FUZZY LEARNING CLASSIFIER SYSTEMS FOR CLASSIFICATION TASK

Helen Afanasyeva

Transport and Telecommunication Institute
Lomonosova 1, Riga, LV 1019, Latvia
Ph: (+371)-7615025. Mob: (+371)-9186405. E-mail: jelena_a@dtc.lv

Abstract

The Fuzzy Learning Classifier System (FLCS) is a crossover between Learning Classifier System (LCS) and Fuzzy Logic controllers. The LFCS allows for variables to take continuous values basing on data interpretation implemented by fuzzy sets. We investigate some modifications of FLCS architecture and algorithms applied to classification problem.

Key words: Fuzzy Logic, Production system, Genetic Algorithms, Reinforcement Learning, Classification task

Introduction

The growing interest of part of Artificial Intelligence (AI) community in Fuzzy Logic (FL) and Learning Classifier Systems (LCS) has generated some hybrid systems to join the advantages of both in new extension of reinforcement learning (RL) algorithm, based on evolutionary methods. So the Fuzzy Learning Classifier System (FLCS) is a crossover between LCS and FL controllers. Fuzzy rule-based systems have been successfully applied to various control problems [2,3,6,7,8]. In these systems fuzzy rules are usually derived from human experts as linguistic if-then rules. Recently several approaches have been proposed to automatically generate fuzzy if-then rules from numerical data without domain experts. So, genetic algorithms [4,5], have been widely used for generating fuzzy if-then rules and tuning membership functions. Genetics-based machine learning methods for rule generation fall into two categories: the Michigan approach and Pittsburgh approach. In the Michigan approach, each rule is handled as an individual, called a classifier. Thus, this approach is referred to as a classifier system. On the other hand, the Pittsburgh approach handles an entire rule set as an individual. In this paper we describe a complete LCS architecture following the Michigan type of classifier system.

While various methods have been proposed for generating fuzzy if-then rules and turning membership functions, only a few methods are applicable to pattern classification problem. This is because the above-mentioned methods lie mainly in the domain of control problems and function approximation problem.

LCS allowed working only with symbolic variables, so made many limits of their implementation especially in classification problems. But recently it became possible to use here FL Controllers. The LFCS allows for variables to take continuous values based on data interpretation implemented by fuzzy sets. Of course it require a lot of changes in classical Holland's LCS architecture in a wide number of systems implementation stages.

The main purpose of this paper is to develop this architecture based on classical one, which is suitable especially for classification task. It is possible, performing a deep analysis of literature, which describes different approaches for such fuzzifications of other related problems, such as control and function approximation problem, making different algorithms changes, turning system's parameters. All our conclusions are made on the basis of Fisher's Iris data classification example.

In continuation of this section we give the basic principles of the mentioned above AI systems Fuzzy Logic and LCS, which perform the hybrid, which is our research purpose. For take more explicit and full knowledge about this problem, it is possible to use referenced literature.

1.1. Fuzzy Logic and operations

Since their introduction in the late sixties, fuzzy sets have been adopted to map real numbers to symbolic labels. Elements of a universe of discourse belong to a fuzzy set to a certain extent, according to the so-called membership function that defines it. Let us consider an universe of discourse X (say an interval of real numbers), and a fuzzy set, associated to the label close, defined by a membership function $\mu_{close}(x)$, that ranges on elements of the universe of discourse and maps them to the real interval [0,1] (Figure 1.).

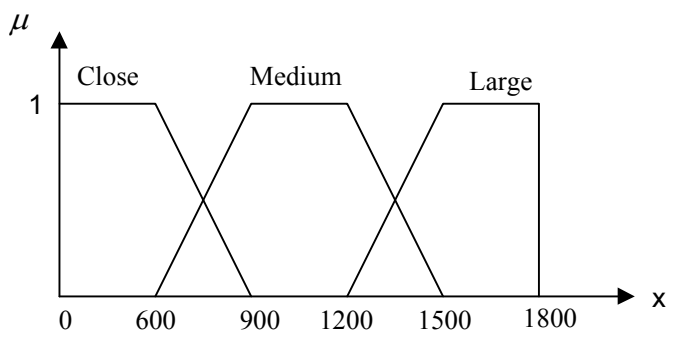

Figure 1. *Fuzzy sets*

Fuzzy sets are often used to classify real values in categories, thus making it possible symbolic reasoning about the phenomena under the incoming numbers. Membership of a number (e.g. x) to a fuzzy set (e.g. *close*) includes two kinds of information: the name of a fuzzy set, that brings information about the category to which the number is classified, and the degree of membership(e.g. $\mu_{close}(x)$), that comes from the definition of the fuzzy set. The information content of a fuzzy overlapping representation is very close to that real-valued representation, and is considerably higher than not overlapping interval based one. Moreover, the selection of certain well-known configuration of fuzzy intervals guarantees high robustness to noise and certain design errors in control software implementation by FLC [7,8].

Let A and B be fuzzy sets over the variable $x \in [x_{min}, x_{max}]$, where $x_{min}, x_{max} \in \Re$. Let A and B be defined by the membership functions $\mu_A(x) \in [0,1]$ and $\mu_B(x) \in [0,1]$ respectively.

The union of A and B is defined by as follows:

$$\mu_{A \cup B}(x) = max(\mu_A(x), \mu_B(x)).$$

The intersection of A and B is defined as follows:

$$\mu_{A \cap B}(x) = min(\mu_A(x), \mu_B(x)).$$

The complement of A is defined as follows:

$$\mu_{A'}(x)=1-\mu_A(x).$$

1.2. Classical Architecture of Learning Classifier Systems

To describe some principles of genetics-based learning systems, the concept of a system of classifiers based on rules and messages is introduced. A classifier system is a machine learning system that learns syntactically simple string rules (called classifiers) able to solve tasks of control, classification and forecasting. From Figure 2 we can see, that LCS consists of three main components [1]:

- Rule and message system
- Apportionment of credit system assignment
- Genetic algorithms.

Sets of rules and messages in the genetic system are a kind of production system. Messages come from the environment or system classifiers. They are compared to classifier rules with regard to matching. A message in the classifier system is a string of fixed length consisting of symbols of a certain alphabet. The more frequently a message matches the condition part of the rule, the more valuable the rule with which it has coincided is. The rule value has to be ascertained in the course of work by means of competition and using credit assignment methods.

The exchange and accumulation of an internal currency provides a natural figure of merit for the application of genetic algorithms. Using a classifier's bank balance as a fitness function, classifiers may be reproduced, crossed and mutated. Thus, not only can the system learn by ranking extant rules, it can also discover new, possibly better rules as innovative combinations of its old rules. Together, apportionment of credit via competition and rule discovery using genetic algorithms form a reasonable basis for constructing a machine learning system [4].

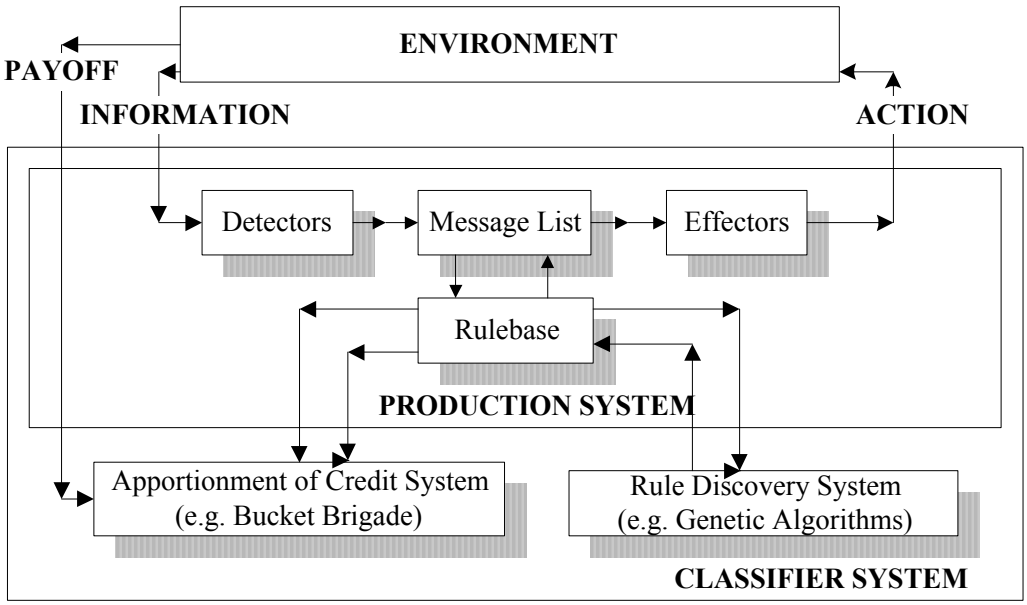

Figure 2. *Classifier system interpretation*

The classifier system under consideration can be simplified; it especially concerns the credit assignment method [3,4]. The distinguishing feature of such systems is that they do not have a message list. In those systems classifiers do not make any chains, but the classifiers, that won in the bids competition receives an environment reward for the correct action.

2. Fuzzy Classifier System

The first, who suggested FLCS was Valenzuela-Rendon in 1991. This system uses fixed-length binary coding for the representation of the fuzzy production rules. Valenzuela-Rendon's system more or less is a Holland LCS that was described above, but with some differences in production system and apportionment of credit system. We will describe them in this section.

2.1. Pattern Classification problem

Let us assume that our pattern classification problem is a c-class problem in the n-dimensional pattern space $[1,0]^n$ with continuous attributes. We also assume that m real vectors $x_p=(x_{p1},x_{p2},...,x_{pn})$, p=1,2,…m, are given as training patterns from the c classes (c<<m). Because the pattern space is $[0,1]^n$, attribute values of each pattern are $x_{pi}\in[0,1]$ for p=1,2,…,m and i=1,2,…,n.

2.2. Fuzzification of input values into fuzzy messages

All input variables in fuzzy CS are numbered, starting from zero. For each variable a group of overlapping fuzzy sets are defined to span its range. Input unit stores the definitions of these membership functions for the input variables. In our system we use triangular and trapezium membership functions. Fuzzy messages in the FCS are implemented as binary strings. A fuzzy message is composed of a tag, that identifies the variable j to which the message refers, and a binary string of length n_j, the number of fuzzy sets defined over the j-th variable. Each of the n_j bits in the binary string indicates if the corresponding fuzzy set is part of message. The tag has as many bits as required to represent the largest variable index. For example, the message 101:0100 says "variable 5 has taken a value that is element of the set 2". The input unit receives real value of input variables and transforms each of these variables into a group of minimal messages (A minimal message is one in which the binary string contains a single bit that is 1); these messages are deposited in the message list. Messages created by the input unit are called input messages. Each message has an activity level which is a measure of belonging of the value of the variable indicated by the tag to the fuzzy set indicated: if an activity level falls to zero, the message is not posted. Each minimal message has the value returned by the corresponding fuzzy set as its activity level.

2.3. Coding of Fuzzy if-then Rule and Fuzzy matching

We will use the coding scheme first suggested by Valenzuela-Rendon in his papers [8]. Fuzzy classifiers in the FCS are formed by a list of conditions, separated by commas, and an action; the action and the condition parts are separated by a slash (/). An implicit "and" is assumed among the conditions. In common way action part may contain also fuzzy overlapping sets and be represented as not minimal message, but in our simple CS for classification task we assume, that each rule may have reference only to one of classes. As an example, the classifier 00:1111,01:101/001 can be expressed in words as: if variable 0 belongs to sets 1,2,3 or 4, and variable 1 belongs to sets 1 or 3, then this rule refers to the class 1.

2.4. Rule Generation

In our fuzzy classifier system, we use fuzzy if-then rules of the following type for the c-class pattern classification problem with the n-dimensional pattern space $[0,1]^n$:
Rule *Rj:* If x_1 is A_{j1} or A_{j1} or … or A_{jn} and … and x_n is A_{j1} or A_{j1} or … or A_{jn} then Class C_j with $CF=CF_j$.

In our fuzzy classifier system, the consequent class C_j and grade of certainty CF_j of each fuzzy rule are determined by the following simple heuristic procedure when its antecedent fuzzy sets $A_{j1},\ldots,A_{jn}$, are specified by genetic operations. This procedure suggested by H. Ishibuchi in [6] helps to find the most suitable action part and the degree of certainty to each rule based on learning set information before learning procedure.

2.5. Fuzzy matching and Fuzzy reasoning for rule evaluation

The FCS performs a fuzzy matching between classifiers and messages. A classifier has an associated activity level, which is a measure of how well its conditions are matched by the messages in the message list. The level of satisfaction of each condition is the maximum of activities levels of the messages that match this condition (the union of fuzzy sets). The activity level of a classifier is the minimal of the levels of satisfaction of each condition (the intersection of fuzzy sets). When a classifier fires (all condition's activity levels are more then 0), this rule is sent to matching list. The rule evaluation module selects the action to be sent into the environment among the classifiers actually presented in matching list. When the antecedent fuzzy sets of each fuzzy if-then rule are given, we can determine the consequent class and the grade of certainty by the heuristic rule generation procedure in the previous section. Here we assume, that we have already generated a set of fuzzy if-then rules for a pattern classification problem. We will describe the voting method that is fuzzy reasoning based on the voting by multiple fuzzy if-then rules. Let us denote the set of fuzzy if-then rules we generated by S. When an input pattern is presented to the fuzzy rule-based classification system, x_p is classified by one of voting methods. Here $\mu_j(x_p)$ is the activity level of the rule $R_j \in S$ and CF_j the degree of certainty of this rule. Now we will describe that algorithm:
Step 1: Calculate $\alpha_{Class\ h}$ *(h=1,2,…,c)* as follows:

$$\alpha_{Class\ h} = \sum_{\substack{R_j \in S \\ C_j = Classh}} \mu_j(x_p) CF_j .$$

Step 2: Classify $\boldsymbol{x}_p$ as the class with the maximum value of $\alpha_{Class\ h}$.

2.6. Credit Distribution Algorithms

We will use in our system well-known from classical LCS Bucket Brigade algorithm, but of course with its corresponding fuzzyfication, suggested in by A. Bonarini and Valenzuela-Rendon [3,8]. First, we assign strength values to the classifiers again. So far as we are dealing with a one-stage system, in which payoff for a certain action is received immediately, where no long-term strategies must be evolved, we can suffice with allowing all matched rules to post their outputs and sharing the payoff among the rules, which were active in the last step, according to their activity levels in this step. For example, if S_t is the set of classifiers, which have been active at time t, and Pt is the payoff received after the t-th step, the modification of the strengths of firing rules can be defined as:

$$u_{i,t-1} = u_{i,t} + P_t \cdot \frac{a_{i,t}}{\sum_{R_j \in S_t} a_{j,t}} \quad \forall R_i \in S_t,$$

where $a_{i,t}$ denotes the activity level of the classifier R_i at time t. The strength of all classifiers R_n is reduced by a certain factor (they pay a tax):

$$u_{n,t-1} = u_{n,t} \cdot (1-T),$$

where T is a small value from [0,1]. The intention of taxation is to punish classifiers, which never contribute anything to the output of the system. In the case, that the problem is so complex that long-term strategies are indispensable, a fuzzification of the bucket brigade mechanism must be found. While Valenzuela-Rendon only provides a few vague ideas, we state one possible variant, where the firing rules pay a certain value to their suppliers, which depend on the activity level. A portion of the payoff then increases the strength of a classifier, which has recently been active in time step t, but it is additionally decreased by a value

$$B_{i,t} = c_L \cdot u_{i,t} \cdot a_{i,t},$$

where $c_L \in [0, 1]$ is the learning parameter. Of course, it is again possible to incorporate terms, which depend on the specifity of the classifier.

2.7. Genetic algorithms

A pre-specified number of fuzzy rules (say N_{rep}) in the current population are replaced with newly generated rules by the genetic operations. In our FCS, the worst N_{rep} rules with the smallest fitness values are removed from the current population and the newly generated fuzzy rules are added. In order to generate N_{rep} fuzzy rules, the genetic operations selection. Crossover and mutation are iterated $N_{rep}/2$ times. That is, $N_{rep}/2$ pairs of fuzzy rules are selected from the current population in the selection operation, and two fuzzy if-then rules are generated from each pair by the crossover and mutation operations. To find classifiers to be replaced we can use different techniques: crowding or, simply, selection classifiers with smallest strength values. Crowding is a procedure that ensures the replacement of the weakest rules, which is closest by structure to the offspring that is to replace it. This enables one to preserve the experience previously accumulated by the system.

2.8. Termination test

We can use any stopping conditions for terminating the execution of our fuzzy classifier system. In computer simulations of this paper, we used the total number of generations as a stopping condition. The final solution obtained by our fuzzy classifier system is the rule set with the maximum classification rate for training patterns over all generations. That is, the final solution is not the final population, but the best population

3. Performance Evaluation

In order to examine the performance of our fuzzy classifier system for multidimensional pattern classification problems with many continuous attributes, we used iris classification data that consists of 150 samples with 4 continuous attributes from three classes. We used the iris data because: this data set is available in the Internet, this data has continuous attributes and simulation

results of our fuzzy classifier systems can be compared with reported results by another classification methods. We pre-processed the data set for our classifier system in following way; the domain of each attribute was linearly normalized to the unit interval [0,1]. By this pre-processing, the maximum and minimum values of each attribute became 1 and 0 respectively. We used the five linguistic values as antecedent fuzzy sets of each fuzzy if-then rules. The task of our fuzzy classifier system is to find a compact rule set with high classification performance. We specified the number of fuzzy if-then rules in each population as 50. The other parameters are specified in our computer simulations and represented in Table1.

Table 1. Computer simulations parameters values

Parameter Name	Parameter value	Parameter Name	Parameter value
GA period	5	Learn Param.	0.2
Life Tax	0.0015	Reward Value	1
GA times	(5/3)	Crossover Prob	0.6
Mutat. Prob.	0.05	Cl. Rep.Type	(The smallest strength/Crowding)

We used 3-fold cross-validation technique to get the average results of system performance and received the following results (Table 2). We can also see from Figures 3, 4 how our classification rate improved during the system's working process. If the parameters are given with sign "/", then they differ from one experiment to another (parameter for experiment #1 / parameter for experiment #2).

Table 2. Average classification rate

Parameter name	Experiment #1	Experiment #2
Iteration number	250	80
Learning data set	98.33%	98.33%
Test data set	98.7%	98%

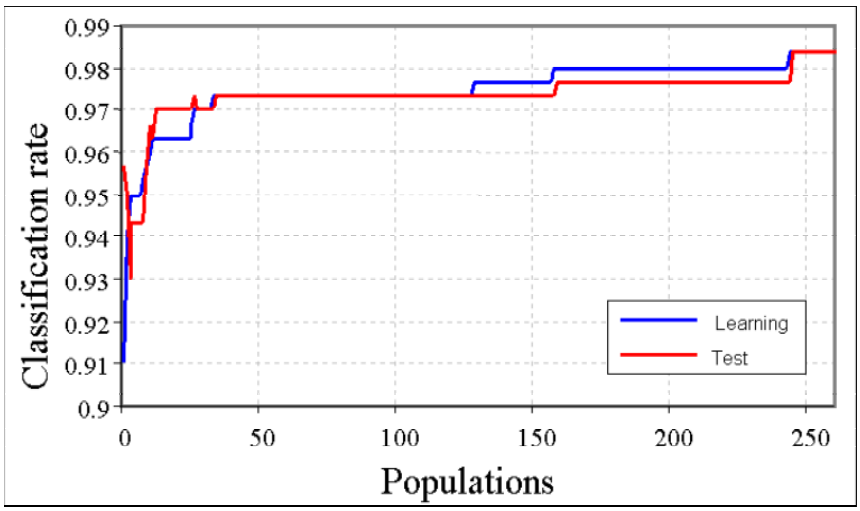

Figure 3. *Classification rates for experiment #1*

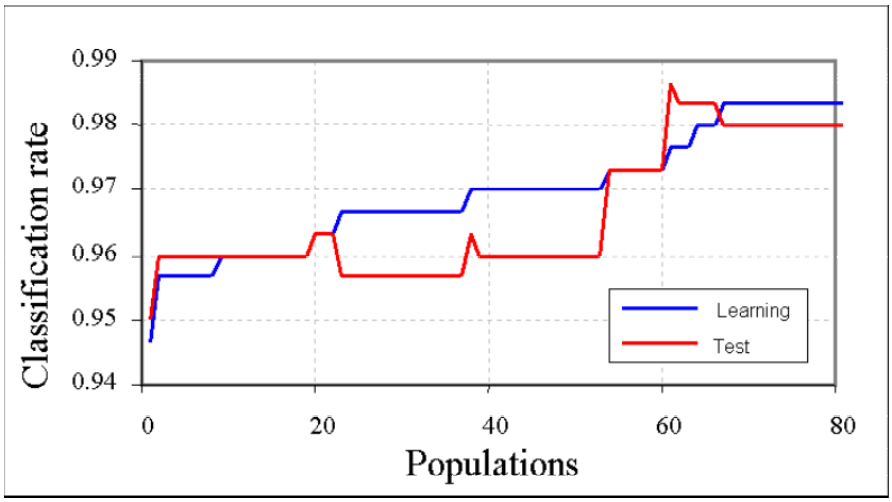

Figure 4. *Classification rates for experiment #2*

The results described in this paper were obtained in the course of hard work in parameters turning and algorithms choosing. Examining those performed experiments, we can assume, that crowding procedure is the most suitable in genetics algorithms, than other techniques, such as simple excluding classifier with the smaller strength value. As we can see such parameter as how much classifiers had to be exchanged also made big influence on classification rate and performance. So the quickest classification rate was obtained with crowding procedure and exchanging only 6 classifiers every genetics algorithms implementing time. Crossover and mutation probabilities values are also very important, here are given their optimal values.

4. Conclusion

In this paper we showed how genetics-based machine learning technique might be used for solving such well-known statistics problem as classification. This is computational intelligence approach, which is developed very quickly, and has a good potentiality in the future. For example, its applications to classification task started only 3 years ago. We build our FLCS based on classical one, suggested by Holland with some differences, which were important due to its fuzzy nature and application field. Since this approach is very new, there is no software available for solving this problem. We developed it using C++ language. This proposed system was tested on well-known Fishier's Iris classification problem. It gave very good results, for this quit difficult non-linear separate task. Other methods such as statistics discriminant analysis or neural networks could not provide better results [6]. So our approach is very perspective and can be a very good alternative to existing classification methods.

Acknowledgement

I appreciate useful comments on the manuscript given by Professor Arkady Borisov from Riga Technical University. I thank also Latvian Education Fond and Aero Navigation Service for financial support.

References

1. Afanasyeva H. Genetics-based Machine Learning Systems for Classification Task // Scientific Proceeding of Riga Technical University. – Riga: RTU, 2001. pp. 8-16.
2. Bodenhofer, U. Genetic Algorithms: Theory and Applications Lecture Notes, Second Edition Institut für Algebra, Stochastik und wissenbasiere mathematische Systeme, Johannes Kepler Universität, Linz, Austria, 2002.
   Internet: http://www.flll.uni-linz.ac.at/lectures/GA/notes.pdf
3. Bonarini, A. (2000). An Introduction to Learning Fuzzy Classifier Systems. In: P. L. Lanzi, W. Stolzmann, and S. W. Wilson (Eds.), Learning Classifier Systems. From Foundations to Applications, Volume 1813 of LNAI, Berlin, pp. 83- 104. Springer-Verlag.
4. Booker, L.B, Goldberg D.E. and Holland J.H., Classifier Systems and Genetic Algorithms, Artificial Intelligence, Vol. 40, 1989, pp. 235-282.
5. Goldberg, D.E., Genetic Algorithms in Search, Ooptimisation, and Machine Learning, Addison-Wesley, Redwood City, 1989.
6. Ishibuchi, H., Nakashima, T., and Murata, T.: Performance Evaluation of Fuzzy Classifier Systems for Multi-Dimensional Pattern Classification Problems, IEEE Trans. on Systems, Man, and Cybernetics- Part B: Cybernetics, Vol. 29, No. 5, pp.601-618 (October, 1999).
7. Matteo Matteucci (2000) Learning Fuzzy Classifier Systems: Architecture and Explorations Issues
8. Valenzuela-Rendon, M. Reinforcement Learning in the Fuzzy Classifier Systems 1997. // Internet: http://citeseer.nj.nec.com.