Rough Evolutionary Fuzzy System Based on Interactive T-Norms

Graciela L. Meza Lovón and Maria Bernadete Zanusso

Universidade Federal do Mato Grosso do Sul
Departamento de Computação e Estatística
Campo Grande - MS - Brasil
gmezal@ucsp.edu.pe, mzanusso@dct.ufms.br

Abstract. A rough evolutionary neuro-fuzzy system for classification and rule generation is proposed. Interactive and differentiable t-norms and t-conorms involving logical neurons in a three-layer perceptron are used. This paper presents the results of application of the methodology based on rough set theory, which initializes the number of hidden nodes and some of the weight values. In search of the smallest network with a good generalization capacity, the genetic algorithms operate on population of individuals composed by integration of dependency rules that will be mapped on networks. Justification of an inferred decision was produced in rule form expressed as the disjunction of conjunctive clauses. The effectiveness of the algorithm is demonstrated on a speech recognition problem. The results are compared with those of fuzzy-MLP and Rough-Fuzzy-MLP, with no logical neuron; the Logical-P, which uses product and probabilistic sum; and other related models.

Keywords: T-norms, Classification, Rule Generation, Hybrid System.

1 Introduction

One of the objective of this work is to present the t-norms-based Interactive and Differentiable Neuro-Fuzzy System (IDNFS). This system has as nucleus a multilayer perceptron with three layers in which its neurons carry out logical operations *And* and *Or* and are trained with the backpropagation algorithm. Sec.2 presents the IDNFS with t-norms defined on [-1,1] following the L-fuzzy set concept [8].

The differentiability is an important characteristic in neuro-fuzzy systems because it allows the direct application of the training algorithms based on the descendant gradient [17]. In a data domain with different degrees of granularity among the attributes the interactivity has advantages on the min and max operators that are completely noninteractive [1].

When applied on a data set, the efficiency of the learning of a network measured in terms of its generalization capacity depends on the number of layers, and neurons and on the weights of connections among its neurons. On a symbolic knowledge-based neural networks these parameters can be determined by mapping rules. In order to solve this problem, the initialization methodology based

H. Geffner et al. (Eds.): IBERAMIA 2008, LNAI 5290, pp. 123–132, 2008.

on Rough Set (RS) theory and Genetic Algorithm (GA) is proposed in Sec. 3. RS is based on the application of mathematical concepts of equivalency classes and quotient set in an environment of uncertainty to generate dependency rules from a data set that will be mapped on the neuro-system. GA is introduced to determine the dependency rule combination that results in a IDNFS with the best generalization capacity. Many of the ideas and procedures of our methodology were previously proposed in [3][16]. However, there has been some attempts to overcome some of the limitations found in their methodology, mainly with the advantages offered by GA and other aspects of RS not considered previously.

The model proposed in this paper performs two main tasks. First we construct the three-layered fuzzy logical network for classifying multiclass patterns. Next, the trained network is used to generate rules. The connection weight in this stage constitutes the knowledge base (in embedded form) for the classification problem under consideration. The model is now capable of inferring the output decision for complete and/or partial quantitative inputs and provide justification for any conclusion. If asked by the user, the proposed model is capable of justifying its decision in rule form (in terms of the salient features) with the antecedent and consequent parts produced in linguistic and natural terms.

Sec. 4 shows that on a speech recognition problem the IDNFS with 17 nodes in the hidden layer, even without the fuzzification at the output, gave the best performance between other logical classifiers with 20 or 22 nodes. The results were very promising not only because the process of initialization was partially automatized but also because the generalization capacity was improved in the data set used in our experiments. The preliminary results about the rule generation for speech data are shown in Sec. 3.

2 Neuro-fuzzy System

In [15][17] several distinct types of fuzzy neurons were proposed, which could potentially give rise to many fuzzy neural networks. Our goal in this work is to use the interactive and differentiable t-norms defined by Zanusso [4][14] and to show that the classification results and the rules generated by the system are satisfactory and sometimes better than [2][10][3].

In the literature, the more frequent t-norms are those from *min* and *max*. However, the lattice operations are completely non interactive. The lack of interaction is well represented by the *min* and *max* operations: note that $min(x,a)$ returns x once x is less than a and this result does not reflect the value of a. If there is an interaction, one should favor some other t- and s-norms that are interactive [1].

The definition of a learning algorithm for a fuzzy perceptron using *min* and *max* is not as straightforward as for its neural counterpart because of the usually non-differentiable t-norms and t-conorms used as activation functions for the units. So a gradient descent method cannot be used. [17] discussed some neuro-fuzzy approaches that use special soft-min/soft-max functions to solve this problem. Another approach to solve it is GA.

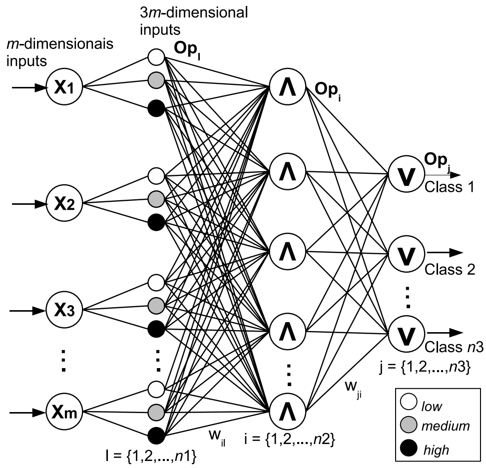

Fig. 1. Architecture of IDNFS

The IDNFS is defined in Fig. 1. This architecture is equal to KDL and Logical-P in [2]. Those models use the conjugate pair *min-max* or the *product-probabilistic sum* operators to represent the *And* and *Or* nodes, respectively. To solve the non interactivity problem, KDL uses various implication operators to introduce different amounts of interaction during back propagation of errors; the results of this are shown in Table 1, Section 4.

2.1 Interactive T-Norm and T-Conorm

[4][14] presents the characterization theorems of t-norm and t-conorm defined on $[-1, 1]$ (observing the extremes) using the axiomatic skeletons defined by Schweizer [7] and the concept of *L-fuzzy set* by Klir [8][9]. The t-norm, which performs the operation *And* among fuzzy sets, is denoted by C and the t- conorm, which carries out the operation *Or*, by D. These are defined via:

$$C(x_1, x_2, \ldots, x_k) = G_1(G_1(x_1) + G_1(x_2) + \ldots + G_1(x_k)) \tag{1}$$

and

$$D(x_1, x_2, \ldots, x_k) = G_3(G_2(x_1) + G_2(x_2) + \ldots + G_2(x_k)), \tag{2}$$

where $G_1(x)$ is the decreasing generator with $G_1^{-1}(x) = G_1(x)$ and $G_2(x)$ is the increasing generator with $G_2^{-1}(x) = G_3(x)$ being defined via:

$$G_1(x) = \frac{1-x}{1+x},$$
$$G_2(x) = \frac{1+x}{1-x},$$
$$G_3(x) = \frac{x-1}{x+1}.$$

In the logical neurons in Fig.1 C and D were used to calculate $\wedge = And$ and $\vee = Or$, respectively. The IDNFS has $n_1 = 3m$ neurons in the input layer, n_2 in the hidden and n_3 in output.

2.2 Input/Output Vector Representation and the LMS Error

The p-esimal m-dimensional pattern (or example)
$X_p = [X_{p1}, X_{p2}, \ldots, X_{pm}]$ is represented as $3m$-dimensional vector:

$$O_p = [\mu_{low}(x_{p1}), \mu_{medium}(x_{p1}), \\ \mu_{high}(x_{p1}), \ldots, \mu_{low}(x_{pm}), \mu_{medium}(x_{pm}), \mu_{high}(x_{pm})], \tag{3}$$

where $\mu_{low}(x_{pi}), \mu_{medium}(x_{pi})$ and $\mu_{high}(x_{pi})$, for $i = 1, \ldots, m$ indicate the membership degrees for the value x_{pi} of the input feature X_{pi} to the low, medium and high fuzzy sets, respectively.

The membership function μ, assuming values in $[-1, 1]$, is defined via:

$$\mu(x) = \begin{cases} 2(2(1 - \frac{|x-c|}{\lambda})^2) - 1 \text{ if } \frac{\lambda}{2} \leq |x - c| \leq \lambda \\ 2(1 - 2(\frac{|x-c|}{\lambda})^2) - 1 \text{ if } 0 \leq |x - c| \leq \frac{\lambda}{2} \\ -1 \qquad\qquad\qquad \text{otherwise} \end{cases}, \tag{4}$$

where $\lambda > 0$, the radius and c, the central point, are calculated using the minimum and maximum values of the X attribute in the training set and depends on the parameter $0 < s \leq 1$ which controls the extension of overlapping.

In the neurons of the And layer the activation functions used are

$$net_{pi} = \bigwedge_{l=1}^{n_1} (O_{pl} \vee w_{il}), \tag{5}$$

whose outputs are $O_{pi} = net_{pi}$, for input-output case p and $i = 1, \ldots, n_2$.

To the neurons from the Or layer, the activation functions used are,

$$net_{pj} = \bigvee_{i=1}^{n_2} (O_{pi} \wedge w_{ji}), \tag{6}$$

whose outputs are $O_{pj} = net_{pj}$, for input-output case p and $j = 1, \ldots, n_3$.

The Least Mean Square (LMS) error was minimized by gradient-descent technique using η (learning-rate) and α (momentum constant) parameters. It is important to mention that after the updating of the weights, a truncation procedure is carried out, aiming to guarantee that their values belong to the interval (-1,1). The derivation of the error is shown in [5][6].

3 The Initialization Methodology

This section presents a summary of the initialization methodology. [11][12] has included more explanatory figures and the pseudocode of the Genetic Algorithm (GA).

The fuzzified training data (Sec. 2.2) is used by the RS theory. It includes tasks such as the binarization which creates granules of information through the application of a cut point (pc) over the data. Values higher than pc become 1, otherwise they become 0. In the theory of RS [13], the information in a data set are considered in a table named Information System (IS). The application of the RS theory allows discovering minimum subsets of attributes named *reducts* whose equivalence classes are the same as those produced by the whole set of attributes. A Decision System (DS) is a IS, which has a decision attribute d, and from which the d-reducts are determined .

For every d-reduct of the DS a dependency rule is generated: the antecedent is determined by conjunction of the attributes of a d-reduct; the consequent is d. These rules indicate the conditional attributes which discriminate the DS classes. They are denoted by RDS_q, $q = 1, 2, \ldots, Q$, where Q is the number of d-reducts. An example for the *Iris* dataset [19] is $high_3 \wedge low_1 \rightarrow specie$, where the subindex refers to the crisp attribute that was fuzzified. This rule discriminates a class representatives from other different class representatives.

Another type of dependency rule allows distinguishing the representatives of a given class. The DS is divided into n subsystems, one for each class, and their reducts are found. From those, the rules are generated: the antecedents like those of RDS, whereas the consequents are determined by the class. These rules are denoted by R_{jq_j}, $j = 1, 2, \ldots, n$, n is the number of classes and $q_j = 1, 2, \ldots, Q_j$, Q_j is the number of reducts of the j-esimal class. An example for the *Iris setosa* is $high_2 \wedge medium_2 \wedge low_1 \rightarrow I.setosa$.

In this work an integration of one RDS, randomly chosen from the set of Q rules, with one R rule, is proposed; named *class rule* and denoted by *R-RDS*. The antecedent is the conjunction of the antecedents of both rules and the consequent is the same as that of the rule R. [3] did not take into consideration the integration of rules which has the advantage of distinguishing the representatives of a given class and, at the same time, distinguishing those representatives from other of different classes.

The Genetic Algorithm (GA) is applied over a IDNFS's population. The aim is to find the IDNFS with the smallest number of neurons in the hidden layer and the highest hit rate on the test set.

The individual representation for GA considers that the antecedents of several class rule with the same consequent can be connected by disjunctions forming a rule $\Re$ whose antecedent is in the Disjunctive Normal Form. An individual is a combination of $\Re_j$, $j = 1, 2, \ldots n$, which defines n regions in the chromosome. Each region is formed by positions (genes) of type $R-RDS$. The individual has $Q_1 + Q_2 + \ldots + Q_n$ positions of the type $R - RDS$. One IDNFS is obtained by mapping an individual.

To define the initial population creation and to diversify the population a policy of $R - RDS$ activation is employed. An active gene participates in the mapping but an inactive one does not. The initial population is created in a way that on a quarter of the individuals all the genes are inactive, consequently, the mapping of those will build architectures with $Q_1 + Q_2 + \ldots + Q_n$ neurons in the hidden layer (the highest possible number). In another quarter, just one gene in the region of each class is randomly chosen to be activated.

The genetic operators are defined in a way different from the traditional GA, all the individuals from the population $P(t)$, in the t-th generation will cross with another individual randomly chosen. The crossover operator is applied by randomly choosing, for each class region, a crossover point from which there will be a change of subsets of $R - RDS$ rules. The sons and their progenitors compose the intermediary population, $IP(t)$ in the t-th generation. The mutation operator is applied over an individual of $IP(t)$, randomly choosing a region in the individual.

The genetic operators do not change the structure of the antecedents and consequents of the rules $R - RDS$ represented in the individuals. Later, the individuals of $IP(t)$ will be mapped over an IDNFS, as reported in [11][12]. The evaluation consists of training and testing each IDNFS from $IP(t)$. The fitness function is defined in terms of n_2, which is the number of neurons in the hidden layer, and hr, hit rate from the test. hr is the mean from the diagonal of confusion matrix. The output of the GA is the IDNFS with a higher hit rate.

The experiments were carried out making learning rate η and the momentum constant α decreases as the number of epochs increases. The methodology of K-fold Cross Validation (CV) for training and testing neural networks was used to determine the best combination from the parameter values. The mutation rate was fixed on 0.10; $s, pc \in \{0.2, 0.4, 0.6, 0.8\}$; ϵ, $\rho \in \{10, 15, 20\}$, that is, the number of evolutions and the number of the best individuals selected after evaluation of IP, respectively. By crossing these parameters, we get the largest mean hr between all folds. These K systems are called evolved-IDNFS.

From K evolved-IDNFS, those that get the largest hr's were chosen. The basic-IDNFS was created fixing s, n_2. The CV methodology, with the former folds, was applied on each basic-IDNFS (aleatory initial weights) and evolved-IDNFS (we have fixed s, n_2 and the initial weights). It is possible, then, to compare the initialization methodology effect when applied on IDNFS.

4 Implementation and Results

The proposed IDNFS with the initialization and rule generation methodology was used on vowel data. These data consist of a set of 871 Indian Telugu vowel sounds [18]; three features: F_1, F_2 and F_3 corresponding to the first, second and third formant frequencies obtained through spectrum analysis of the speech data. There is a 3-dimensional feature space of the six vowel classes ∂(72 examples), a(89), i(172), u(151), e(207) or o(180). This data set has overlapping and nonlinear class boundary. The goal is to compare all the results with [2][3][10][16].

Classification: At first the results from IDNFS without the application of the initialization methodology will be presented. The network was trained with the following parameters: $\eta = 0.01$ (fixed), $\alpha = 0.01$ (fixed), $maxepoch = 4500$,

Table 1. Output Performance for Vowel Data

			Logical				
Model Used	Fuzzy-MLP		KDL	Logical-P		IDNFS	
Hidden nodes n_2	20	22	20	20	22	15	17
∂	44.6	69.8	9.3	31.2	24.6	4.6	4.6
α	65.4	72.8	97.5	76.5	93.8	95.1	93.8
i	79.2	81.8	96.7	90.2	92.2	90.3	88.4
u	88.3	85.9	85.9	80.0	93.5	91.2	91.2
e	75.0	75.0	15.8	75.1	62.3	73.3	76.5
o	85.7	87.2	3.0	86.5	77.9	91.4	91.4
Overall	76.8	80.1	48.8	77.8	77.1	80.0	80.3

$n_2 = 17$ and $s = 1.0$. The last weights from the train are clamped on the neuron connections. In all cases 10% of the samples are used as training set, and the remaining samples are used as a test set. The test examples are fuzzified with the same membership functions as the train examples to enter in the net.

Table 1 compares the average percent of the correct recognition score (on the test set using the best match criterion, both classwise and overall) of the proposed IDNFS model with that of Fuzzy-MLP, KDL and Logical-P. The best match criterion tests whether the jth neuron output has the maximum O_{pj} when the jth component of the desired output vector also has the highest value.

In Table 1 the non logical model Fuzzy-MLP [10] is defined by using weighted sum and sigmoid functions.The logical model KDL uses implication operator of Kleene-Dienes-Lukasiewicz [2] to introduce interaction during back propagation of errors with the non interactive standard min-max operators. The model Logical-P uses the product and probabilistic sum operators [10]. In all these models the input vector consists of membership values for linguistic properties as in Sec. 2.2 with μ transformed into [0,1] while the output vector is fuzzy class membership values. The logical model IDNFS gave the best performance between them with 17 nodes in the hidden layer while the other models had 20 and 22 nodes without implementing the fuzzification at the output.

Table 2 was gotten from application of experiments in Sec. 3 and shows the mean hr for the basic and evolved-IDNFS together with the standard deviation S. The results were attained with $s = 0.8$, $pc = 0.7$, $\epsilon = 15$, $\rho = 15$ and 550 epochs for training during the evolutive process and 4500 epochs on the last

Table 2. Basic-IDNFS and Evolved-IDNFS for the vowel dataset on the test set

n_2	Basic-IDNFS mean $hr \pm S$	Evolved-IDNFS mean $hr \pm S$
21	0.7797 ± 0.039	0.8162 ± 0.020
22	0.7934 ± 0.036	0.8139 ± 0.025
23	0.8071 ± 0.036	0.8151 ± 0.020

Table 3. Rule Generation for Vowel Data with IDNFS

Serial No.	Input Features F_1	F_2	F_3	Justification/Rule Generation If Clause	Then Conclusion
1	900	1400	*unobt*	F_2 is very medium	Mol likely class a
2	700	2300	3100	F_1 is high *And* F_3 is very high	Very likely class e
4	700	1100	2200	F_1 is very medium *And* F_3 is very medium	Likely class a
5	700	1000	2600	F_1 is very medium *And* F_3 is very medium *Or* F_3 is mol high	Likely class a
6	700	1000	*unobt*	F_1 is very medium	Likely class a
8b	700	*Unobt*	*unobt*	F_1 is very medium	Not unlikely class a
9a	600	1600	*unobt*	F_1 is mol high *And* F_2 is very medium *And* F_2 is mol low	Not unlikely class e *Or* o
10a	600	1200	*unobt*	F_1 is very medium *And* F_2 is very medium	Not unlikely class a
10b	600	1200	*unobt*	F_1 is mol high *And* F_2 is very medium	Not unlikely class o
11	50	2400	*unobt*	F_1 is low *Or* F_1 is very medium	Likely class e, but not unlikely class i
13	400	*unobt*	*unobt*	F_1 is very low	Not unlikely class e
15b	250	1550	*unobt*	F_2 is very medium *And* F_2 is mol low	Mol likely class i

train. Note that the increase on the mean hit rates from evolved-IDNFS are larger than those found from basic-IDNFS in average about 2%.

Rule Generation from the Trained Best IDNFS: The rules on Table 3 were generated from the learned knowledge base embedded among the connection weights using the *path generation by backtracking* and the same *certainty measure* for linguistic output as reported in [2].

The magnitudes of the connection weights of trained network were used in every stage of the inferencing procedure. The input feature values of a test pattern, presented in quantitative form, influence the generation of the rule from the trained set of connection weights. This helps extract rules relevant to that region of the feature space that is local to the area pointed to by the feature values of the pattern. On the table, *mol* stands for more or less and *unobt* means that the membership value of that feature value to *low*, *medium* and *high* will be equal to 0, the most ambiguous value. The consequent part clauses were generated for all output neurons that had certainty measure larger than 0.1.

5 Conclusion and Discussion

A three-layered rough evolutionary neuro fuzzy system has been presented. Interactive and differentiable conjugate pair of t-norms was used to perform the logical operations in the neurons. It could be seen in Table 1, the performance of IDNFS classification showed to be comparable to those obtained by models Fuzzy-MLP, KDL and Logical-P. It is important to point out that this performance was obtained by using a network with less neurons in the hidden layer and not yet introducing the fuzzyfication in the net output. [2] introduced different amounts of fuzziness at the output on model Logical-P; the performance increased from 77.8% to 86.1%, that is, 8.3% above. With further development of this work, a comparison is expected between the results of the applications of differentiable and interactive t-norms, proposed in Sec. 2, and the t-norms of max and min and other t-norms proposed in the literature in a data set with different levels of granularity.

The use of rough set theory, which also deals with "information granules", has been used to optimize the number of neurons in the hidden layer and to initialize the weights for some connections between the layers characterizes a knowledge-based network. Note that we have attained satisfactory results when comparing the performance on the test set from speech data between the evolved-IDNFS (Table 2) and various neural net models. [3] has introduced the RS-initialization in Fuzzy-MLP (3 hidden layer, 22 nodes) and renamed it Rough-Fuzzy-MLP. The performance increased from 83.6% to 85.1%, that is, 1.5% better; while the evolved-IDNFS is better than basic-IDNFS in average 2.0% and with one hidden layer. Perhaps if we implement the fuzzification at the output in the IDNFS model we can increase its performance.

After the design, training and test of the network is complete, it is expected the learned hypothesis be able to infer the correct classification for future examples and, if asked by the user, the proposed model is capable of justifying its decision in rule form with the antecedent (in terms of the salient features) and the consequent parts produced in linguistic terms. The IDNFS model generates a number of such rules in If-Then form corresponding to each presentation of examples extracted from the same population of the train and test sets. The rules generated by IDNFS (Table. 3) showed to be satisfactory in relation to the classification given to test input feature and consistent with the rules generated by Logical-P and FUZZY-MLP models [2] (Tables 9,10), when the same set of test input feature was presented. It is still necessary to apply some measures in order to evaluate the performance of the rules.

Acknowledgments. The authors acknowledge CAPES-Brazil for the scholarship.

References

1. Klösgen, W., Zytkow, J.M.: Handbook of Data Mining and Knowledge Discovery. Cap. 10 by Witold Pedrycz, 1st edn. Oxford University Press, New York (2002)
2. Mitra, S., Pal, S.K.: Logical Operation Based Fuzzy MLP for Classification and Rule Generation. Neural Networks 7(2), 683–697 (1994)
3. Banerjee, M., Mitra, S., Pal, S.K.: Rough Fuzzy MLP: Knowledge Encoding and Classification. IEEE Transactions On Neural Networks 9(6), 1203–1216 (1998)
4. Zanusso, M.B.: Familias de T-Normas Diferenciáveis, Funções de Pertinência Relacionadas e Aplicações. Universidade Federal de Santa Catarina. Tese de Doutorado, Brasil (November 1997)
5. Oliveira, F.R., Zanusso, M.B.: A Fuzzy Neural Network with Differentiable T-Norms: Classification and Rule Generation. In: International Conference on Artificial Intelligence(ICAI), Las Vegas, Nevada, June 27-30, vol. I(1), pp. 195–201 (2005)
6. Oliveira, F.R.: Rede Neural Difusa com T-normas Diferenciáveis e Interativas. Universidade Federal de Mato Grosso do Sul, Dissertação de Mestrado, Brasil, Novembro (2006)
7. Schweizer, B., Sklar, M.: Associative Functions and Statistical Inequalities. Publ. Math. Debrecen 8(1), 169–186 (1961)
8. Klir, G.J., Folger, T.A.: Fuzzy Sets, Uncertainty and Information. Prentice Hall, New Jersey (1988)
9. Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall, New Jersey (1995)
10. Pal, S.K., Mitra, S.: Multilayer Perceptron, Fuzzy Sets, and Classification. IEEE Transactions on Neural Networks 3(5), 683–697 (1992)
11. Lovón, G.L.M.: Rough Sets e Algoritmo Genético para Inicializar um Sistema Neuro-Fuzzy. Universidade Federal de Mato Grosso do Sul. Dissertação de Mestrado, Brasil (April 2007)
12. Lovón, G.L.M., Zanusso, M.B.: Rough Sets e Algoritmo Genético para Inicializar um Sistema Neuro-Fuzzy. In: XXXIII Conferencia Latinoamericana en Informática, San José, Costa Rica (2007)
13. Pawlak, Z.: Rough Sets - Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
14. Zanusso, M.B., Araújo, A.: Differentiable T-Norms and Related Membership Functions Families and their Applications. In: IBERAMIA-SBIA, November 19-22, vol. 2, pp. 294–303 (2000)
15. Pedrycz, W., Gomide, F.: An Introduction to Fuzzy Sets - Analysis and Designs, 1st edn. MIT Press, Cambridge (1994)
16. Pal, S.K., Mitra, P.: Pattern Recognition Algorithms for Data Mining. Chapman/Hall-CRC, New York (2004)
17. Nauck, D., Klawonn, F., Kruse, R.: Neuro-Fuzzy Systems. John Wiley and Sons, New York (1997)
18. Indian Statistical Institute, Calcutta, `http://www.isical.ac.in/~miu`
19. Fisher, R.A. (1936), `http://archive.ics.uci.edu/ml/datasets/Iris`