Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

A Bootstrap-based Linear Classifier Fusion System for Protein Subcellular Location Prediction

Yunfeng Wu, Yuezhu Ma, Xiaona Liu, and Cong Wang

Abstract—**The subcellular location plays a pivotal role in the functionality of proteins. In this paper we develop a multi-stage linear classifier fusion system based on Efron's bootstrap sampling for predicting subcellular locations of yeast proteins. Three different types of classifiers, i.e. the Naive Bayes (NB) classifier, Radial Basis Function (RBF) network, and Multilayer Perceptron (MLP), are utilized to construct the component modules in the fusion system. Ten bootstrapped instance sets are generated for training each type of component classifiers respectively. The linear fusion models, updated by the Least-Mean-Square (LMS) algorithm, are used to integrate the local decisions of the component classifiers and derive the final predictions. The empirical results show that the RBF classifiers can reach at slightly higher accuracy and better precision versus the NB or MLP ones. The linear fusion system consistently improves the overall prediction accuracy, in particular 6.65%, 1.77%, and 3.21%, superior to the NB, RBF, and MLP component classifiers, respectively.**

I. Introduction

The investigations in the field of bioinformatics and computational biology have been blooming for years, especially since the eventual completion of the human genetic sequence in the Human Genome Project [1]. One of recent challenges in bioinformatics is the representation of the mass of sequence information, with a view not only to deriving more efficient means of data storage, but also to developing more effective mathematical and statistical tools for analysis of the composition and structure of biomolecules [2].

Apart from a minority of proteins which are coded in the genomes of mitochondria and chloroplasts, all other proteins are synthesized in the cytosol [3]. Proteins need to be sorted to one or other subcellular compartment to perform their functions [4]. Since the subcellular location of a protein affects its potential functionality as well as its accessibility to drug treatments [5], it is therefore essential to develop computational systems in order to expedite the functionality determination of new proteins, which can be used in the prioritization of genes and proteins identified by genomic efforts as potential molecular targets for drug design.

With the pioneering effort of Nakai *et al.* [6], [7], the rule-based expert system was first introduced for predicting protein subcellular locations. The system works with two rule groups: the first group stores results of several subprograms in the working memory; and the second group later utilizes these results to make a prediction. The drawback of such an expert system is that it required a time-consuming hand-tuned training process. In [8], Horton *et al.* used the probabilistic reasoning and decision tree models in order to remedy this weakness, however, the prediction accuracy still stays no high and the interpretation of relationships between the classes in the decision tree mainly depends on a human expert's effort.

As one of the prevailing soft computing technologies, artificial neural networks have been extensively applied in building computer-assisted decision aids used in medical diagnosis [9] and analyzing complex biological systems [10]. Nowadays it becomes more and more popular to design multiple classifier systems [11], [12], which may provide performance improvement over a sole classifier, when solving complex classification problems [13]–[16]. Different classifiers may have various lopsided decisions complying with their knowledge generalization principle. It is therefore necessary to develop multiple classifier systems that combine the variant knowledge acquired by each component classifier by following a given fusion strategy, in order to gain a better generalization ability. Among a variety of fusion strategies [17], the linear fusions [18], [19] are most frequently used [13], [14], [20]. In this paper, we develop a bootstrap-based multiple classifier fusion system for predicting the subcellular locations of yeast proteins. The system contains a bootstrap sampling procedure which generates a series of instance sets by sampling with replacement from the original data set. Then the component classifiers are independently trained with these bootstrapped instance sets, and their outputs are linearly combined with the Least-Mean-Square (LMS) fusion strategy to form the final prediction.

The paper is organized as follows. Section II describes in detail the composition of our prediction system, including the bootstrap sampling approach, the three types of component classifiers, and the LMS algorithm for updating the linear fusion models. Section III discusses the empirical results of subcellular location prediction in yeast protein amino acid sequences. Concluding remarks and future directions are presented in Section IV.

The authors would like to acknowledge the support provided by National Science Foundation of China under the Grant No. 60575034, and by the 2005 Innovation Research Funds from Graduate School, Beijing University of Posts and Telecommunications.

Mr. Yunfeng Wu, Ms. Yuezhu Ma, Ms. Xiaona Liu, and Prof. Cong Wang are with School of Information Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China.

II. Protein Subcellular Location Prediction System

A. Dataset

The data set of yeast proteins was obtained by the subprograms of the expert system developed by Nakai *et al.* [6], [7], and is also online available from the UCI machine learning repository [21]. The 8 features calculated from a total of 1484 amino acid sequences include: the presence or absence of an HDEL pattern (substring) as a signal for retention in the endoplasmic reticulum lumen [22]; the result of discriminant analysis on the amino acid content of vacuolar and extracellular proteins; the result of discriminant analysis on the amino acid composition of the 20-residue N-terminal region of mitochondrial and non-mitochondrial proteins; the presence or absence of nuclear localization consensus patterns [23] combined with a term reflecting the frequency of basic residue; and some combination of the presence of a short sequence motif and the result of discriminant analysis of the amino acid composition of the protein sequence; a modification of McGeoch's signal sequence detection parameter [24]; the output of a weight matrix method [25] for detecting cleavable signal sequences; the output of the ALOM program [26] for identifying membrane spanning regions on the amino acid sequences.

The task can be simplified to categorize the yeast proteins into 10 disjointed locations (mentioned as *classes* hereafter): cytoplasmic, including cytoskeletal (CYT, 463 instances); nuclear (NUC, 429 instances); vacuolar (VAC, 30 instances); mitochondrial (MIT, 244 instances); peroxisomal (POX, 20 instances); extracellular, including those localized to the cell wall (EXC, 35 instances); proteins localized to the lumen of the endoplasmic reticulum (ERL, 5 instances); membrane proteins with a cleaved signal (ME1, 44 instances); membrane proteins with an uncleaved signal (ME2, 51 instances); and membrane proteins with no N-terminal signal (ME3, 163 instances), where ME1, ME2, and ME3 proteins may be localized to the plasma membrane, the endoplasmic reticulum membrane, or the membrane of a golgi body.

B. Bootstrap Sampling

Bootstrap sampling [27], which is effectively used in many ensemble learning algorithms, e.g. Bagging [28], can help reduce prediction variance and overcome over-fitting. The motivation of introducing the Efron's bootstrap in our system is to obtain reliable standard errors and confidence intervals, without making assumptions about the distribution of the original data. Assume that we have a *P*-size set of training feature-class instances $S = \{(\mathbf{x}, \omega_m)_{p=1}^{P}\}$, where the vector $\mathbf{x} = [x_1, \cdots, x_N]^{\mathrm{T}}$ represents the input features of the yeast sequences and $\omega_m \in \boldsymbol{\omega}^M$ denotes the corresponding location class. New training sets S^k, $k = 1, \cdots, K$, also of size *P*, are sampled uniformly from *S* with replacement, and used for generating the *K* component classifiers.

C. Component Classifiers in the Fusion

We consider three different types of component classifiers, i.e., Naive Bayes (NB) classifier, Radial Basis Function (RBF) network, and Multilayer Perceptron (MLP), to form the linear fusion decision in the prediction system.

1) Naive Bayes Classifiers: In spite of its simplicity, the NB classifier has been widely used in practical applications [29]. With the assumption that the class conditional probability densities $p(\mathbf{x}|\omega_m)$ of the instances are mutual independent within each class, the NB classifier provides the maximum likelihood solutions as

$$\omega_{NB} = \arg\max_{m=1}^{M} P(\omega_m) \prod_{n=1}^{N} p(x_n|\omega_m) \tag{1}$$

where $P(\omega_m)$ denotes the *a priori* probabilities of classes.

2) Radial Basis Function Classifiers: RBF and MLP are two well-known types of nonlinear input-output mapping networks, which have been employed in a variety of practical applications, e.g., pattern analysis, function approximation, time-series signal prediction. Typically, it has been justified by researchers [30]–[33], that a feedforward neural network can approximate any continuous function within an arbitrary accuracy, provided that its topology includes a sufficient number of hidden nodes.

The design of the RBF component classifiers is as follows. According to Duda *et al.* [29], the numbers of input and output nodes of a RBF are equal to the dimensionality of the input features and the number of categories, respectively. The 30 centers in the Gaussian kernel function were fixed from the training sets [29], and the spread parameter σ was set to be 4.0, which could lead the nonlinear nodes to respond strongly to the overlapping regions of the input space [34].

3) Multilayer Perceptron Classifiers: Although having the similar nonlinear layered structure, a MLP performs global approximations with its inner log-sigmoid activation nodes, whereas a RBF network uses a large number of exponentially decaying localized Gaussian kernel functions to construct local approximations. In the prediction system, we also employed a group of MLP component classifiers with the identical architecture (*8-20-10*)[1], and trained them by the Resilient Backpropagation algorithm [35].

D. Linear Fusion Model with the LMS Update Algorithm

A combination of classifiers with a trained fusion strategy aims to integrate their local decisions to achieve superiority [15]. In our prediction system, a total of 10 linear combination models (equivalent to the number of yeast locations) updated[2] by the LMS algorithm were employed in the fusion module, because a single LMS fusion model is

[1] In accordance with notation usage in this paper, we dictate that a feedforward neural network with (*N-J-M*) architecture contains *N* input nodes, *J* hidden nodes, and *M* output nodes.

[2] For distinction purpose, we state in this paper that the generations of component classifiers are via *training* processes, and the generations of linear fusion models are via *update* processes.

only competent for binary classification, due to its linear characteristic. When solving a multi-class problem, the number of the LMS fusion models is usually set to be the same as the number of classes. As we mentioned in Section II, each component classifier was independently trained with a bootstrapped set, in this case we may have a total of K component classifiers on hand. Thus the output of a LMS fusion model can be expressed as

$$f = \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{o} \tag{2}$$

where $\boldsymbol{\alpha} = [\mathrm{bias}, \alpha_1, \cdots, \alpha_K]^{\mathrm{T}}$ represents the vector of the bias parameter and fusion coefficients correspondingly allocated to the outputs of component classifiers, $\mathbf{o} = [o_0, o_1, \cdots, o_K]^{\mathrm{T}}$, particularly, o_0 is fixed at +1 and associated with the bias. According to the LMS error criterion, the instantaneous cost function on the i-th update iteration is

$$C(i) = e(i)^2/2 \tag{3}$$

where $e(i)$ is the instantaneous estimate error, i.e.,

$$e(i) = \omega(i) - \hat{f}(i) = \omega(i) - \hat{\boldsymbol{\alpha}}(i)^{\mathrm{T}}\mathbf{o}(i) \tag{4}$$

Differentiating (3) with respect to fusion coefficients yields the gradient approximation of $C(i)$, i.e.,

$$\begin{aligned}\nabla_{\boldsymbol{\alpha}} C(i) &\approx \frac{1}{2}\frac{\partial e(i)^2}{\partial \boldsymbol{\alpha}}\bigg|_{\boldsymbol{\alpha}=\hat{\boldsymbol{\alpha}}(i)} \\ &= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\alpha}}\left[\omega(i)^2 - 2\omega(i)\hat{\boldsymbol{\alpha}}(i)^{\mathrm{T}}\mathbf{o}(i) + \hat{\boldsymbol{\alpha}}(i)^{\mathrm{T}}\left(\mathbf{o}(i)\mathbf{o}(i)^{\mathrm{T}}\right)\hat{\boldsymbol{\alpha}}(i)\right] \\ &= -\omega(i)\mathbf{o}(i) + \left(\hat{\boldsymbol{\alpha}}(i)^{\mathrm{T}}\mathbf{o}(i)\right)\mathbf{o}(i) \\ &= -\left(\omega(i) - \hat{\boldsymbol{\alpha}}(i)^{\mathrm{T}}\mathbf{o}(i)\right)\mathbf{o}(i) = -e(i)\mathbf{o}(i)\end{aligned} \tag{5}$$

Hence the fusion coefficients update rule can be written following the steepest descent gradient method as

$$\boldsymbol{\alpha}(i+1) = \boldsymbol{\alpha}(i) + \lambda[-\nabla_{\boldsymbol{\alpha}} C(i)] = \boldsymbol{\alpha}(i) + \lambda e(i)\mathbf{o}(i) \tag{6}$$

where the parameter $\lambda > 0$ is commonly regarded as the update rate which specifies the magnitude of the update step for the fusion coefficients in the negative gradient direction.

Then, the highest output among the LMS fusion models is assigned as the final prediction class, which can be regarded as one-versus-rest scheme, i.e.,

$$\text{instance } \mathbf{x} \to \text{class } \omega_l \quad \text{if } f_l(\mathbf{x}) = \max_{m=1}^{M} f_m(\mathbf{x}), l, m \in M \tag{7}$$

where M is the total number of location classes.

III. Empirical Results

Table I provides a summary of prediction accuracy achieved by three types of component classifiers trained by different bootstrapped instance sets (labeled as *Bootstrapped Networks* herein). The averaged performance of each type of component classifiers is quantized in terms of mean and standard deviation values. As one can observe, the RBF classifiers can attain averaged 60.02% prediction accuracy, 1.37% and 1.64% better than the NB and MLP classifiers, respectively. Moreover, the RBF classifiers also excel in prediction precision when fitting a series of bootstrapped sets because their standard deviation of accuracy is slightly lower than those of the other two types of classifiers. In other words, the RBF classifier, as one style of neural classifiers, shows more robust than the rest ones in our experiments.

TABLE I Results of Protein Subcellular Location Prediction By Component Classifiers on Different Bootstrapped Sets

Bootstrapped Network ID	Prediction Accuracy (%) of Component Classifiers		
	NB	RBF	MLP
#1	58.56	60.85	58.76
#2	58.96	60.85	58.49
#3	57.55	58.96	57.68
#4	59.43	59.10	58.15
#5	59.23	60.24	58.22
#6	58.69	59.10	57.35
#7	60.04	60.11	57.82
#8	57.28	60.65	58.29
#9	58.42	60.11	59.97
#10	58.36	60.24	59.03
Mean	**58.65**	**60.02**	**58.38**
Standard Deviation	**0.83**	**0.72**	**0.75**

Fig. 1 illustrates the performance of the LMS fusions which linearly combine three types of component classifiers. And it is clear that the percentage accuracy have been ameliorated in virtue of the LMS fusions. The entire fusion results in terms of prediction accuracy and the Mean Squared Error (MSE) are listed in Table II. It is interesting that the accuracy improvement via the LMS fusion is great for the NB classifiers (a significant 6.65% increase) and the MLP classifiers (a 3.21% increase), whereas only a 1.77% increase for the RBF classifiers. Our preliminary hypothesis is that the fusion prediction may be reinforced by the variance (or referred to *diversity* in [36]) of local classifiers.

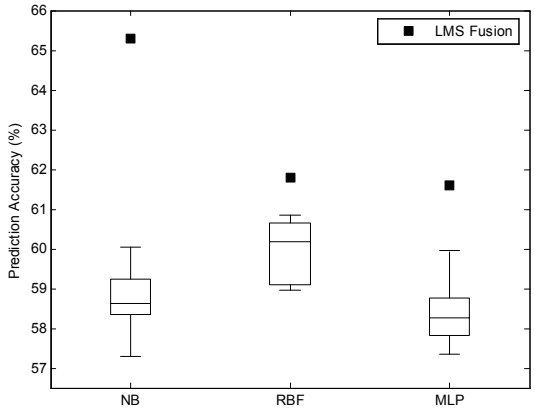

Fig. 1. Prediction accuracy of the LMS fusions comparing with the mean and extent values of their component classifiers.

TABLE II Prediction Results of the LMS Fusions

Fusion Kernels	LMS-NB	LMS-RBF	LMS-MLP
MSE	0.0481	0.0536	0.0558
Accuracy	**65.30%**	**61.79%**	**61.59%**

The specific class-by-class location prediction results of

the LMS fusions are displayed in Table III. It can be found that the LMS fusions for the RBF and MLP component classifiers perform quite similar, whereas the LMS-NB exhibits its superiority for most instances.

TABLE III PREDICTION RESULTS (%) OF SPECIFIC SUBCELLULAR LOCATIONS OF YEAST PROTEINS

Instances	Class	LMS-NB	LMS-RBF	LMS-MLP
463	CYT	58.32	71.27	71.71
429	NUC	66.43	49.88	52.68
244	MIT	66.39	59.43	59.43
163	ME3	78.53	84.66	84.05
51	ME2	64.71	37.25	13.73
44	ME1	93.18	77.27	79.55
35	EXC	74.29	60.00	45.71
30	VAC	20.00	0.00	0.00
20	POX	65.00	55.00	55.00
5	ERL	100.00	100.00	100.00

IV. CONCLUSION

The fusion system is considered to have the merits of integrating variant knowledge learned by multiple classifiers to provide a comprehensive solution. Our empirical results of demonstrate the advantages of the linear classifier fusion system for prediction of protein subcellular locations. The future work is to design new hybrid computational prediction systems with a variety of learners involved, and to investigate how the diversity of component learners affects the final fusion result.

REFERENCES

[1] J. Pevsner, *Bioinformatics and Functional Genomics*, John Wiley, NY: New York, 2003.

[2] Z.P. Feng, "An overview on predicting the subcellular location of a protein," *In Silico Biology*, vol. 2, pp. 291–303, 2002.

[3] K.C. Chou and D.W. Elrod, "Protein subcellular location prediction," *Protein Engineering*, vol. 12, pp. 107–118, 1999.

[4] K.C. Chou, "Prediction of protein signal sequences and their cleavage sites," *Proteins*, vol. 42, pp. 136–139, 2000.

[5] O. Emanuelsson, "Predicting protein subcellular localisation from amino acid sequence information," *Briefings in Bioinformatics*, vol. 3, no. 4, pp. 361–376, 2002.

[6] K. Nakai and M. Kanehisa, "Expert system for predicting protein localization sites in gram-negative bacteria," *Proteins: Structure, Function and Genetics*, vol. 11, pp. 95–110, 1991.

[7] K. Nakai and M. Kanehisa, "A knowledge base for predicting protein localization sites in eukaryotic cells," *Genomics*, vol. 14, pp. 897–911, 1992.

[8] P. Horton and K. Nakai, "A probabilistic classification system for predicting the cellular localization sites of proteins," *Intelligent Systems in Molecular Biology*, vol. 4, pp. 109–115, 1996.

[9] T. Andre and R.M. Rangayyan, "Classification of tumors and masses in mammograms using neural networks with shape and texture features," in *Proc. 25th IEEE EMBS Annu. Int'l Conf. (EMBC'03)*, Cancun, Mexico, 2003, vol. 3, pp. 2261–2264.

[10] D.L. Hudson and M.E. Cohen, *Neural Networks and Artificial Intelligence for Biomedical Engineering*, New York, NY: IEEE Press, 1999.

[11] L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Hoboken, NJ: Wiley, 2004.

[12] F. Roli and G. Giacinto, "Design of multiple classifier systems," in H. Bunke and A. Kandel (Eds.) *Hybrid Methods in Pattern Recognition*, Singapore: World Scientific Publishing, 2002.

[13] Y. Wu, J. He, Y. Man, and J.I. Arribas, "Neural network fusion strategies for identifying breast masses," in *Proc. 2004 Int'l Joint Conf. Neural Networks (IJCNN'04)*, Budapest, Hungary, 2004, vol. 3, pp. 2437–2442.

[14] Y. Wu, J.M. Zhang, C. Wang, and S.C. Ng, "Linear decision fusions in multilayer perceptrons for breast cancer diagnosis," in *Proc. 17th IEEE Int'l Conf. Tools with Artificial Intelligence (ICTAI'05)*, Hong Kong, 2005, pp. 699–700.

[15] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, Mar 1998.

[16] Y. Wu and J.I. Arribas, "Fusing output information in neural networks: Ensemble performs better," in *Proc. 25th IEEE EMBS Annu. Int'l Conf. (EMBC'03)*, Cancun, Mexico, 2003, vol. 3, pp. 2265–2268.

[17] L.I. Kuncheva, "A theoretical study on six classifier fusion strategies," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 281–286, Feb 2002.

[18] N. Ueda, "Optimal linear combination of neural networks for improving classification performance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 2, pp. 207–215, Feb 2000.

[19] S. Hashem and B. Schmeiser, "Improving model accuracy using optimal linear combinations of trained neural networks," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 792–794, May 1995.

[20] Y. Wu and C. Wang, "Linear least-squares fusion of multilayer perceptrons for protein localization sites prediction," in *Proc. 32nd IEEE Annu. Northeast Bioeng. Conf. (NEBC'06)*, Easton, PA, USA, 2006, pp.157-158.

[21] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. (1998). *UCI repository of machine learning databases*. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html .

[22] H.R.B. Pelham, "The retention signal for soluble proteins of the endoplasmic reticulum," *Trends Biochem. Sci.*, vol. 15, pp. 482─486, 1990.

[23] G. von Heijne, "The structure of signal peptides from bacterial lipoproteins," *Protein Engineering,* vol. 2, pp. 531–534, 1989.

[24] D.J. McGeoch, "On the predictive recognition of signal peptide sequences," *Virus Research*, vol. 3, pp. 271–286, 1985.

[25] G. von Heijne, "A new method for predicting signal sequence cleavage sites," *Nucleic Acids Research*, vol. 14, pp. 4683–4690, 1986.

[26] P. Klein, M. Kanehisa, and C. DeLisi, "The detection and classification of membrane-spanning proteins," *Biochim. Biophy. Acta*, vol. 815, pp. 949–951, 1985.

[27] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*, New York, NY: Chapman and Hall, 1993.

[28] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996.

[29] R.O. Duda, P.E. Hart, and D.G. Strock, *Pattern Classification, 2nd ed.*, New York, NY: Wiley, 2001.

[30] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximatiors," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[31] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[32] E.J. Hartman, J.D. Keeler, and J.M. Kowalsky, "Layered neural networks with Gaussian hidden units as universal approximators," *Neural Computation*, vol. 2, no. 2, pp. 210–215, 1990.

[33] J. Park and I.W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991.

[34] S. Haykin, *Neural Networks: A Comprehensive Foundation, 2nd ed.*, Englewood Cliffs, NJ: Prentice Hall PTR, 1998.

[35] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Proc. 1993 IEEE Int'l Conf. Neural Networks (ICNN'93)*, San Francisco, CA, USA, 1993 vol. 1, pp. 586-591.

[36] T. Windeatt, "Diversity measures for multiple classifier system analysis and design," *Information Fusion*, vol. 6, no. 1, pp.21-36, 2005.