LETTER

Extraction and Optimization of Fuzzy Protein Sequences Classification Rules Using GRBF Neural Networks

Dianhui Wang*, Nung Kion Lee[+] , and Tharam S. Dillon*

*Department of Computer Science and Computer Engineering
La Trobe University, Melbourne, VIC 3083, Australia
E-mail: csdhwang@ieee.org

[+]Faculty of Cognitive Sciences and Human Development
University Malaysia Sarawak, Kota Samarahan, Sarawak, Malaysia
E-mail: nklee@fcs.unimas.my

(Submitted on May 14, 2003; Accepted on October 24, 2003)

Abstract—Traditionally, two protein sequences are classified into the same class if their feature patterns have high homology. These feature patterns were originally extracted by sequence alignment algorithms, which measure similarity between an unseen protein sequence and identified protein sequences. Neural network approaches, while reasonably accurate at classification, give no information about the relationship between the unseen case and the classified items that is useful to biologist. In contrast, in this paper we use a generalized radial basis function (GRBF) neural network architecture that generates fuzzy classification rules that could be used for further knowledge discovery. Our proposed techniques were evaluated using protein sequences with ten classes of super-families downloaded from a public domain database, and the results compared favorably with other standard machine learning techniques.

Keywords—Neural classification systems, data mining, rules extraction and optimization, generalized radial basis function networks, protein sequence

1. Introduction

A protein super-family consists of protein sequence members that are evolutionally related and therefore functionally and structurally relevant with each other [1]. One of the benefits from this category grouping is that some molecular analysis can be carried out within a particular super-family instead of individual protein sequence. It has also become apparent that the function of most genes is still unknown and classification into functionally related groups will provide valuable information on the protein function. Traditionally, two protein sequences are classified into the same class if they have high homology in terms of feature patterns extracted through sequence alignment algorithms. These algorithms, for instance, iPro-Class [4], SAM[5], MEME[6], compare an unseen protein sequence with all the identified protein sequences and provide a score based on similarity of sequences. As the size of the protein sequence databases is large, it is a very time consuming job to perform exhaustive comparison of existing protein sequences. Therefore, it is useful and helpful to build an intelligent classification system for effectively searching protein sequences in some large protein databases. Motivated by this, recently neural networks have been successfully applied in this domain and the results obtained demonstrate some merits of the methodology [1,2]. Neural networks have been chosen as technical tools for the protein sequence classification task due to the following two reasons: (i) the extracted features of protein sequences are distributed in a high dimensional space with complex characteristics which is difficult to satisfactorily model using some statistical or parameterized approaches; and (ii) neural networks are able to use the raw continuous values as system inputs. Basically, there are two types of neural models applicable for protein sequences classification task, i.e., unsupervised self-organizing mapping (SOM) networks [7] and supervised feed-forward neural networks (FNNs) [8,12]. The use of the SOM networks is to discover relationships within a set of protein sequences by clustering them into different groups. In contrast, the FNN based classification systems emphasizes on matching patterns through supervised learning. Once off-line training of the neural

network is accomplished, the resulting neural classifier is ready to be used for future protein sequence classification and only few seconds are needed to classify a new protein sequence. This saves a lot of time as compared to sequence alignment methods. Besides the direct protein classification, the supervised neural classifier could also been used to reduce the search scope of the sequence alignment program by only searching members of super-families [1].

The objective of this paper is to construct a generalized RBF network, which generates a set of fuzzy rules [3], for protein sequence classification tasks. The rest of the paper is organized as follows: Section 2 discusses some issues on rule representation and extraction using GRBF neural networks. Section 3 presents a novel objective function for rule set optimization. Section 4 evaluates the performance of the proposed intelligent protein sequence classification system, where a data preprocessing description and a comparison are given. We conclude this paper in the last section.

2. Rules Representation and Extraction

Let the classification task contain n-class data sets, denoted by $\Theta_1, \Theta_2, \ldots, \Theta_n$, characterized by continuous attributes in the subset $[0,1]^m \subset R^m$. The data sets are divided into a training set denoted by S_{tra} and a test set denoted by S_{tes} , respectively. The training set will be used to initialize and optimize the rules, and the test set, unseen by the learning methods, will be used to stop the refinement process and evaluate the performance of the final neuro-fuzzy classifier.

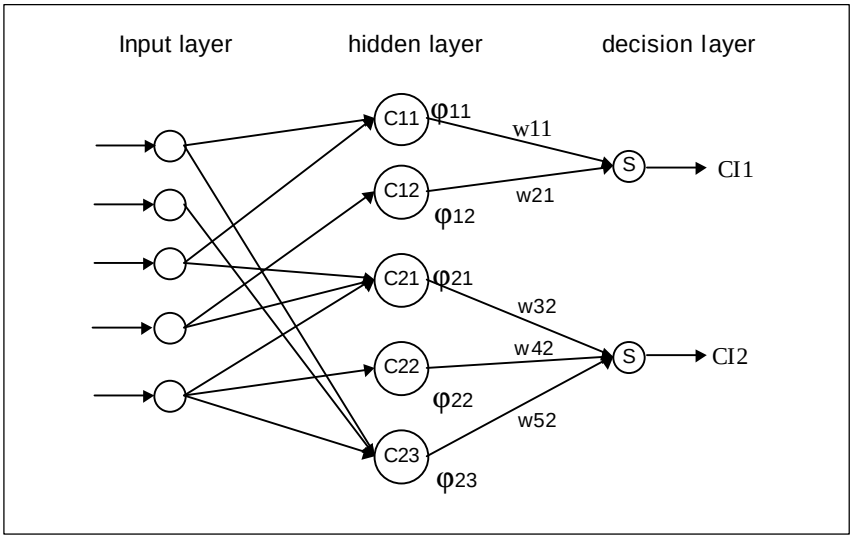

Figure 1. A typical GRBF network architecture

The network consists of *m* input features $X = [x_1, x_2, \ldots, x_m]^T$, M hidden units and *n* output units at the decision layer. The activation functions j in the hidden units are the Gauassian functions defined by

$$j(X_j) = \exp[-d(X_j, C_j)], \tag{1}$$

where $X_j = [x_{j1}, x_{j2}, \ldots, x_{jp}]^T$, $jp \le m$, represents a subset or a projection of X onto a subspace of the feature space, which is the contributory input vector to the j-th hidden unit, C_j the corresponding cluster center of the unit, $d(X_j, C_j)$ represents the weighted Euclidean distance measure:

$$d(X_j, C_j) = \sum_{k=1}^{p} (x_{jk} - c_{jk})^2 / s_{jk}^2 . \tag{2}$$

A fuzzy T-norm operator, namely, fuzzy plus operator $\oplus$, defined by

$$a \oplus b = a + b - ab , \tag{3}$$

is applied as the activation function at the output layer of the GRBF network. Therefore, each unit within a group in the hidden layer will contribute some certain classes disjunctively for classification decision-making. The network outputs, as classification indicator (*CI*), are given by

$$CI_P(X) = \sum_{k=1}^{N_p} (-1)^{k+1} \sum_{j_1 < \cdots j_k} a_{Pj_1} \cdots a_{Pj_k} , \tag{4}$$

where $CI_p(X)$ represents the p-th output of the network for a given feature input *X*, N_p is the number of the neurons in the hidden layer connected with the p-th output, and a_{Pj} is given by

$$a_{Pj} = W_{jP} \exp(-d(X_{Pj}, C_{Pj})) , \tag{5}$$

which can be interpreted as a firing strength of the local fuzzy classification rules with confidence factor W_{jP} .

The classification criterion used in this paper follows the maximum component principle, that is, a given pattern X will be assigned to Class *Q* as

$$CI_Q(X) = \arg\max_P \{CI_P(X)\} . \tag{6}$$

A set of fuzzy rules for protein sequence classification can be directly extracted from the GRBF network described above. The following is a typical fuzzy rule:

$$R_{Pj}: \text{ IF } X_{Pj} \text{ is around } C_{Pj} \text{ THEN } X \in \Theta_P \text{ with } CF = W_{jP} \tag{7}$$

where *CF* represents a confidence factor of the fuzzy classification rule.

There are two core steps to build up the neural classifier, that is, the network structure determination and model parameters optimization. In the rest of this section, we discuss the first issue briefly and the second issue will be examined in the next section.

Structure identification contains two parts, i.e., initialization of a group of hidden units for each classes using supervised Expectation Maximization (EM) algorithm [10], and the input features selection for the hidden units. We omit the detailed description about the well-known EM algorithm and only outline a Feature Subset Selection (FSUBS) technique, which, in fact, is a modified version of the PROCLUS (Projected Clustering) [11] clustering subspace feature selection approach. Briefly, the FSUBS algorithm finds a partition of the points into clusters so that the points within each cluster are close to one another. The cost function for evaluating FSUBS is the classification rate (CR) of the training data. The CR of the training data is calculated after each removal of a feature from a cluster. The FSUBS algorithm differs from the PROCLUS feature selection in several aspects. In PROCLUS, the evaluation on the objective function is performed after selecting feature subsets for all clusters. In the FSUBS, the evaluation is done just after a feature is removed from a cluster. The reason for doing this is that the initial clusters are not formed by the training data sets. The FSUBS tries to reduce the misclassification by removing some features. The main objective of finding subspace feature using the PROCLUS is to increase the closeness between points assigned to a cluster. On the other hand, the objective of the FSUBS is to reduce overlap between clusters from different classes. As a result, the input features associated with each individual neuron in the hidden layer can be selected successively. To discard permanently a feature for a cluster, it is necessary to check out the CR value to ensure that the performance will not decrease largely.

3. Rule Sets Optimization

Rule sets optimization here refers to parameters refinement of the GRBF networks. The key for doing this is to define an objective function, which may characterizes the performance of the classifier. The commonly concerned performance index for neural classifiers is the misclassification rate (MR) for both training and test data sets. The MR for the training data set can be measured by

$$M_p(X) = \frac{\max_{q \neq p} \{CI_q(X)\}}{CI_p(X)} , X \in \Theta_p \bigcap S_{tra} . \tag{8}$$

However, the expression above does not really demonstrate the Generalization Capability (GC) of the neural classification system, or equivalently, the MR index for the test data set. In order to embed this significant information into an objective function for the parameter tuning purpose, some indirect methods will be helpful and necessary although there is still a lack of rigorous theoretical basis. In this paper, we use the following mathematical formula to express this idea:

$$G_p(X) = CI_p(X) + \sum_{j=1}^{N_p} \sum_{k=1}^{p} s_{jk}^2 , \ X \in \Theta_p \bigcap S_{tra} \tag{9}$$

The first term of the right hand side in (9) implies the CI quality or the reliability of the neural classifier, and the second term is associated with a geometric size constraint. As the ellipsoidal region tends to be large

enough to enclose many training patterns for a certain category of data, the misclassification rate could be increased for this class. It could lead to lower recognition for other classes. On the other hand, if the size of region is reduced and only a small amount of training data for a class is enclosed, a good generalization ability of the rule cannot be expected because of over-fitting, also the recognition rate of this class could be reduced due to poor coverage. Thus, a larger value of this term implies a higher recognition possibility for unseen patterns, which have similar nature to the examples from class p. The higher the value $\sum_{X\in\Theta_p\cap S_{tra}} G_p(X)$ takes, the better the GC performance should be. Therefore, a tradeoff between the size of the ellipsoidal region to achieve good GC power and a low misclassification rate should be addressed. Finally, we define an objective function to refine the classification system as follows:

$$y_p(X) = (1-l_p)M_p(X) + l_p[G_p(X)]^{-1}, \tag{10}$$

where $0 < l_p < 1$ is a regularizing factor.

An overall cost function for optimizing the GRBF neural classifier is defined by

$$y(l) = D_{tra}\sum_{p=1}^{n}\sum_{X\in(\Theta_p\cap S_{tra})} y_p(X), \tag{11}$$

where

$$D_{tra} = \frac{1}{n}\sum_{k=1}^{n} c_k(X) \neq \hat{c}_k(X), \tag{12}$$

and $c(X)$ and $\hat{c}(X)$ represent real class and predicted class for pattern X from the training data set, respectively.

Given a set of regularizing factors, minimization of the cost function (11) will result in a better classifier with improved performance. It must be mentioned that the regularizing factors allow us to balance the importance of misclassification against the generalization ability for each individual class. This importance comes from users and it is usually quite subjective. Within our best knowledge, so far, there is no better way to express the subjective nature and to assign the values of these regularizing factors satisfactorily. In this paper, all the regularizing factors take value of 0.5.

4. Performance Evaluation

A protein sequence is made from combinations of variable length of 20 amino acids ∑ = {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. The n-grams or k-tuples [8] features will be extracted as an input vector of the neural network classifier. The n-gram features are a pair of values (v_i, c_i), where v_i is the feature i and c_i is the counts of this feature in a protein sequence for i = 1… 20^n. In general, a feature is the number of occurrences of an animal in a protein sequence. These features are all the possible combinations of n letters from the set ∑. For example, the 2-gram (400 in total) features are (AA, AC,…,AY, CA, CC,…,CY,…,YA, …,YY). Consider a protein sequence VAAGTVAGT, the extracted 2-gram features are {(VA, 2), (AA, 1), (AG, 2), (GT, 2), (TV, 1)}. The 6-letter exchange group is another commonly used piece of information. The 6-letter group actually contains 6 combinations of the letters from the set ∑. These combinations are A={H,R,K}, B={D,E,N,Q}, C={C}, D={S,T,P,A,G}, E={M,I,L,V} and F={F,Y,W}. For example, the protein sequence VAAGTVAGT mentioned above will be transformed using 6-letter exchange group as EDDDDEDDD and their 2-gram features are {(DE, 1), (ED, 2), (DD, 5)}. We will use e_n and a_n to represent *n*-gram features from a 6-letter group and 20 letters set. Each sets of n-grams features, i.e., e_n and a_n, from a protein sequence will be scaled separately to avoid skew in the counts value using equation (13) below:

$$\bar{x} = \frac{x}{L-n+1}, \tag{13}$$

where x represents the count of generic gram feature, $\bar{x}$ is the normalized x, which will be the inputs of the neural networks; L is the length of the protein sequence and n is the size of n-gram features.

In this study, the protein sequences covering ten super-families (classes) were obtained from the PIR databases comprised by PIR1 and PIR2 [4]. The 949 protein sequences selected from PIR1 were used as the training data and the 533 protein sequences selected from PIR2 as the test data. The ten super-familes to be trained/classified in this study are: Cytochrome c (113/17), Cytochrome c6 (45/14), Cytochrome b (73/100), Cytochrome b5 (11/14), Triose-phosphate isomerase (14/44), Plastocyanin (42/56), Photosystem II D2 protein (30/45), Ferredoxin (65/33), Globin (548/204), and Cytochrome b6-f complex 4.2K(8/6). The 56 features were extracted and comprised by e_2 and a_1. Table 1 gives the setup of our experiment. A standard GA algorithm

with parameter constraint is used to minimize the objective function (11). The GA population size and the maximum generation step are set as 30 and 1000, respectively, in our simulation studies. The crossover and mutation probability may be adjusted in different simulations to find the most suitable ones. In EXPR1, the mutation probability takes 0.01, the cross over probability is 0.95 and the replacement probability is 0.95. In EXPR3, the crossover probability is 0.90, the mutation probability is 0.01, and replacement probability is 0.85, respectively. The optimization procedure terminates when the maximum generation index is met. The best solution with minimum objective value is recorded during optimization. There are 43 and 48 clusters produced by the EM clustering algorithm, for EXPR1 and EXPR3, respectively. Figure 2 depicts the numbers of subset features of every cluster in EXPR 3.

Table 1: Experiment Setup

EXP#	Experiments Description
EXPR1	The Principal Components Analysis (PCA) feature selection algorithm is applied to the e2, a1 features. The EM clustering algorithm is then applied to the training data with the selected feature subset. The GRBF model is then constructed and further optimized using the GA algorithm.
EXPR2	The FSUBS algorithm is applied to the clusters using the features selected in EXPR1. Then the optimization using GA algorithm is carried out.
EXPR3	The FSUBS algorithm is applied directly to the 56 features. The GRBF model is then optimized using GA algorithm.

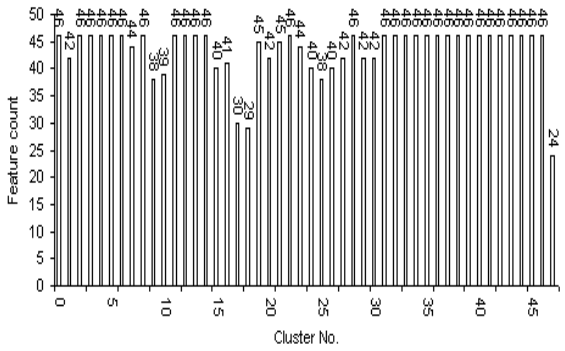

Figure 2. Features selected in each cluster for EXPR3

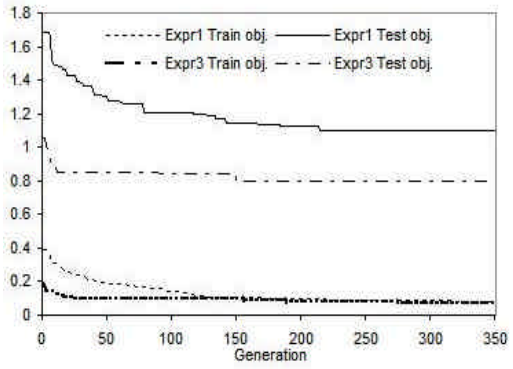

Figure 3. GA optimization progress

Table 2: Performance Evaluation

EXP#	before optimization		after optimization	
	Training	Test	Training	Test
EXPR1	93.57%	76.74%	96.94%	90.62%
EXPR2	96.84%	85.74%	96.94%	88.18%
EXPR3	95.04%	81.24%	97.37%	92.68%

Table 3: Performance Comparison

Classifiers	Training	test
MLP-CE	99.16%	90.81%
MLP-MSE	99.37%	91.37%
RBF-CE	97.15%	90.99%
RBF-MSE	99.26%	91.56%
MRBF-CE	98.74%	89.31%
MRBF-MSE	99.89%	87.62%
C4.5	98.40%	79.74%

In EXPR1, a PCA feature selection algorithm with 90% component rate is firstly applied to the e2 and a1 features respectively before using the EM algorithm. Then, the GA optimization process is applied to a GRBF network. To investigate whether the performance can be further improved from the new features, in EXPR2, the subspace feature selection method is applied to the 43 new features selected in EXPR1. The results observed show that the FSUBS can slightly improve the performance but not too much. Applying the FSUBS algorithm in EXPR3, an average of 43 features is selected, which reduces about 23.21% in feature number. It has been seen that less features are associated to the classes that have fewer number of clusters. The initial CR after the subspace feature selection is 95.04% (902) for the training data set and 81.24% (433) for the test data set, respectively. After refinement, the CR reaches 97.37% (924) for the training data set and 92.68% (494) for the

test data set, respectively. This performance is the best one among all the experiments, but it required more GA generation as compared to the other two experiments. This is because the search space in EXPR3 is a little larger than other two cases. Table 2 demonstrates the performance of our presented neural classifier.

Figure 3 above shows the working progress of GA optimization for experiments EXPR1 and EXPR3 within the first 350 generations. It can be seen that the objective function values for both training and test data in EXPR3 are smaller than that in EXPR1. This indicates that the selected feature subsets using FSUBS in EXPR3 are more efficient to achieve the tradeoff objectives described in the objective function. The objective values for both data are decreasing quickly in the first 100 generations and slowly in the following generation. Table 3 gives the recognition rate of the standard RBF network, the MLP neural classifiers, Modular RBF (MRBF) network classifier with mean square error (MSE) and cross entropy (CE) learning criteria [12], and the well-known decision tree classifier C4.5 [9]. The C4.5 result is obtained using 10 trials and then average them to get to the final result. The best performance is 92.68% reached by the proposed method in EXPR3 as compared to the best performance of 91.37% from the BP-MSE neural classifier. The C4.5 has the worst performance for this classification task.

5. Conclusion

Building improved intelligent protein sequence classification systems for effectively searching large biological database is significant for developing competitive pharmacological products. This paper describes a methodology for constructing a neural protein classifier with various input features, rather than to train a neural classifier based on a given neural network architecture and some available data. A set of fuzzy classification rules with confidence factors can be extracted directly from the GRBF networks. The initial fuzzy rule set is refined using a new objective function, which compromises between misclassification rate and generalization capability, and GA programming. Experimental and comparative results demonstrate that our proposed method outperforms other neural classifiers for this data set.

The following issues will be studied in our further work: (i) Learning strategies comparison; (ii) System quality improvement, for instance, increasing the recognition rate for super-families with low-probability items, reliability and robustness.

References

[1] P. Baldi and S. Brunak, Bioinformatics: The Machine Learning Approach, The MIT Press, 2001.
[2] C. M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.
[3] K. Hirota and W. Pedrycz, “Fuzzy computing for data mining”, Proc. of the IEEE, pp.1575 –1600, 1999.
[4] Protein Information Resources (PIR), http:// pir.Georgetown.edu
[5] SAM: Sequence Alignment and Modeling Software System, Baskin Center for Computer Engineering and Science, http://www.cse.ucsc.edu/researchcompbio/
[6] MEME: Multiple EM for Motif Elicitation UCSD Computer Science and Engineeringhttp://meme.sdsc.edu
[7] H. C Wang, J. Dapazo, L. G. De La Fraga, Y. P. Zhu, J. M. Carazo, “Self-organizing tree-growing network for the classification of protein sequences”, Protein Science, pp. 2613-2622, 1998.
[8] C. H. Wu, G. Whitson, J. McLarty, A. Ermongkonchai, T. C. Change, “PROCANS: Protein classification artificial neural system”, Protein Science, pp. 667-677, 1992.
[9] J. R. Quilan, C4.5: programs for machine learning, San Mateo, CA Morgan Kaufmann, 1994.
[10] A. P. Dempster, N. M. Laird and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm”, J. Royal Statiscal Soc., Serial *B*. Vol. 39, No. 1, pp 1-38, 1977.
[11] C. C. Aggarwal, C. Procopiuc, J. L. Wolf, P. S. Yu and J. S. Park, Fast algorithms for projected clustering. In SIGMOD'99, Philadephia, *PA*. June, 1999.
[12] D. H. Wang, N. K. Lee, T. S. Dillon and N. J. Hoogenraad, “Protein sequences classification using Radial Basis Function (RBF) neural networks”, R.I. McKay, J. Slaney (Eds.): AI 2002: Advances in Artificial Intelligence, Lecture Notes in Computer Science, LNAI 2557, pp. 477-486, Springer 2002.

Dianhui Wang received his PhD degree from Northeastern University, China in March 1995, and currently a lecturer in the Department of Computer Science and Computer Engineering at La Trobe University, Melbourne, Australia. His research interest includes data mining fundamental, neural nets applications, soft information retrieval systems, and singular systems theory. (Home page: http://homepage.cs.latrobe.edu.au/dhwang/)

Nung Kion Lee graduated from La Trobe University in 2002, and currently a lecturer in Faculty of Cognitive Sciences and Human Development, University Malaysia Sarawak, Malaysia.

Tharam S. Dillon received his PhD degree from Monash University and currently a professor of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia.