Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Theoretical Biology ▌(▌▌▌▌) ▌▌▌–▌▌▌

Journal of Theoretical Biology

www.elsevier.com/locate/yjtbi

# Using LogitBoost classifier to predict protein structural classes

Yu-Dong Cai[a,b], Kai-Yan Feng[c], Wen-Cong Lu[a], Kuo-Chen Chou[b,d,*]

[a]*Department of Chemistry, College of Sciences, Shanghai University, 99 Shang-Da Road, Shanghai 200436, China*
[b]*Shanghai Center for Bioinformation Technology, 100 Qing-Zhou Road, Shanghai 200235, China*
[c]*Imaging Science and Biomedical Engineering, Medical School, the University of Manchester, Manchester M13 9PT, UK*
[d]*Gordon Life Science Institute, 13784 Torrey Del Mar, San Diego, CA 92130, USA*

Received 5 April 2005; received in revised form 4 May 2005; accepted 5 May 2005

**Abstract**

Prediction of protein classification is an important topic in molecular biology. This is because it is able to not only provide useful information from the viewpoint of structure itself, but also greatly stimulate the characterization of many other features of proteins that may be closely correlated with their biological functions. In this paper, the LogitBoost, one of the boosting algorithms developed recently, is introduced for predicting protein structural classes. It performs classification using a regression scheme as the base learner, which can handle multi-class problems and is particularly superior in coping with noisy data. It was demonstrated that the LogitBoost outperformed the support vector machines in predicting the structural classes for a given dataset, indicating that the new classifier is very promising. It is anticipated that the power in predicting protein structural classes as well as many other bio-macromolecular attributes will be further strengthened if the LogitBoost and some other existing algorithms can be effectively complemented with each other.

*Keywords:* Protein structure classification; LogitBoost; Support vector machines; Amino acid composition

## 1. Introduction

Prediction of protein structural class is an important topic in protein science (see, e.g., a review by Chou, 2000). A series of previous studies have shown that some correlation between the protein structural class and amino acid composition does exist. Actually, many efforts were made to predict the structural classes of proteins based on their amino acid composition (Bahar et al., 1997; Cai et al., 2000; Cai and Zhou, 2000; Chou and Zhang, 1993, 1994, 1995; Chou, 1995; Klein and Delisi, 1986; Mao et al., 1994; Liu and Chou, 1998; Zhou, 1998; Zhou and Assa-Munt, 2001). The present study was initiated in an attempt to introduce a new approach, the so-called "LogitBoost" (Friedman et al., 2000), to predict the protein structural classes.

## 2. Boosting

Introduced first by Schapire and Singer (1999), LogitBoost is one of the boosting algorithms developed in recent years. Boosting was originally proposed to combine several weak classifiers to improve the classification performance. Later on, a more capable and practical boosting algorithm, the so-called "AdaBoost", was proposed by Freund and Schapire (1997). AdaBoost, an abbreviation for Adaptive Boosting, is a meta-learning algorithm. It tries to build a weak classifier iteratively on others according to the performance of the previous weak classifiers. Accordingly, AdaBoost is driven to focus on the hard samples by giving more weight on them that could not be correctly classified

*Corresponding author. Gordon Life Science Institute, 13784 Torrey Del Mar, San Diego, CA 92130, USA. Tel.: +1 858 484 1018; fax: +1 858 484 1018.
*E-mail address:* kchou@san.rr.com (K.-C. Chou).

0022-5193/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jtbi.2005.05.034

with the previous weak classifiers. Boosting has been used to solve various classification problems, including cancer classification (Dettling and Buhlmann, 2003; Zhou et al., 2002), text classification (Schapire and Singer, 1999), natural language processing (Haruno et al., 1999), etc. AdaBoost is able to reduce training errors exponentially fast as long as the weak classifiers perform just better than random (Freund and Schapire, 1997). It was observed (Breiman, 1998; Drucker and Cortes, 1996) that AdaBoost had very good generalization (the ability to classify new data). However, like most other classifiers, AdaBoost also suffered from the over-fit problem when dealing with very noisy data (Ratsch et al., 2001). To cope with this situation, Friedman et al. (2000) found that using LogitBoost could reduce training errors linearly and hence yield better generalization.

### 2.1. Binary LogitBoost

AdaBoost can be considered as fitting an additive logistic regression model $F(\vec{x}) = \sum_{t=1}^{T} \alpha_t f_t(\vec{x})$ to minimize the expectation of an exponential loss function $\mathrm{ELOSS}(F) = E(\mathrm{e}^{-yF(\vec{x})})$, which is monotone and smooth and can be solved effectively (Friedman et al., 2000). However, the exponential loss function changes exponentially with the classification error, rendering the AdaBoost algorithm vulnerable while handling noisy data. To solve the problem, Friedman et al. (2000) proposed a binomial log-likelihood loss function $\mathrm{LLOSS}(F) = E[-\log(1 + \mathrm{e}^{-yF(\vec{x})})]$, which changes linearly with the classification error and turns out to be less sensitive to noise and outliers. The optimization can be achieved by using Newton steps to fit an additive symmetric logistic model. The pseudo-code of LogitBoost is given in Box 1.

The construction of weak classifiers is one of the key factors affecting the performance of the boosting algorithms. The weak classifier $f_t(\vec{x})$ in step 3 should be able to cope with reweighing of the data and resistant to over-fit. Decision trees try to divide the input space into nested regions, usually rectangles, in order to minimize the least-squares error, which is quite suitable to the weak classifiers for boosting.

In step 3(a) of Box 1, the hard samples are given higher weight by putting more weights to the data potentially falling at the boundary between classes. The above LogitBoost is a binary classifier, which can only separate two classes. Here we need to separate four classes. This can be done as follows.

### 2.2. Multi-class problems

There are two common strategies to solve the multi-class problem: one is the one-vs.-others LogitBoost, and the other the multi-class LogitBoost. However, it is very difficult for the multi-class LogitBoost to define the hard samples. Here we adopt the one-vs.-others strategy (Brown et al., 2000; Ding and Dubchak, 2001) which is quite straightforward. The entire training dataset is divided into two sets in turn for each class, with one set of data belonging to the singled-out class and the rest of

Box 1
The LogitBoost Algorithm.
The Pseudo-Code of LogitBoost

```
1. Input data set $S = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_N, y_N)\}$, where $\vec{x}_i \in X,\ y_i \in Y = \{-1, 1\}$. Input number of iterations $T$.
2. Initialise the weight $w_i = 1/N (i = 1, \ldots, N)$; initialize committee function $F(\vec{x}) = 0$ and probabilities $p(\vec{x}) = P(y = 1|x) = 1/2$.
3. Repeat $t = 1, \ldots, T$
   a. Compute the weights and working response
         $w_i = p(\vec{x}_i)[1 - p(\vec{x}_i)]$
         $z_i = \frac{y_i^* - p(\vec{x}_i)}{w_i}$, where $y_i^* = (y_i + 1)/2$
   b. Fit the function $f_t(\vec{x})$ by a weighted least-squares regression of $z_i$ to $\vec{x}_i$ using weights $w_i$. In our study we use regression decision tree to fit the data $\{(\vec{x}_1, z_1), \ldots, (\vec{x}_N, z_N)\}$ using weights $w_i$.
   c. Update $F(\overleftarrow{x}) \leftarrow F(\overleftarrow{x}) + \frac{1}{2} f_t(\overleftarrow{x})$ and $p(x) \leftarrow \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}}$.
4. Output the final classifier $LF(\vec{x}) = \mathrm{sign}[F(X)]$.
```

Table 1
Comparison between LogitBoost and SVMs on the 204 proteins classified into 4 structural classes[a]

| Algorithm | All-$\alpha$ | All-$\beta$ | $\alpha/\beta$ | $\alpha+\beta$ | Overall |
|---|---|---|---|---|---|
| *Re-substitution test* | | | | | |
| SVMs[b] | $52/52 = 100\%$ | $61/61 = 100\%$ | $45/45 = 100\%$ | $46/46 = 100\%$ | $204/204 = 100\%$ |
| LogitBoost | $52/52 = 100\%$ | $61/61 = 100\%$ | $45/45 = 100\%$ | $46/46 = 100\%$ | $204/204 = 100\%$ |
| *Jackknife test* | | | | | |
| SVMs[b] | $39/52 = 75.00\%$ | $55/61 = 90.16\%$ | $29/45 = 64.44\%$ | $30/46 = 63.30\%$ | $153/204 = 75.00\%$ |
| LogitBoost | $47/52 = 90.38\%$ | $54/61 = 88.52\%$ | $36/45 = 80.00\%$ | $34/46 = 73.91\%$ | $171/204 = 83.82\%$ |

[a] The dataset used here was taken from Chou (1999).

[b] No optimization procedures were taken for the kernels and parameters in the SVMs operation.

the data (from all the other classes) belonging to another dataset. Thus if there are $k$ classes, $k$ binary LogitBoost classifiers are built. Since each LogitBoost generates the probability of a testing data belonging to the class, $k$ binary LogitBoost classifiers will output a vector of classification probability $P(\vec{x}) = [p_1(\vec{x}), p_2(\vec{x}), \ldots, p_k(\vec{x})]$. The testing data will be predicted to belong to the class with the highest probability, i.e., $C(x) = \arg \mathrm{Max}_i[p_i(x)]$. For the current case, the pairwise classes are $\alpha$-vs.-others, $\beta$-vs.-others, $(\alpha/\beta)$-vs.-others and $(\alpha+\beta)$-vs.-others.

## 3. Implementation, training and testing

The program for the one-vs.-others LogitBoost was downloaded from Dettling and Buhlmann (2003). However, instead of using stumps, the classification trees with depth three were used; then turned out to be much better than stumps because they were able to generate several unconnected regions for a category.

The working dataset was taken from Chou (1999) that contains 204 protein chains, of which 52 are all-$\alpha$ proteins, 61 all-$\beta$ proteins, 45 $\alpha/\beta$ proteins and 46 $\alpha+\beta$ proteins. Their average sequence similarity scores are 21% for all-$\alpha$, 30% for all-$\beta$, 15% for $\alpha/\beta$ and 14% for $\alpha+\beta$. Therefore, the majority of the proteins are not similar to each other in this dataset.

In this study the protein samples are represented by their amino acid compositions, and hence each input of the LogitBoost corresponds to a vector or point in a 20-dimensional space (Chou and Zhang, 1993, 1994; Chou, 1995).

## 4. Results and discussion

The demonstrations were conducted by two different approaches, the re-substitution test and the jackknife test, as reported below.

### 4.1. Success rate of the re-substitution test

The re-substitution test is used to examine the self-consistency of a prediction method. During the re-substitution process, the class for each of the proteins in the dataset is in turn identified using the rule parameters derived from the same dataset, the so-called training dataset. It was observed that by just a few iterations LogitBoost already achieved the 100% overall success rate in the self-consistency test (Table 1). The 100% success rate also indicates that LogitBoost, after undergoing an efficient training process, has grasped the complicated relationship between the amino acid composition and the structural class. It should be pointed out that during the above process the rule parameters derived from the training dataset include the information of the query protein later plugged back for testing itself. This will certainly enhance the success rate because the same samples are used to derive the rule parameters and to test themselves. Therefore, the success rate thus obtained merely represents some sort of optimal estimation (Chou, 1995; Chou and Zhang, 1994; Zhou, 1998; Zhou and Assa-Munt, 2001). Nevertheless, the re-substitution test is useful because it reflects the self-consistency. A predictor with a poor self-consistency certainly cannot be deemed as a good one. However, to really reflect the power of a predictor, a cross-validation test by excluding the tested samples from the training dataset is needed.

### 4.2. Success rate of the jackknife test

Three different examinations are often used in statistical prediction for cross-validation in the literature. They are independent dataset test, sub-sampling test and jackknife test. Of these three, however, the jackknife test is deemed as the most rigorous and objective one [see Chou and Zhang (1995) for a comprehensive discussion about this, and Mardia et al. (1979) for the underlying mathematical principle]. For the cross-validation by jackknifing, each of the proteins in the dataset is in turn singled out as a tested sample and all the rule parameters are calculated based on the remaining proteins without including the one being identified. Therefore, both the training dataset and testing dataset during the jackknifing process are actually open, and a sample will in turn move from one to the other. The overall jackknife success rate obtained by the LogitBoost was $171/204 = 83.82\%$ (Table 1).

### 4.3. Comparison with support vector machines (SVMs)

As a comparison, the support vector machines (SVMs) (Vapnik, 1998) were also applied to the same problem. In this study, the width of the Gaussian RBFs was selected as that which minimized an estimate of the VC-dimension. The parameter C that controlled the error-margin trade-off was set at 150. Because the current case was a 4-class problem, the "one-against-others" approach (Ding and Dubchak, 2001) was adopted to transfer it into a 2-class problem. For comparison, the success rates by the SVMs for the same working dataset are also given in Table 1, from which we can see that the overall jackknife success rate by the LogitBoost is about 8% higher than that by the SVMs.

## 5. Conclusion

The LogitBoost is a very powerful classifier. If it can be effectively complemented with other existing powerful algorithms, such as the covariant discriminant algorithms (Chou et al., 1998; Liu and Chou, 1998; Chou and Maggiora, 1998; Zhou, 1998; Zhou and Assa-Munt, 2001), the pseudo-amino acid composition approach (Chou, 2001, 2005; Pan et al., 2003; Wang et al., 2004), the SVM approach (Cai et al., 2004a, b), the functional domain composition approach (Chou and Cai, 2002, 2004c) and the hybridization approach (Chou and Cai, 2003a, 2004b, d, e, 2005a, c), then our power can be further strengthened in predicting the structural classes of proteins and their other important attributes such as subcellular locations (Chou and Cai, 2003c, 2005a; Chou and Elrod, 1999b; Pan et al., 2003; Xiao et al., 2005; Zhou and Doctor, 2003), membrane types (Cai et al., 2003; Chou and Cai, 2005b; Chou and Elrod, 1999a; Wang et al., 2004, 2005), enzyme family and subfamily classes (Chou, 2005; Chou and Cai, 2004b, f; Chou and Elrod, 2003), enzyme active sites (Cai et al., 2004a; Chou and Cai, 2004a), G-protein-coupled receptor classification (Chou and Elrod, 2002; Elrod and Chou, 2002) and protein quaternary structure types (Chou and Cai, 2003b).

## References

Bahar, I., Atilgan, A.R., Jernigan, R.L., Erman, B., 1997. Understanding the recognition of protein structural classes by amino acid composition. PROTEINS: Struct. Funct. Genet. 29, 172–185.

Breiman, L., 1998. Arcing classifiers. Ann. Stat. 26, 801–849.

Brown, M.P.S., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C., Ares, J.M., Haussler, D., 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc. Natl Acad. Sci. USA 97, 262–267.

Cai, Y.D., Zhou, G.P., 2000. Prediction of protein structural classes by neural network. Biochimie 82, 783–785.

Cai, Y.D., Li, Y.X., Chou, K.C., 2000. Using neural networks for prediction of domain structural classes. Biochim. Biophys. Acta. 1476, 1–2.

Cai, Y.D., Zhou, G.P., Chou, K.C., 2003. Support vector machines for predicting membrane protein types by using functional domain composition. Biophys. J. 84, 3257–3263.

Cai, Y.D., Zhou, G.P., Jen, C.H., Lin, S.L., Chou, K.C., 2004a. Identify catalytic triads of serine hydrolases by support vector machines. J. Theor. Biol. 228, 551–557.

Cai, Y.D., Pong-Wong, R., Feng, K., Jen, J.C.H., Chou, K.C., 2004b. Application of SVM to predict membrane protein types. J. Theor. Biol. 226, 373–376.

Chou, J.J., Zhang, C.T., 1993. A joint prediction of the folding types of 1490 human proteins from their genetic codons. J. Theor. Biol. 161, 251–262.

Chou, K.C., 1995. A novel approach to predicting protein structural classes in a (20-1)-D amino acid composition space. Proteins: Struct. Funct. Genet. 21, 319–344.

Chou, K.C., 1999. A key driving force in determination of protein structural classes. Biochem. Biophys. Res. Comm. 264, 216–224.

Chou, K.C., 2000. Review: Prediction of protein structural classes and subcellular locations. Curr. Protein Peptide Sci. 1, 171–208.

Chou, K.C., 2001. Prediction of protein cellular attributes using pseudo-amino-acid-composition. PROTEINS: Struct. Funct. Genet. 43, 246–255 (erratum: ibid., 2001, Vol. 44, 60).

Chou, K.C., 2005. Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics 21, 10–19.

Chou, K.C., Cai, Y.D., 2002. Using functional domain composition and support vector machines for prediction of protein subcellular location. J. Biol. Chem. 277, 45765–45769.

Chou, K.C., Cai, Y.D., 2003a. A new hybrid approach to predict subcellular localization of proteins by incorporating gene ontology. Biochem. Biophys. Res. Comm. 311, 743–747.

Chou, K.C., Cai, Y.D., 2003b. Predicting protein quaternary structure by pseudo amino acid composition. PROTEINS: Struct. Funct. Genet. 53, 282–289.

Chou, K.C., Cai, Y.D., 2003c. Prediction and classification of protein subcellular location: sequence-order effect and pseudo amino acid composition. J. Cellular Biochem. 90, 1250–1260 (Addendum, ibid. 2004, 91, No.5, P.1085).

Chou, K.C., Cai, Y.D., 2004a. A novel approach to predict active sites of enzyme molecules. PROTEINS: Struct. Funct. Genet. 55, 77–82.

Chou, K.C., Cai, Y.D., 2004b. Predicting enzyme family class in a hybridization space. Protein Sci. 13, 2857–2863.

Chou, K.C., Cai, Y.D., 2004c. Predicting protein structural class by functional domain composition. Biochem. Biophys. Res. Comm. 321, 1007–1009 (corrigendum: ibid., 2005, Vol. 329, 1362).

Chou, K.C., Cai, Y.D., 2004d. Predicting subcellular localization of proteins by hybridizing functional domain composition and pseudo-amino acid composition. J. Cell. Biochem. 91, 1197–1203.

Chou, K.C., Cai, Y.D., 2004e. Prediction of protein subcellular locations by GO-FunD-PseAA predicor. Biochem. Biophys. Res. Comm. 320, 1236–1239.

Chou, K.C., Cai, Y.D., 2004f. Using GO-PseAA predictor to predict enzyme sub-class. Biochem. Biophys. Res. Comm. 325, 506–509.

Chou, K.C., Cai, Y.D., 2005a. Predicting protein localization in budding yeast. Bioinformatics 21, 944–950.

Chou, K.C., Cai, Y.D., 2005b. Prediction of membrane protein types by incorporating amphipathic effects. J. Chem. Info. Model. 45, 407–413.

Chou, K.C., Cai, Y.D., 2005c. Using GO-PseAA predictor to identify membrane proteins and their types. Biochem. Biophys. Res. Comm. 327, 845–847.

Chou, K.C., Elrod, D.W., 1999a. Prediction of membrane protein types and subcellular locations. PROTEINS: Struct. Funct. Genet. 34, 137–153.

Chou, K.C., Elrod, D.W., 1999b. Protein subcellular location prediction. Protein Eng. 12, 107–118.

Chou, K.C., Elrod, D.W., 2002. Bioinformatical analysis of G-protein-coupled receptors. J. Proteome Res. 1, 429–433.

Chou, K.C., Elrod, D.W., 2003. Prediction of enzyme family classes. J. Proteome Res. 2, 183–190.

Chou, K.C., Maggiora, G.M., 1998. Domain structural class prediction. Protein Eng. 11, 523–538.

Chou, K.C., Zhang, C.T., 1994. Predicting protein folding types by distance functions that make allowances for amino acid interactions. J. Biol. Chem. 269, 22014–22020.

Chou, K.C., Zhang, C.T., 1995. Review: Prediction of protein structural classes. Crit. Rev. Biochem. Mol. Biol. 30, 275–349.

Chou, K.C., Liu, W., Maggiora, G.M., Zhang, C.T., 1998. Prediction and classification of domain structural classes. PROTEINS: Struct. Funct. Genet. 31, 97–103.

Dettling, M., Buhlmann, P., 2003. Boosting for tumor classification with gene expression data. Bionformatics 19, 1061–1069.

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

Drucker, H., Cortes, C., 1996. Boosting decision trees. Adv. Neural Inf. Process. Syst. 8, 479–485.

Elrod, D.W., Chou, K.C., 2002. A study on the correlation of G-protein-coupled receptor types with amino acid composition. Protein Eng. 15, 713–715.

Freund, Y., Schapire, R., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55, 119–139.

Friedman, J., Hastie, T., Tibshirani, R., 2000. Additive logistic regression: a statistical view of boosting. Ann. Stat. 337–407.

Haruno, M., Shirai, S., Ooyama, Y., 1999. Using decision trees to construct a practical parser. Mach. Learning 34, 131–149.

Klein, P., Delisi, C., 1986. Prediction of protein structural class from amino acid sequence. Biopolymers 25, 1659–1672.

Liu, W., Chou, K.C., 1998. Prediction of protein structural classes by modified Mahalanobis discriminant algorithm. J. Protein Chem. 17, 209–217.

Mao, B., Chou, K.C., Zhang, C.T., 1994. Protein folding classes: a geometric interpretation of the amino acid composition of globular proteins. Protein Eng. 7, 319–330.

Mardia, K.V., Kent, J.T., Bibby, J.M., 1979. Multivariate Analysis: Chapter 11 Discriminant analysis; Chapter 12 Multivariate analysis of variance; Chapter 13 Cluster analysis (pp. 322–381). Academic Press, London.

Pan, Y.X., Zhang, Z.Z., Guo, Z.M., Feng, G.Y., Huang, Z.D., He, L., 2003. Application of pseudo amino acid composition for predicting protein subcellular location: stochastic signal processing approach. J. Protein Chem. 22, 395–402.

Ratsch, G., Onoda, T., Muller, K.R., 2001. Soft margins for AdaBoost. Mach. Learning 42, 287–320.

Schapire, R.E., Singer, Y., 1999. Improved boosting algorithms using confidence-rated predictions. Mach. Learning 37, 297–336.

Vapnik, V., 1998. Statistical Learning Theory. Wiley-Interscience, New York.

Wang, M., Yang, J., Liu, G.P., Xu, Z.J., Chou, K.C., 2004. Weighted-support vector machines for predicting membrane protein types based on pseudo amino acid composition. Protein Eng. Des. Selection 17, 509–516.

Wang, M., Yang, J., Xu, Z.J., Chou, K.C., 2005. SLLE for predicting membrane protein types. J. Theor. Biol. 232, 7–15.

Xiao, X., Shao, S., Ding1, Y., Huang, Z., Huang, Y., Chou, K.C., 2005. Using complexity measure factor to predict protein subcellular location. Amino Acida 28, 57–61.

Zhou, G.P., 1998. An intriguing controversy over protein structural class prediction. J. Protein Chem. 17, 729–738.

Zhou, G.P., Assa-Munt, N., 2001. Some insights into protein structural class prediction. PROTEINS: Struct. Funct. Genet. 44, 57–59.

Zhou, G.P., Doctor, K., 2003. Subcellular location prediction of apoptosis proteins. PROTEINS: Struct. Funct. Genet. 50, 44–48.

Zhou, Z.H., Jiang, Y., Yang, Y.B., Chen, S.F., 2002. Lung cancer cell identification based on artificial neural network ensembles. Artif. Intel. Med. 24, 25–36.