Journal of Computational Intelligence in Bioinformatics
ISSN0973-385X Volume 6, Number 2 (2013) pp. 93-102
© Research India Publications
http://www.ripublication.com/jcib.htm

Protein Function Prediction using Artificial Neural Network (Dynamic) Model

SudhakarTripathi[1*], Arvind Kumar Tiwari[2] and R.B.Mishra[3]

*[1, 2, 3]Department of Computer Engineering
Indian Institute of Technology(BHU)
Varanasi, U.P., 221005, India
E-mail: stripathi.rs.cse@iitbhu.ac.in[1], arvind.rs.cse12@itbhu.ac.in[2],
ravibm@bhu.ac.in[3]*

Abstract

Protein function prediction is very important and challenging task in Bioinformatics. In this paper we have used proteins represented by a set of enzymes i.e. Oxidoreductase, Transferase, Hydrolase, Isomerase, Ligase, and Lyase, extracted from the Enzyme Commission (EC) classification to build the models. We have used ANN (Dynamic) model to predict protein function and the result of this model have been compared to SVM, C5, and CHAID model. ANN (Dynamic) model predicts the protein function with high accuracy i.e. >90% compared to that of SVM, C5 and CHAID.ANN (Dynamic) model have been build using balance importance of all the features whereas other models based on some important feature. We have usedprotein dataset available at PDB using features such as primary structures, secondary structures, molecular weight, structural molecular weight, chain length, atom count as training parameters andEC number as corresponding output. The result in this paper shows that ANN (Dynamic) model prediction accuracy is high in overall prediction accuracy as well as individual class prediction accuracy.

Keywords-Proteins, prediction, ANN (Dynamic), SVM, C5, CHAID.

INTRODUCTION

Proteins are formed from a set of 20 amino acids and the function of a protein is closely related to the structure. There are various function of protein such as catalysis, transport and Information.Enzyme behaves like a catalyst which speed up the rate of reaction without becoming the part of reaction. The primarystructure of a protein isthe sequence of amino acids, secondary structure is theformation of alpha helixes, beta sheets and loops and thetertiary structure isresponsible for the spatial arrangement of the proteinandthequaternary structurerefers to the proteins that have more than one chain of amino acids. In this paper we used the proteins that are classified according to EC number. Finding of protein function is an important task which supports into the research of drugs design. In this paper we used six class of enzymes Oxidoreductase, Transferase, Hydrolase, Isomerase, Ligase, and Lyase. In this paper we used six features primary structures, secondary structures, molecular weight, structural molecular weight, chain length, atom count to predict the protein function. Using these features we construct the model by using various classifiers such as SVM, C5 and CHAID andArtificial Neural Network(ANN)(Dynamic).

After comparative study of this entire model we find that ANN (Dynamic) model approach is more accurate in protein function prediction.

RELATED WORK

Here we describe the previous research work carried out for the protein function prediction or classification and we also discuss about the various classifiers models that are used in this study. [1] proposed a method to predict functional family of protein that is useful for protein function prediction. Every protein sequence is represented by a set of amino acid composition by using these composition he used SVM, supervised machine learning and the result of this model is compared with the Naïve Bayes and C 4.5. [2] used the SVM for protein function classification. He used a various protein classes such as RNA-binding, homodimer, drug absorption, drug excretion etc. He found the testing accuracy between 84-96%.Reference [3] proposed a method that can assign the function from the structure of protein by using EC number. He used one-class versus one-class SVM to predict the protein function. He found the between 35-60%. [4] proposed a method for predicting EC number. He used various features of the protein structure find from STRING_DB and used Bayesian classifier to predict the protein function. He found the accuracy 45.3%. [5] proposed a method to predict enzyme functions using amino acid composition, their neighborhood relationship to each other, and the hierarchical structure of the class. He compared the results from the attributes considered and concludes that the information from all three together offers better results. Using the SVM classifier, they obtain a prediction rate of between 81% and 98%.

METHODOLOGY

MODELS

Following models are used in this paper:-

1) SVM

The SVM can be characterized as a machine learning algorithm capable of resolving linear and non-linear classification problems. The principal idea of classification by support vector is to separate examples with a linear decision surface and maximize the margin of separation between the classes to be classified [11]. Originally called "optimal margin classifier", the SVM was introduced in [12] for application in binary classification problems. In [13], where it was called a "support vector network", a method was proposed for dealing efficiently with notably incorrect examples, specifically, those outwiththe region of their class. The name "Support Vector Machine" (SVM) emphasizes the importance of the vectors closest to the margin of separation due to the fact that they determine the complexity of the SVM [14].The SVM is based on Statistical Learning Theory, usingthe inductive principle of Structural Risk Minimization. The learning process is supervised where the training data, along with the corresponding outputs, are presented to the machine so that its parameters can be adjusted [11].SVM uses kernel denominated functions. These functions are capable of mapping the data set in different spaces, making it possible to use a hyper plane to do the separation. This directly influences the results obtained by the classifier. The parameters of the SVM are highly sensitive and vary for each problem and data set.

2) C5

C5 (improvement of C4.5)is an algorithm used to generate a decision tree developed by Ross Quinlan[7][8].The changes in various versions of C5 are available at [8] The decision trees generated by C5 can be used for classification, and it is used as a statistical classifier. C5 builds decision trees or corresponding rule sets from training data set, using the concept of information entropy. The training data setis a set S_T= s_1, s_2, ... of already classified samples, known as supervised training. Each sample s_i = x_1, x_2, ... is a linear vector where x_1, x_2, ... represent input features of the sample. The training data is augmented with a vector C = c_1, c_2, ...$c_n$wherec_1, c_2, ... represent the class to which each sample belongs and n is total number of classes[6]. At each node of the tree, C5 chooses one feature of the data setthat most effectively splits its set of samples into subsets enriched in one class or the other. The criterion for splitting is the normalized information gain (difference in entropy) that results from choosing a feature for splitting the data into subsets [6]. The feature with the highest normalized information gain is chosen to make the decision. The C5 algorithm then follows the same steps on the smaller sub lists [6].

To maximize interpretability, C5.0 classifiers are expressed as decision trees or sets of if-then rules, forms that are generally easier to understand than neural networks. C5.0 is easy to use and does not presume any special knowledge of Statistics or Machine Learning.

3)CHAID(CHi-squared Automatic InteractionDetection)

CHAID stands for Chi-squared Automatic Interaction Detector. It is a highly efficient statistical technique for segmentation, or tree growing, developed by Gordon V. Kass[9]. Using the significance of a statistical test as a criterion, CHAID evaluates all of the values of a potential predictor field. It merges values that are judged to be statistically homogeneous (similar) with respect to the target variable and maintains all other values that heterogeneous (dissimilar). It then selects the best predictor to form the first branch in the decision tree, such that each child node is made of a group of homogeneous values of the selected field. This process continues recursively until the tree is fully grown. The statistical test used depends upon the measurement level of the target field. If the target field is continuous, an *F* test is used. If the target field is categorical, a chi-squared test is used. CHAID is not a binary tree method; that is, it can produce *more* than two categories at any particular level in the tree. Therefore, it tends to create a wider tree than do the binary growing methods. It works for all types of variables, and it accepts both case weights and frequency variables. It handles missing values by treating them all as a single valid category.

4)ANN (DYNAMIC)

As described in [10], the basic element of a neural network is a neuron. This is a simple virtual device that accepts many inputs, sums them, applies a (usually nonlinear) transfer function, and generates the result, either as a model prediction or as input to other neurons.

A neural network is a structure of many such neurons connected in a systematic way. In Clementine, the neural networks used are feed-forward neural networks, also known as multilayerperceptrons. The neurons in such networks (sometimes called units) are arranged in layers. Typically, there is one layer for input neurons (the input layer), one or more layers of internal processing units (the hidden layers), and one layer for output neurons (the output layer).Each layer is fully interconnected to the preceding layer and the following layer. For example, in a network with an input layer, a single hidden layer, and an output layer, each neuron in the input layer is connected to every neuron in the hidden layer, and each neuron in the hidden layer is connected to every neuron in the output layer. The connections between neurons have weights associated with them, which determine the strength of influence one neuron has on another. Information flows from the input layer through the processing layer(s) to the output layer to generate predictions. By adjusting the connection weights during training to match predictions to target values for specific records, the network learns to generate better and better predictions.

Dynamic Method

When the dynamic method is selected, the topology of the network changes during training, with neurons added to improve performance until the network achieves the desired accuracy. There are two stages to dynamic training: finding the topology and training the final network.

Finding the Topology

Finding the topology follows these steps:

i. Set the training parameters:

Persistence: 5
Alpha: 0.9
Initial eta: 0.05
Stop tolerance: 0.02

ii. Build a network with two hidden layers, each with two neurons. Train the initial network as usual through one cycle.

iii. Create two copies of the initial network, a left and a right network. To the right network, add one neuron to the second hidden layer.

iv. Train both augmented networks through one cycle, and determine the overall error for each network, calculated as the sum of the across the j outputs and the p records in the cycle.

v. If the left network has lower error, keep it and add one neuron to the right network's first hidden layer.

vi. If the right network has lower error, replace the left network with a copy of the right network, and add a neuron to the second hidden layer of the right network.

vii. Train both networks through another cycle, and repeat the training/augmentation cycle until the stopping criteria are met.

Adjusting Eta

With the dynamic training method, changes to eta take the performance of the networks so far into account. At each cycle, two vectors are computed: movement, based on the changes to the weights over the cycle, M(t)= 2[W(t)-W(t-1)], where is the vector of weights at cycle t and is the vector of weights at the previous cycle, and change, based on the momentum at the current cycle, C(t)=0.8*C(t-1) + M(t).

The ratio of the magnitudes of these vectors, m (t) = (/ M (t)/) / (/ C(t)/)is an index of the acceleration of training. If the index is less than, training is slowing and eta is increased by a factor of 1.2. If the index is greater than 5.0, training is accelerating, and eta is decreased by a factor of 4/m (t).

Training the Final Network

After a good topology has been found, the final network is trained in the normal back-propagation manner, with the following settings:

Persistence: 5
Alpha: 0.9
Initial eta: 0.02
Stop tolerance: 0.005

DATA SET

The protein raw data set used in this paper is obtained from PDB. In the data set 538 protein enzymes taken from PDB are classified according to EC Number and Enzyme name. Six features, primary protein structures (Sequence), secondary protein structures (PSS), molecular weight, structural molecular weight (MW), chain length, atom count are extracted from PDB. Table 1 shows the proteins according to class and the counts used in training, testing and validation. Data preparation and all manipulations have been done using Microsoft Excel.

Table 1: data set description

EC No.	Class (Enzyme)	Function	Total Set	Train Set	Test Set	Valid-ation Set
1	Oxidoreductases	Catalyze the reduction-oxidationreactions.	81	56	11	14
2	Transferases	Transfer a functional groupingand a donor group to a receptor.	125	82	22	21
3	Hydrolases	Catalyze hydrolysis, the breakingof links and structures by theaction of water.	140	98	26	16
4	Lyases	Enzymes which catalyze thecleavage of C-C, C-O and C-Nlinks.	52	32	11	9
5	Isomerases	Catalyze the isomerizationreactions of simple molecules.	68	52	11	5
6	Ligases	Formation of links bycondensation of substances.	72	57	13	2
	Total		538	377	94	67

IMPLEMENTATION

For raw data manipulation Microsoft Excel is used. We implemented the SVM, C5, CHAID, ANN (Dynamic)Models using SPSS Clementine 11.1 computing environment. Firstly all the models were trained with training data set and then tested and validated by testing and validation datasets. Following are the descriptions of final architecture and parameter values models used for classification generated by computing tool (SPSS Clementine 11.1).Table 2 shows final models architecture and parameters value build using the tool.

Table 2:final implemented models architecture and parameters

Model	Architecture & parameters	Elapsed time for model build
SVM	Stopping criteria: 1.0E-3, Kernel type: RBF Regularization parameter (C): 10 Regression precision (epsilon): 0.1 RBF gamma: 0.1 Gamma: 1.0 Bias: 0.0 Degree: 3	2 secs
C5	Tree depth: 1 Pruning severity: 75 Minimum records per child branch:2 Expected noise(%): 0	< 1 secs
CHAID	Tree depth: 2 Levels below root: 5 Alpha for splitting: 0.05 Alpha for merging: 0.05 Chi-square for categorical targets:Pearson Epsilon for convergence:0.001 Max. Iterations for convergence: 100	>1 secs
ANN(Dynamic)	Input Layer: 1, 719 neurons Hidden Layer 1: 4 neurons Hidden Layer 2: 7 neurons Output Layer: 6 neurons normal back-propagation Persistence: 5 Alpha: 0.9 Initial eta: 0.02 Stop tolerance: 0.005	6 mins, 49 secs

RESULTS AND DISCUSSION

Model analysis of the models used in this paper shows input features and their variable importance (measure of impact of the feature on classification ranging [0, 1]) that was considered for the classification by model implementation algorithms. It is clear from table 3 that C5 took only one feature biased towards chain length only. CHAID took only two features chain length and molecular weight. SVM shows better unbiased skipping PSS and taking features primary structures (Sequence), secondary structures, structural molecular weight (MW), chain length, atom count, molecular weight. ANN (Dynamic) took all the features having rational impact of all the features on classification.

Table 3: features and their relative importance for themodels built

Model	C5		CHAID		SVM		ANN(DYNAMIC)	
	F	*RI*	*F*	*RI*	*F*	*RI*	*F*	*RI*
Feature (F) Relative Importance (RI) [0,1]	Chain Length	1	Chain Length	0.87	Chain Length	0.55	Chain Length	0.2
			Molecular Weight	0.13	ATOM Count	0.36	ATOM Count	0.19
					Molecular Weight	0.04	Structure MW	0.17
					Sequence	0.03	Molecular Weight	0.15
					Structure MW	0.03	Sequence	0.15
							PSS	0.14

The Training, testing and validation of the models was performed using the final architectures and parameters value shown in Table 2. The results in terms of overall accuracy for all classes are shown in Table 4 It is clear from the results obtained that the training accuracy is highest for SVM model i.e. 100%, having testing accuracy 84.04% and validation accuracy to be 86.57% better than C5 and CHAID. Although the training accuracy of ANN (Dynamic) model is less than SVM and is 92.04%, but the testing and validation accuracy is highest having 91.49% and 98.51% respectively.

Table 4: results obtained by models for the protein classes

Models	% Accuracy		
	Training Accuracy	Testing Accuracy	Validation Accuracy
SVM	100	84.04	86.57
C5	92.57	79.79	80.6
CHAID	93.37	77.66	83.58
ANN(Dynamic)	92.04	91.49	98.51

Table 5: results obtained by models for the indivisual protein classes

Model	Class/Enzyme	% Accuracy		
		Training Accuracy	Testing Accuracy	Validation Accuracy
SVM	EC1(Oxidoreductases)	100	81.81	85.71
	EC2(Transferases)	100	90.9	95.23
	EC3(Haydrolases)	100	96.15	93.75
	EC4(Lyases)	100	72.72	66.66
	EC5(Isomerases)	100	54.54	60
	EC6(Ligases)	100	86.61	100

C5	EC1(Oxidoreductases)	96.42	81.81	84.61
	EC2(Transferases)	92.68	90.9	80.95
	EC3(Haydrolases)	100	100	93.75
	EC4(Lyases)	93.75	72.72	77.77
	EC5(Isomerases)	84.61	54.54	60
	EC6(Ligases)	82.45	46.15	50
CHAID	EC1(Oxidoreductases)	96.42	81.81	84.61
	EC2(Transferases)	96.34	90.9	90.47
	EC3(Haydrolases)	97.95	96.15	93.75
	EC4(Lyases)	93.75	72.72	77.77
	EC5(Isomerases)	100	54.54	60
	EC6(Ligases)	71.93	38.46	50
ANN (Dynamic)	EC1(Oxidoreductases)	**100**	**90.9**	**100**
	EC2(Transferases)	**98.78**	**100**	**100**
	EC3(Haydrolases)	**97.95**	**96.15**	**100**
	EC4(Lyases)	**78.12**	**72.72**	**88.88**
	EC5(Isomerases)	**84.61**	**100**	**100**
	EC6(Ligases)	**78.94**	**76.92**	**100**

The results obtained by the models for individual protein classes are shown in Table 5. The above result shows thatalthough SVM is having highest training accuracy 100% for all the classes but the testing accuracy and validation accuracy of ANN(Dynamic model) is nearly better to all models. From the analysis of the result it is observed that testing and validation accuracy for Oxidoreductases, Transferages and Hydrolases is betterfor all models compared to that of Lyases, Isomerages andLigases but ANN(Dynamic) still performs better for these classes.

CONCLUSUION

ANN (Dynamic method) uses unbiased features with variable importance for classification of protein enzymes. Whereas other models used in this paper as well as other models uses lesser number of features with variable importance. Although building model with more features is complex andlacks in time complexity, but havingall features is more generalized. ANN(Dynamic) model elapsedvery much time for classification training by 377 protein instances i.e. more than 6 minutes while others took less than 2 secs, thus ANN(Dynamic) is least efficient in terms of time complexity. But it shows much better results in terms of classification accuracy compared to other models.

References

[1] L.Y. Han, C.Z. Cai, Z.L. Ji, Z.W. Cao, J. Cui, and Y.Z. Chen, "Predicting functional family of novel enzymes irrespective ofsequence similarity, " Nucleic Acids Research, vol. 32, pp. 6437-6444, 2004.

[2] C.Z. Cai, W.L. Wang, L.Z. Sun, and Y.Z. Chen, "Protein functionclassification via support vector machine approach, " MathematicalBiosciences, vol. 185, pp. 111-122, 2003.

[3] Paul D. Dobson and Andrew J. Doig, "Predicting Enzyme ClassfromProtein Structure without Alignments, " JMB, vol. 345, pp.187-199, 2005.

[4] Luiz C. Borro, Stanley R.M. Oliveira, Michel E.B. Yamagishi, AdaultoL. Mancini, Jose G. Jardine, Ivan Mazoni, Edgard H. dos Santos, Roberto H. Higa, Paula R. Kuser, and GoranNeshich, "Predictingenzyme class from protein structure using Bayesian classification, "Genetics and Molecular Research, vol. 5, pp. 193-202, 2006.

[5] Yong-Cui Wang, Yong Wang, Zhi-Xia Yang, Nai-Yang Deng.Support vector machine prediction of enzyme function withconjoint triadfeature and hierarchical context. BMC SystemsBiology, 5(Suppl 1):S6(2011). trabalhosrelacionados.

[6] Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

[7] Quinlan, J. (1996). Bagging, Boosting, and C4.5, Proceedings of theThirteenth National Conferenceon Artificial Intelligence, Portland, Oregon (American Association for Artificial Intelligence Press, Menlo Park, California), pp. 725 – 730.

[8] Rulequest Research. (2013) See5/c5.0.[Online]. Available:http://www.rulequest.com/see5-info.html.

[9] Kass, Gordon V.; An Exploratory Technique for Investigating Large Quantities of Categorical Data, Applied Statistics, Vol. 29, No. 2 (1980), pp. 119–127.

[10] "ANN Dynamic", pg. 1-8, Clementine® 11.1 Algorithms Guide, Copyright © 1995–2003 by International Business Machines Corporation and others Copyright © 2007 by Integral Solutions Limited.

[11] Heizmann, C. W., Fritz, G, Schäfer, B. W.: S100 Proteins: Structure, Functions and Pathology. Frontiers in Bioscience 7, d1356-1368 (2002).

[12] Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimalmargin classifiers. In Computational Learing Theory, páginas 144–152 (1992).

[13] Cortes, C. Vapnik, V. Support-vector networks. Machine Learning, 3(20):273–297 (1995).

[14] Burbidge, R., Buxton, B.: An introduction to support vector machinesfor data mining. In M. Sheppee (Ed.), Keynote Papers, Young OR12, University of Nottingham, páginas 3–15, .Operational Research Society:Operational Research Society, March(200