Iranian Journal of Fuzzy Systems **Vol. 5, No. 2,** (2008) pp. 21-33

GENERATING FUZZY RULES FOR PROTEIN CLASSIFICATION

E. G. MANSOORI, M. J. ZOLGHADRI, S. D. KATEBI, H. MOHABATKAR,
R. BOOSTANI AND M. H. SADREDDINI

ABSTRACT. This paper considers the generation of some interpretable fuzzy rules for assigning an amino acid sequence into the appropriate protein superfamily. Since the main objective of this classifier is the interpretability of rules, we have used the distribution of amino acids in the sequences of proteins as features. These features are the occurrence probabilities of six exchange groups in the sequences. To generate the fuzzy rules, we have used some modified versions of a common approach. The generated rules are simple and understandable, especially for biologists. To evaluate our fuzzy classifiers, we have used four protein superfamilies from UniProt database. Experimental results show the comprehensibility of generated fuzzy rules with comparable classification accuracy.

1. Introduction

Bioinformatics[4]is basically conceptualizing biology in terms of macromolecules and applying informatics techniques to understand and organize the information associated with these molecules. It deals primarily with the application of computer and statistical techniques to the management of biological information. Because of the Human Genome Project and other similar efforts, a large number of biological data are regularly collected. It is important to organize and annotate this massive amount of sequential data to maximize its utility. In this regard, DNA sequences are translated into protein sequences using standard bioinformatics tools. Among these is protein sequence classification, which determines the type or group of proteins to which an unknown protein sequence belongs. One of the benefits from this type of category grouping is that molecular analysis can be carried out within a particular superfamily instead of an individual protein sequence. A protein superfamily consists of protein sequence members that are evolutionally related and therefore functionally and structurally relevant to each other.

Several approaches dealing with the protein classification problem have been proposed in the past. These include alignment of protein sequences [2], hidden Markov modeling [14], application of artificial neural networks [23, 24, 25], using

Received: June 2007; Revised: August 2007; Accepted: September 2007
Key words and phrases: Amino acid sequence, Protein classification, Fuzzy rule-based classifier.

support vector machines [12, 13] and extreme learning machines [22]. In addition to these are approaches based on principal component analysis [6] and a combination of fuzzy clustering with nearest neighbor classifiers [3]. Though all these methods have high classification accuracy, they suffer from an important limitation from the point of view of biologists; The classifiers cannot not be easily interpreted and the results are often in a black box.

A fuzzy rule-based classification system is a special case of fuzzy modeling where the output of the system is crisp. The main advantage of this classifier is the interpretability of the model and its greater comprehensibility [17]. We have designed a fuzzy rule-based system for classification of protein sequences to make the best use of this advantage. Using the distribution of amino acids in each protein sequence in a set of training examples, our approach generates fuzzy classification rules which are simple and easily comprehensible, especially for biologists who can utilize them for classifying new proteins in more readable manner.

The subsequent sections are organized as follows. Section 2, deals with the extraction of features from protein sequences. Section 3 describes our approach for designing fuzzy rule-based classifiers from data. In section 4, experimental results on some real-world protein sequences are presented. Section 5 concludes the paper.

2. **Extraction of Features from Protein Sequences**

A protein sequence contains characters from the 20-letter amino acid alphabet Σ = {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. An important issue in applying any algorithm to protein sequence classification is the encoding of the protein sequences. For this purpose we have used the encoding technique in [24] that entails the extraction of high-level features from protein sequences by counting the number of occurrences of each amino acid in the sequence. For instance, given a protein sequence PVKVPTKPKV, this encoding method gives the following result: 3 for P (indicating P occurs thrice), 3 for V, 3 for K, and 1 for T.

Exchange groups are another commonly used piece of information. There are six groups of amino acids which represent high evolutionary similarity. The 6-letter exchange groups are {e_1, e_2, e_3, e_4, e_5, e_6} [24] where e_1={H, R, K}, e_2={D, E, N, Q}, e_3={C}, e_4={S, T, P, A, G}, e_5={M, I, L, V} and e_6={F, Y, W}. Exchange groups are effective equivalence classes of amino acids and are derived from PAM [5]; for example, the protein sequence PVKVPTKPKV is represented by $e_4e_5e_1e_5e_4e_4e_1e_4e_1e_5$ and hence the exchange group encoding for this sequence is: 4 for e_4, 3 for e_5, and 3 for e_1.

Using the amino acid encoding scheme, the problem of protein classification would have 20 features whereas it is 6-dimensional if the exchange group encoding

technique is used. Because of the great amount of computation required in the approach used for rule generation in this paper, we have only counted the occurrences of exchange groups in the sequences to extract features. Also, to avoid skewness in the counts value, each feature value c is scaled to a probability estimate p by:

$$p = \frac{c}{l} \tag{1}$$

where l is the length of protein sequence. Hence for the sequence PVKVPTKPKV, we obtain the feature vector {(e_1, 0.3), (e_2, 0.0), (e_3, 0.0), (e_4, 0.4), (e_5, 0.3), (e_6, 0.0)}.

3. **Designing a Fuzzy Rule-based Classifier**

Fuzzy if-then rules for a pattern classification problem with n attributes can be written as:

$$\text{Rule } R_j\text{: If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \text{ then class } C_j\text{, for } j=1, 2, \ldots, N \tag{2}$$

where $X=[x_1, x_2, \ldots, x_n]$ is an n-dimensional pattern vector, A_{ji} (i=1, 2,…, n) is an antecedent linguistic value of R_j such as ***Small*** or ***Large***, C_j is the consequent class, and N is the number of fuzzy rules. Generally, m labeled patterns $X_p=[x_{p1}, x_{p2}, \ldots, x_{pn}]$, p=1, 2, …, m are given for an M-class problem. Generally, each attribute is first normalized to fall within the unit interval [0, 1]. Using the information provided by labeled patterns, the task of a classifier design is to generate a set of fuzzy rules of the form (2).

For this purpose, the pattern space is first partitioned into fuzzy subspaces and then, if there are some patterns in a subspace, it is identified by a fuzzy rule [9]. To avoid an explosion of rules in problems of high dimension, several approaches have employed rule evaluation criteria to select a small subset of rules among the larger set of candidate rules [10]. Assuming that the partitioning of the pattern space is provided in advance, triangular membership functions are commonly used to partition each input attribute into K fuzzy subsets. Membership functions of this type are simpler and more comprehensible than others. Moreover under certain assumptions, the fuzzy partitions built out of the triangular membership functions lead to entropy equalization [18]. Figure 1 shows these membership functions for four different values of K. For K=3, the linguistic labels A_3, A_4 and A_5 can be interpreted as linguistic values ***Small***^{3}, ***Medium***^{3} and ***Large***^{3} respectively where superscript 3 shows the value of K.

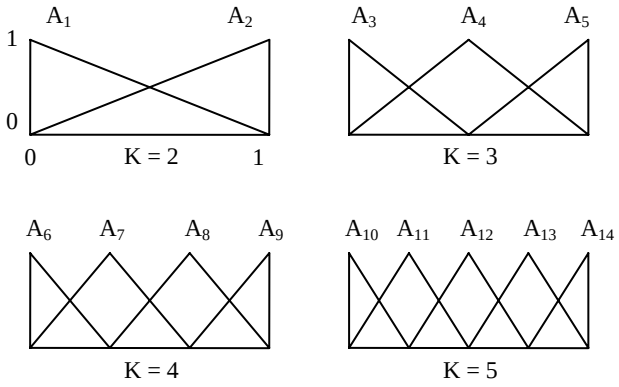


FIGURE 1. Different partitioning of each input attribute.

The consequent class C_j of a fuzzy rule R_j in (2) is determined by training patterns in the corresponding fuzzy subspace. The grade of compatibility of this rule with a training pattern X_p is defined by the antecedent part $A_j = A_{j1} \times A_{j2} \times \ldots \times A_{jn}$ of rule R_j using the product operator

$$\mu_j(X_p) = \prod_{i=1}^{n} \mu_{ji}(x_{pi}) \tag{3}$$

where $\mu_{ji}(.)$ is the membership function of the antecedent fuzzy set A_{ji}. To select the consequence class of a fuzzy rule, we have used the heuristic method proposed by Ishibuchi et al. [11], which is based on the confidence of association rules from the field of data mining. A fuzzy classification rule in (2) can be viewed as an association rule of the form $A_j \Rightarrow Class\ C_j$, where A_j is a multidimensional fuzzy set representing the antecedent combination of the rule and C_j is a class label. In [11], a measure for evaluating the confidence of a fuzzy association rule is provided by:

$$Conf(A_j \Rightarrow Class\ C_j) = \frac{\sum_{X_p \in Class\ C_j} \mu_j(X_p)}{\sum_{p=1}^{m} \mu_j(X_p)} \tag{4}$$

The consequent class C_j of fuzzy rule R_j is specified by identifying the class with maximum confidence.

The most popular fuzzy reasoning method in fuzzy rule-based classification systems, which is also simple and intuitive for human users, is the reasoning based on a single winner rule [8]. In this method, a new pattern $X_t=[x_{t1}, x_{t2}, \ldots, x_{tn}]$ is classified by the single winner rule $R_{\hat{j}}$ defined as:

$$\mu_{\hat{j}}(X_t) = \max\{\mu_j(X_t), j = 1, 2, ..., N\} \quad (5)$$

where $\mu_j(X_t)$ is the compatibility grade (3) of fuzzy rule R_j with X_t . In other words, among the rules fired by X_t, the winner is one with the highest compatibility grade. Note that the classification of a pattern not covered by any rule in the rule base is rejected. This also occurs for a pattern X_t when the maximum value of $\mu(X_t)$ corresponds to two rules with different consequent classes.

Since in fuzzy rule-based classifiers, the main objective is the interpretability of the system, the classification accuracy is not too high. Though it is possible to use weighted fuzzy rules [11, 16, 26] or weighting functions [15] to achieve higher accuracy, in order to maintain the comprehensibility of rules, in this paper we have used fuzzy rules with no weights.

3.1. Rule Evaluation Criteria. In the field of data mining, the support of association rules have been often used as rule selection criteria [1]. The fuzzy version of support (s_F) for the fuzzy classification rule $A_j \Rightarrow Class\ C_j$ is defined in [11] by

$$s_F(A_j \Rightarrow Class\, C_j) = \frac{1}{m} \sum_{X_p \in Class\, C_j} \mu_j(X_p) \quad (6)$$

where m is the number of given training patterns. The crisp version of support (s_C) for such rule is

$$s_C(A_j \Rightarrow Class\, C_j) = \frac{\eta_j}{m} \quad (7)$$

where η_j is the number of patterns truly classified by rule R_j as class C_j (i.e. the number of true positives).

In [7], a heuristic rule selection criterion based on the support is used for extracting fuzzy rules from numerical data. This simple criterion is specified by

$$f_C(A_j \Rightarrow Class\, C_j) = \eta_j - \overline{\eta}_j \quad (8)$$

where $\overline{\eta}_j$ is the number of patterns incorrectly classified by rule R_j as class C_j (i.e. the number of false positives). This evaluation measure can be fuzzified as follows:

$$f_F(A_j \Rightarrow Class\, C_j) = \sum_{X_p \in Class\, C_j} \mu_j(X_p) - \sum_{X_p \notin Class\, C_j} \mu_j(X_p) \ . \quad (9)$$

Dividing the right-hand sides of (8) and (9) by m, these measures can be expressed by means of the crisp and fuzzy versions of the support. However, since the number of training patterns in (6) and (7) is equal for all fuzzy rules, it is not considered.

In this paper, we have used four distinct criteria for evaluating candidate rules. The first criterion, following [7], uses the measures (9). In second criterion, we prefer the crisp measure (8) to the fuzzy measure (9), because in classification problems, the number of patterns truly classified by a rule is more important than the summation of their compatibility grade (e.g. the compatibility grade summation of two positive patterns might be less than the compatibility grade of one negative pattern). However, in the situation of a tie, the measure (9) is also considered.

The third and fourth criteria use the same formulas as the first and second respectively, but they only consider training patterns in the decision subspace of each rule instead of its covering subspace. The patterns in the covering subspace of a rule have compatibility grade above zero (i.e. will cause the rule to be fired) whereas the patterns in the decision subspace will be classified by this rule. However, since it is not possible to identify the decision subspace of a rule precisely in the rule generation phase, we have heuristically specified a threshold (τ_j) for rule R_j and the patterns with compatibility grade above this threshold are assumed to be in its decision subspace. Since the membership functions in Fig. 1 are 0.5-complete, we have defined $\tau_j = 0.5^{k_j}$ where k_j is the length of rule R_j.

3.2. Rule Generation. Given an input partitioning of the pattern space, one approach for rule generation is to consider all possible combinations of antecedent linguistic values and generate a fuzzy rule for each combination if there is at least one training pattern covered by this rule. For higher dimensional problems, this approach can generate too many rules which are practically impossible to handle. For example, for a data set having n input variables and K fuzzy sets for each variable, K^n fuzzy rules may be generated by this method. One way to tackle this problem is to use a fuzzy rule evaluation measure which selects a small subset of rules from the large set of candidate rules [10].

This approach considers different partitions in Fig. 1 for each attribute simultaneously. For example, suppose one of 14 fuzzy sets can be used for each attribute when generating a candidate fuzzy rule. then, for an n-dimensional problem, there are 14^n antecedent combinations to consider and this is quite impractical.

One solution for the above problem, which adds the fuzzy set "don't care" to each attribute, is presented in [10]. The membership function of this fuzzy set is defined as $\mu_{don't\ care}(x) = 1$ for all values of x. The trick is that instead of considering all antecedent combinations (15^6 = 11 390 625 in this example), we only consider short fuzzy rules having a limited number of antecedent conditions (excluding

"don't care") as candidate rules. In our example, if we use at most three antecedent variables, we will have $15^3 = 3375$ candidate rules.

In our method, the generated candidate rules are divided into M groups according to their consequent class. The rules in each group are sorted in descending order of their evaluation measures (e.g. one of the four criteria mentioned in subsection 3.1). For a pre-specified number of Q, an initial rule base is constructed by choosing the best Q rules from each class (at most $N=M\times Q$ rules). When multiple rules have the same evaluation measure, the rule having fewer antecedent conditions (the simplest rule) is selected. In the situation of a tie, the rule having the larger subspace is preferred. Finally, when multiple rules have the same value for all the above criteria, we randomly select a rule from the set of best rules. The following algorithm outlines our approach briefly:

Algorithm: Generating fuzzy classification rules from data.

Input: m labeled patterns of an n-dimensional M-class problem and Q.

Output: $M\times Q$ rules.

1. Generate all fuzzy rules having three or less antecedent variables as candidates (at most 15^3 rules).
2. Determine the consequent class of each candidate rule.
3. Divide the candidate rules into M groups according to their consequent class.
4. Rank the candidate rules in each group in descending order of their evaluation measures.
5. Choose the best Q rules from each class as the initial rule base (at most $M\times Q$ rules).

We propose a heuristic approach for considering the cooperation between rules to increase the classifier accuracy; this extracts more cooperative rules from the initial rule base and establishes a compact rule base with a good cooperation among rules. First, the "best" rule for each class according to evaluation measures of rules is selected. To select the second best rules, training patterns are classified by each of the remaining rules and, again, only those that improve the classification accuracy are selected. This algorithm continues for classes that add more rules to the rule base.

4. **Experimental Results**

The data used in the experiments are obtained from the Universal Protein Resource (UniProt), release 11.0, maintained by the UniProt Consortium [21]. The UniProt Knowledgebase, accessible at http://www.uniprot.org has 2 299 834 sequences based on Release 6.0 (September 2005). In this paper, only the sequence, sequence type and superfamily of the entries are used. Table 1 illustrates the specification of four superfamily classes used in the experiments.

Superfamily name	Number of sequences	Minimal length of sequences	Maximal length of sequences
Globin	492	29	351
Kinase	543	59	493
Ras	391	96	339
Trypsin	503	48	362

TABLE 1. Data used in the experiments.

To illustrate the interpretability of the generated rules, we have run our algorithm using the whole data as training patterns for constructing the rule base. These data are also used for testing the classifier. The error rates are computed as the number of misclassified patterns divided by total number of tested samples and expressed as percentages. The following are the best rules per class of the initial rule base, where $p(.)$ is the probability estimate (1) and $A_1, A_2, \ldots, A_{14}$ are the linguistic labels in Fig. 1 interpreted as $Small^2, Large^2, \ldots, Large^5$ respectively. Using the first evaluation criterion, our rules have error rate = 32.92 percent:

If $p(e_3)$ is A_{10} then Globin.

If $p(e_1)$ is A_8 and $p(e_6)$ is A_{13} then Kinase.

If $p(e_2)$ is A_8 then Ras.

If $p(e_1)$ is A_{11} then Trypsin.

Translating these rules to linguistic statements will reveal their interpretability and readability to biologists. For incoming protein sequence X with length l, these rules can be expressed as:

If the number of the amino acid C in X divided by l is $Small^5$ then

X belongs to superfamily *Globin*.

If the number of the amino acids H, R and K in X divided by l is $MediumLarge^4$ and

the number of the amino acids F, Y and W in X divided by l is $MediumLarge^5$ then

X belongs to superfamily *Kinase*.

If the number of the amino acids D, E, N and Q in X divided by l is $MediumLarge^4$ then

X belongs to superfamily ***Ras***.

If the number of the amino acids H, R and K in *X* divided by *l* is ***MediumSmall***^{5} then

X belongs to superfamily ***Trypsin***.

The following rules are obtained using the second criterion (error rate = 30.95%):

If $p(e_2)$ is A_6 and $p(e_3)$ is A_{10} then ***Globin***.

If $p(e_2)$ is A_{11} and $p(e_4)$ is A_6 and $p(e_6)$ is A_5 then ***Kinase***.

If $p(e_2)$ is A_8 and $p(e_4)$ is A_6 and $p(e_6)$ is A_{11} then ***Ras***.

If $p(e_1)$ is A_7 and $p(e_3)$ is A_8 and $p(e_4)$ is A_8 then ***Trypsin***.

The rules corresponding to the third criterion (error rate = 29.45%) are:

If $p(e_3)$ is A_{10} then ***Globin***.

If $p(e_3)$ is A_{11} and $p(e_6)$ is A_8 then ***Kinase***.

If $p(e_2)$ is A_8 and $p(e_6)$ is A_7 then ***Ras***.

If $p(e_1)$ is A_{11} then ***Trypsin***.

Finally, using the fourth rule evaluation criterion, the we obtain the following rules (error rate = 24.42%):

If $p(e_2)$ is A_3 and $p(e_3)$ is A_3 then ***Globin***.

If $p(e_1)$ is A_8 and $p(e_2)$ is A_2 and $p(e_6)$ is A_8 then ***Kinase***.

If $p(e_2)$ is A_8 and $p(e_4)$ is A_1 and $p(e_6)$ is A_7 then ***Ras***.

If $p(e_1)$ is A_1 and $p(e_2)$ is A_1 and $p(e_3)$ is A_2 then ***Trypsin***.

Clearly, the generated rules are simple and comprehensible for humans. The rules obtained from the first and third criteria use the measure in (9) and are shorter and so more interpretable. On the other hand, the rules generated from patterns in the decision subspace (i.e. third and fourth criteria) are more accurate, as they rely more on positive patterns.

To compare the fuzzy rule-based classifiers designed on the basis of the four criteria mentioned above, we have performed five runs of ten-fold cross validation (10-CV) on the whole data (1929 sequences). This testing method will clarify the generalization ability of classifiers in classifying test sequences of proteins. Table 2

compares the results of four rule evaluation criteria where all values are averaged. These results illustrate that higher accuracies are obtainable when evaluating candidate rules using patterns in the decision subspace. In this case, combining the measures in (8) and (9) achieves more accurate classifiers whereas this situation is reversed when using the patterns of covering subspaces.

Criterion number	Short description of criterion	Error rate	Number of rules
First	Measure in (9); Patterns in covering subspace	32.92	4.00
		28.11	5.00
		26.14	6.00
		25.15	7.00
		24.76	8.00
Second	Measures in (8) and (9); Patterns in covering subspace	32.85	4.00
		30.38	5.00
		29.43	5.98
		28.96	6.96
		28.53	7.92
Third	Measure in (9); Patterns in decision subspace	29.45	4.00
		27.70	5.00
		25.69	6.00
		23.70	7.00
		22.30	8.00
Fourth	Measures in (8) and (9); Patterns in decision subspace	25.10	4.00
		22.05	5.00
		21.27	6.00
		20.77	7.00
		20.38	8.00

TABLE 2. Comparing the performance of fuzzy classifiers.

For comparing the performance of our fuzzy classifiers with others in the literature, we have implemented a modified version of some approaches. Since the length of rules in our algorithm is restricted to three, the other methods are run under conditions that the comparisons fair. The PCNSA linear classifier in [6], first reduces the dimension of the training data set to r and then finds a null space of size s for each class by extracting the least variance of this class by means of eigenvalue decomposition. In order to provide similar conditions in this experiment, both parameters r and s are set to three.

Similarly, to employ the neural networks for protein classification as in [23], we have used only three attributes of the data set. For this purpose, the Fisher interclass separability criterion [20] is used to rank the features and then three highest ranked

features (i.e. e_2, e_1 and e_3) are selected for experiment. Finally, since the C4.5 algorithm [19] is the well-known non-fuzzy rule-based classifier, its result is also included, butthe length of rules is again limited to three. Table 3 illustrates the error rates obtained from five runs of the 10-CV testing method for these approaches. It is clear that among these classifiers, the fuzzy approach which uses the fourth rule evaluation criterion is the best and the fuzzy approach which uses the second criterion is the worst.

Classifier	Error rate
Fuzzy, 1st	24.76
Fuzzy, 2nd	28.53
Fuzzy, 3rd	22.30
Fuzzy, 4th	20.38
PCNSA	23.70
Neural Network	26.41
C4.5	24.35

TABLE 3. Accuracy comparison of some protein classifiers.

5. **Conclusion**

In this paper, we have modified a common approach for designing fuzzy rule-based classifiers in order to generate a compact set of simple and interpretable fuzzy rules for classifying the protein sequences. To extract relevant features from protein sequences, we counted the occurrences of six exchange groups in each sequence. Since the rule generation approach needed a great amount of CPU time, we generated fuzzy rules with only three or less antecedent variables, but in order to obtain higher classification accuracy, we investigated four distinct criteria for evaluating candidate rules and compared their accuracy with other well-known rules using an experimental dataset.

We note that the accuracy of classification obtained by our fuzzy classifiers was low as we used a very limited number of features. It is possible to increase the performance through weighting fuzzy rules, but this is not favored by biologists. In future work, we hope to find an efficient approach, using a greater number features with less computational effort, for generating more accurate fuzzy rules.

REFERENCES

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A. I. Verkamo, *Fast discovery of association rules,* in U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining,* AAAI Press, 1996.

[2] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman, *Gapped blast and PSI-blast: A new generation of protein database search programs,* Nucleic Acids Research, **25 (17)** (1997), 3389-3402.

[3] S. Bandyopadhyay, *An efficient technique for superfamily classification of amino acid sequences: feature extraction, fuzzy clustering and prototype selection,* Fuzzy Sets and Systems, **152** (2005), 5-16.

[4] A. Baxevanis and F.B.F. Ouellette, *Bioinformatics: A practical guide to the analysis of genes and proteins,* Wiley, New York, 1998.

[5] M. O. Dayhoff, R. M. Schwartz and B. C. Orcutt, *A model of evolutionary change in proteins,* Atlas of Protein Sequence and Structure, **5** (1978), 345-352.

[6] L. French, A. Ngom and L. Rueda, *Fast protein superfamily classification using principal component null space analysis,* Proc. 18th Canadian Conference Artificial Intelligence, Victoria, Canada, (2005), 158-169.

[7] A. Gonzalez and R. Perez, *SLAVE: A genetic learning system based on an iterative approach,* IEEE Trans. Fuzzy Systems, **7 (2)** (1999), 176-191.

[8] H. Ishibuchi, T. Nakashima and T. Morisawa, *Voting in fuzzy rule-based systems for pattern classification problems,* Fuzzy Sets and Systems, **103 (2)** (1999), 223-238.

[9] H. Ishibuchi, K. Nozaki, and H. Tanaka, *Distributed representation of fuzzy rules and its application to pattern classification,* Fuzzy Sets and Systems, **52 (1)** (1992), 21-32.

[10] H. Ishibuchi and T. Yamamoto, *Comparison of heuristic criteria for fuzzy rule selection in classification problems,* Fuzzy Optimization and Decision Making, **3 (2)** (2004), 119-139.

[11] H. Ishibuchi and T. Yamamoto, *Rule weight specification in fuzzy rule-based classification systems,* IEEE Trans. Fuzzy Systems, **13 (4)** (2005), 428-435.

[12] T. Jaakkola, M. Diekhans and D. Haussler, *A discriminative framework for detecting remote protein homologies,* Journal of Computational Biology, 2000.

[13] C. Leslie, E. Eskin and W.S. Noble, *The spectrum kernel: a string kernel for SVM protein classification,* Pac. Symp. Biocomputing, (2002), 564-575.

[14] M. Madera and J. Gough, *A comparison of profile hidden Markov model procedures for remote homology detection,* Nucleic Acids Res., **30** (2002), 4321–4328.

[15] E. G. Mansoori, M. J. Zolghadri and S. D. Katebi, *A weighting function for improving fuzzy classification systems performance,* Fuzzy Sets and Systems, **158 (5)** (2007), 583-591.

[16] E. G. Mansoori, M. J. Zolghadri and S. D. Katebi, *Using distribution of data to enhance performance of fuzzy classification systems,* Iranian Journal of Fuzzy Systems, **4 (1)** (2007), 21-36.

[17] R. Mikut, J. Jäkel and L. Gröll, *Interpretability issues in data-based learning of fuzzy systems,* Fuzzy Sets and Systems, **150** (2005), 179-197.

[18] W. Pedrycz, *Why triangular membership functions?,* Fuzzy Sets and Systems, **64 (1)** (1994), 21-30.

[19] J. R. Quinlan, *Improved use of continuous attributes in C4.5,* Journal of Artificial Intelligence Research, **4** (1996), 77-90.

[20] J. A. Roubos, M. Setnes and J. Abonyi, *Learning fuzzy classification rules from labeled data,* IEEE Trans. Fuzzy Systems, **8 (5)** (2001), 509-522.

[21] The UniProt Consortium, *The Universal Protein Resource (UniProt),* Nucleic Acids Research, **35** (2007), D193-D197.
[22] D. Wang and G. Huang, *Protein sequence classification using extreme learning machine,* Proc. Int. Joint Conf. Neural Networks, Canada, 2005.
[23] D. Wang, N. K. Lee and T. S. Dillon, *Extraction and optimization of fuzzy protein sequences classification rules using GRBF neural networks,* Neural Information Processing - Letters and Reviews, **1 (1)** (2003), 53-59.
[24] J. T. L. Wang, Q. C. Ma, D. Shasha and C. H. Wu, *New techniques for extracting features from protein sequences,* IBM Systems Journal, **40 (2)** (2001), 426-441.
[25] C. H. Wu and J. W. McLarty, *Neural Networks and Genome Informatics,* Elsevier, Amsterdam, (2000).
[26] M. J. Zolghadri and E. G. Mansoori, *Weighting fuzzy classification rules using Receiver Operating Characteristics (ROC) analysis,* Information Sciences, **177 (11)** (2007), 2296-2307.

EGHBAL G. MANSOORI*, COMPUTER SCIENCE AND ENGINEERING DEPARTMENT, COLLEGE OF ENGINEERING, SHIRAZ UNIVERSITY, SHIRAZ, IRAN
E-mail address: **mansoori@shirazu.ac.ir**

MANSOOR J. ZOLGHADRI, COMPUTER SCIENCE AND ENGINEERING DEPARTMENT, COLLEGE OF ENGINEERING, SHIRAZ UNIVERSITY, SHIRAZ, IRAN
E-mail address: **zjahromi@shirazu.ac.ir**

SERAJ D. KATEBI, COMPUTER SCIENCE AND ENGINEERING DEPARTMENT, COLLEGE OF ENGINEERING, SHIRAZ UNIVERSITY, SHIRAZ, IRAN
E-mail address: **katebi@shirazu.ac.ir**

HASSAN MOHABATKAR, BIOLOGY DEPARTMENT, COLLEGE OF SCIENCE, SHIRAZ UNIVERSITY, SHIRAZ, IRAN
E-mail address: **mohabatkar@susc.ac.ir**

REZA BOOSTANI, COMPUTER SCIENCE AND ENGINEERING DEPARTMENT, COLLEGE OF ENGINEERING, SHIRAZ UNIVERSITY, SHIRAZ, IRAN
E-mail address: **boostani@shirazu.ac.ir**

MOHAMMAD H. SADREDDINI, COMPUTER SCIENCE AND ENGINEERING DEPARTMENT, COLLEGE OF ENGINEERING, SHIRAZ UNIVERSITY, SHIRAZ, IRAN.
E-mail address: **sadredin@shirazu.ac.ir**

* CORRESPONDING AUTHOR