,L) \). When consider \( k \) learning sets \( \{L_k\} \) of \( N \) iid (independent identical distribution) observation from the same distribution of \( L \), it is obvious to use the average to get a better estimate of the classification:

\[ \phi_{A}(X) = E_{L} \phi(X,L), \]  
where the \( E_L \) denotes the expectation over \( L \). Unfortunately, we usually only have one learning set \( L \), so, one resolvable way is to take bootstrap samples \( \{L_{(B)}\} \) of \( N_{(B)} \) cases, randomly, but with replacement from \( L \), and form predictor set \( \{\phi(X, L_{(B)})\} \). Then the predictor \( \bar{\phi} \), which get most voting score of test set is the best classification. The algorithm of Bagging method is showed below Table I:

<table>
<thead>
<tr>
<th>Table I</th>
<th>Bagging algorithm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>For ( k = 1, 2, \ldots, K ) Do</td>
<td></td>
</tr>
<tr>
<td>Draw ( N_{(B)} ) samples randomly and with replacement from the ( X ).</td>
<td></td>
</tr>
<tr>
<td>And form the learning set ( L_{(B)} ).</td>
<td></td>
</tr>
<tr>
<td>Train classification ( \phi_k(L_{(B)}) ) from the learning set ( L_{(B)} ).</td>
<td></td>
</tr>
<tr>
<td>Make a plurality vote for every ( \phi_k ) of ( K ) classifications on the test set.</td>
<td></td>
</tr>
<tr>
<td>The classification, which gets the most voting score, is the best one.</td>
<td></td>
</tr>
</tbody>
</table>

Choice of Weak Classifier

The program is downloaded from WEKA (16). The random tree algorithm for constructing a tree that considers one randomly chosen attribute at each node is used in the Bagging procedure.

Implementation and Dataset

The working dataset was taken from (17) that contains 204 protein chains. Fifty-two of the 204 are all-\( \alpha \) protein domain, 61 are all-\( \beta \), 45 are \( \alpha/\beta \), and 46 are \( \alpha + \beta \). The average sequence similarity is 21% for all-\( \alpha \), 30% for all-\( \beta \), 15% for \( \alpha/\beta \), and 14% for \( \alpha + \beta \). Therefore, there is no similarity for each amino acid sequence within the same class. Twenty amino acid compositions of these 204 protein chains are, thus, taken as the input feature of 20-dimension vectors.

Results and Discussion

The demonstrations are conducted by two different approaches, the re-substitution test and the 10 cross-validation test, with comparing against other two popular methods: SVMs and LogitBoost. The results show that the Bagging predictor works at least as well as the two others.

Re-substitution and 10 Cross-validation Test

All the methods (SVMs, LogitBoost, Bagging) get 100% accuracy on the re-substitution test, which indicates that these methods grasp the complicated relationship between the amino acid composition and the structural class, and reflects good self-consistency of these predictors. However, a cross-validation test is needed to really reflect the power of the predictor. Leave-one-out test usually works well
for estimating generalization error for continuous error functions, but it probably perform poorly for discontinuous data set such as the number of misclassified cases (18), like 204 protein structural dataset in this paper. Breiman (19) and Kohavi (20) found that 10 cross-validation test works better than leave-one-out test on regression and decision trees. Since the target of our work is to obtain as few as misclassified samples by the Bagging approach with random feature decision tree method mentioned above, the 10 cross-validation is preferred in our test.

**Comparison with SVMs and LogitBoost**

The classifier based on SVMs (21-23) and LogitBoost (24) are applied to the same problem as a comparison. The main advantage of SVMs classifier is less overfitting. It tries to find out a hyperplane, by which it can optimally separate the data into classes on the term of maximizing the distance between the closest vectors to the hyperplane. This is because right now, people just can believe the results of linear regression. So the separating hyperplane can be trusted. However, it is quite difficult to decide the sophisticated parameters. In this paper the polynomial kernel is chosen. The complexity parameter C and the exponent of the polynomial kernel are adapted to 10.0 and 3.0. The LogitBoost is based on the boosting method, which, like Bagging, can also improve an unstable and weak classifier into a better one. It iteratively makes the wrong classified sample more weighted and concentrated to improve the classification accuracy. Because it focuses on adjusting the weights for the wrong classified samples, it may lose some accuracy on classifying other samples. For some datasets, it will also cause overfitting problem. While for the method of Bagging, as introduced above: it focus on the global accuracy, that is why it can obtain less overfitting and try to get the averagely classification results. For the LogitBoost and Bagging algorithm, some parameter setting is adapted empirically. We try two classifiers: stumps (default) and M5 trees (25) for LogitBoost to obtain a better performance, it seems that M5 trees turn out to be much better than stumps, because they are able to generate several unconnected regions for a category. For the same reason, the number of iterations of both LogitBoost is set to be 100. With the setting of 99% out-of-bag error being calculated, random trees for classifier and 96 times iterations, the Bagging obtain a local optimal result.

All the success rates are showed in Table II. On the term of overall success rate, SVMs is a little better than the Bagging and LogitBoost (M5 trees) methods. But if we take a look at the $\alpha + \beta$ class, we found most algorithm don’t do well on this structural class prediction, which is considered to be difficulty predicted. Among these approaches, Bagging is the best predictor for the $\alpha + \beta$ class, while the SVMs gets much worse prediction. This implies the advantage of Bagging: the independent aggregating method can make an averagely good classification inferred from the above Equation [2].

### Table II

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>All-$\alpha$</th>
<th>All-$\beta$</th>
<th>$\alpha/\beta$</th>
<th>$\alpha + \beta$</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVMs</td>
<td>46/52</td>
<td>59/61</td>
<td>38/45</td>
<td>26/46</td>
<td>169/204 = 82.84%</td>
</tr>
<tr>
<td>LogitBoost</td>
<td>41/52</td>
<td>55/61</td>
<td>35/45</td>
<td>30/46</td>
<td>161/204 = 78.92%</td>
</tr>
<tr>
<td>LogitBoost(M5 trees)</td>
<td>42/52</td>
<td>58/61</td>
<td>36/45</td>
<td>30/46</td>
<td>166/204 = 81.37%</td>
</tr>
<tr>
<td>Bagging</td>
<td>44/52</td>
<td>57/61</td>
<td>34/45</td>
<td>33/46</td>
<td>168/204 = 82.35%</td>
</tr>
</tbody>
</table>

All the tests run on a 2 GHz Intel Core Duo Macintosh, which has two processes. Thus, another advantage of Bagging that can run in parallel mode by independent bootstrap procedure cause a significant prevail over LogitBoost (M5 trees). The run time of these algorithms of building prediction model is showed in Table III. Although the run time is not an essential problem of algorithm, however, it means that the Bagging method has potential to run on other larger scale bio-dataset, if needed.

### Table III

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVMs</td>
<td>1.23 seconds</td>
</tr>
<tr>
<td>LogitBoost</td>
<td>1.35 seconds</td>
</tr>
<tr>
<td>LogitBoost(M5 trees)</td>
<td>18.82 seconds</td>
</tr>
<tr>
<td>Bagging</td>
<td>0.82 seconds</td>
</tr>
</tbody>
</table>
Acknowledgement

The authors would like to thank Prof. Dr. K. C. Chou for providing the 204 protein structural dataset, and Dr. S. Grunewald for checking the grammar.

References and Footnotes


Date Received: July 30, 2006

Communicated by the Editor Ramaswamy H. Sarma