Proceedings of the 8th International Conference on Applied Informatics
Eger, Hungary, January 27–30, 2010. Vol. 1. pp. 107–114.

Cross Entropy Optimization for Induction of Fuzzy Rule System*

László Kovács

University of Miskolc, Department of Information Technology
kovacs@iit.uni-miskolc.hu

1. Introduction

Fuzzy technology became a very important controlling method in complex systems where traditional methods are unsuccessful. It was proved in [19] that fuzzy rule systems can be used as general approximators of any complex continuous systems. The key element of the approximation process is the construction of the corresponding fuzzy rule system that encapsulates the knowledge on the problem domain. A fuzzy rule base is defined as a set of deduction rules of the form

$$\frac{W_{i1}W_{i2}W_{i3}\cdots W_{iM}}{W_i} \quad (i = 1, \dots, N)$$

where W_{ik}, $W_i$are literals of the form $a_k = \mathrm{A}_{ik}$, $b = B_i$ and i denotes the index of the rule. In the rules

a : linguistic variable of the antecedent space,
b : linguistic variable of the consequence space,
A: a linguistic value in the antecedent space,
B: a linguistic value in the consequence space,
N: number of rules,
M: number of linguistic variables in the antecedent space.

The rule base can be considered as mapping from the Cartesian product of antecedent space into the consequence space

$$f : D_1 \times D_2 \times ... \times D_M \to D$$

where

D_i: the domain of the antecedent space of a_k
D: the domain of the consequence space of b.

*The research was supported by the Hungarian National Scientific Research Fund Grant OTKA K77809.

In the classical approach, the $D_i$domains are defined in the Euclidean space. Usually, the rules are generated by human experts based on the gained experiences in the problem domain. The definition of the rule system is a very critical phase as the quality of the control system depends on the correctness of the included rule system. In the case of complex systems the discovering of the hidden rules is a time consuming process for the experts too. In order to improve the efficiency of rule generation some kind of automatic rule generation method can be applied in the control system. These systems usually generate the rule base with some kind of analysis of the training data. The main benefits of automatic rule generation engines using numerical approximation are among others the followings: it can manage model-free system; it can process large training pools and it can provide a high time- and cost-efficiency. In the last decades, there were several methods developed to solve the rule generation problem.

One of the first proposals for automatic fuzzy rule generation was given in the work of Wang and Mendel [7]. The proposed model uses one-dimensional D domains and it divides the input and output domains into disjoint fuzzy regions. A fuzzy membership function is assigned to each region as shown in Fig 1. Here, five regions are defined where the regions are denoted by S2 (small 2), S1 (small 1), CE (central region), B1 (big 1), B2 (big 2).

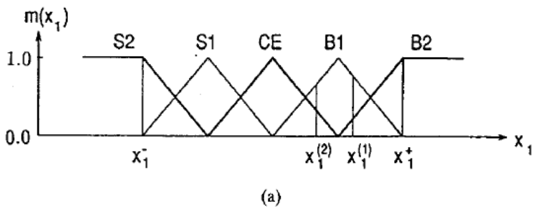

Figure 1: Fuzzy regions and membership functions [7]

As it can be seen in Fig 1, each region has its own membership function (linguistic value) where the shape of each membership function is triangular. The support of the triangular membership function covers not only its own region but the neighboring regions too. In the next step of the processing, a set of base rules is generated from the training pool. In the training pool, the training elements are given in the form

$$(x \in D_I, y \in D_0)$$

where D_I denotes the input domain and D_0 is the symbol of the output domain. For every training elements a fuzzy rule is created in the form

IF x is X THEN y is Y

or in our notation

$$\frac{x = X}{y = Y}$$

where X and Y denotes linguistic values from the corresponding domain. In order to reduce the number of rules, an importance filtering is performed. The weight of a rule is given as

$$d\left(\frac{x = X}{y = Y}\right) = m_X(x) \cdot m_Y(y).$$

where $m_X(x)$denotes the value of the membership function Xat position x. Based on this importance weight, the rules below a threshold value are eliminated from the rule system. In the literature, some improvement of the base algorithm can be found, that perform additional post processing of the rule base to provide a better filtering result.

In [2], the rules are generated using a clustering algorithm. In the antecedent-consequence domain space, the training examples lying near to each other will belong to the same rule. The method partitions the input space into hyper rectangles and defines a fuzzy rule for each cluster. The objective function measures the density and typicality of the distribution. A specialty of this approach is that is uses a Gaussian membership function instead of the usual triangle format.

In the approach of [5], a genetic algorithm was developed to determine the rule systems. The goal of the stochastic optimization is to determine the rule base containing those rules that

- have a great frequency value
- have a big set of positive examples
- have no negative examples

Every chromosome of the rule base represents a fuzzy rule. The proposed method uses the traditional selection, crossover and mutation operators to determine the winner chromosomes. The fitness function calculates the covering values for every training example. The covering value for $e = (x \in D_I, y \in D_0)$ is given as

$$C(e) = \sum d_i(e)$$

where $d_i()$denotes the weight of the i-th rule.

In [4], a hybrid method is applied to generate the rule system. A human expert defines the set of linguistic values in the preparation phase. After determination of the values, a genetic algorithm is applied to define the best matching rule base. In the second phase, after having a rule base, a second genetic algorithm is executed to refine the shape of the membership function for the linguistic values.

One of the latest improvements is presented in the paper [3] from 2005. The proposed method is a combined genetic algorithm – gradient based optimization method. The genetic algorithm is used to find those features of the input with which the separation of classes is optimal. The second step of the method refines the initial fuzzy membership functions in order to give better accuracy. The model is novel in the sense that logical information is directly available and that the fuzzy membership functions are optimized instead of the network weights, so that there is no need to round the weights to integers and thus lose information because of it. The rules are concise and easily understandable because of their disjunctive normal form which is guaranteed by the special network structure.

Summarizing the literature, it can be seen that several approaches exist for generation of fuzzy rule base. Beside the trivial enumeration method several soft computing algorithms were adapted for this problem, like genetic algorithm, neural network or clustering. Our investigation focuses on analysis of cross entropy method for rule base optimization.

2. Rule generation algorithm with CE method

Due to simplicity, only one dimensional antecedent domain is used and only one output rule exists in the investigation domain. The input of the module is the training set

$$T = \{e_i(x_i, y_i)\}.$$

The output is the generated rule base that can be given as one dimensional fuzzy associative memory (FAM) in the form of table:

A_1	A_2	A_3	A_4	A_5					A_n
α_1	α_2	α_3	α_4	α_5					α_n

where A_i are the generated linguistic values in the antecedent space and α_i are membership values for the single output value. Our method uses a model-based approach: it is assumed that the fuzzy membership functions have a sphere shape in one dimensional space. Three parameters are used to describe the membership function:

- center of the sphere (c_i)
- internal radius (core) (r_i)
- external radius (support) (R_i)

and one parameter is needed for the rule base:

- the weights of the antecedent values (α_i)

In the model, the number of input linguistic values is fixed, it is set to K. The goal is to find the optimal parameter quartet for the K linguistic values. The objective or fitness function is defined as follows:

$$E = \sum_T \left(\bigcup_K (m_i(x_j)) - y_j \right)^2$$

where

- m_i denotes the membership function of the i-th linguistic value, it has the following shape:

$$m_i(x) = \begin{cases} 0, \ if\, d(x, c_i) \geq R_i \\ 1, \ if\, d(x, c_i) \leq r_i \\ 1 - \frac{x - c_i - r_i}{R_i - r_i}, \ if \ c_i + r_i \leq x \leq c_i + R_i \\ \frac{x - c_i + R_i}{r_i - R_i}, \ if \ c_i - R_i \leq x \leq c_i - r_i \end{cases}$$

- $\bigcup_K$ denotes the maximum, disjunction of the membership values for the set of linguistic values.

The proposed approach is based on the concept of cross entropy (CE) optimization. The main concept of the CE optimization method can be summarized in the followings [1]. Let $\overline{X} = (X_1, X_2, ..., X_n)$ be a random vector taking values from some real space. Let $\{f(,\bar{v})\}$be a family of probability density functions on this space, where $\bar{v}$is the parameter vector. For any measurable function H, the mean value can be calculated as

$$E_{\bar{v}}(H) = \int H(x) f(x, \bar{v}) dx$$

The symbol $S(\bar{X})$ denotes a real valued objective function. The goal of the investigation is to find the maximum of $S(\bar{X})$ over $\bar{X}$and to determine the position of the optimum value:

$$S(x^*) = \max\{S(x)\}$$

The probability that $S(\bar{X})$ is greater than some γ under given $f(x, \bar{u})$ can be expressed as

$$P_{\bar{u}}(S(X) \geq \gamma) = E_{\bar{u}}(I\{S(X) \geq \gamma\})$$

where $I\{\}$ denotes the indicator function. The goal is to find such $f(x, \bar{u})$density function that is a good approximation of the measured indicator function $I\{S(X) \geq \gamma\}$. A particular convenient measure of distance between two density functions is the Kullback-Leibler distance

$$d(g, h) = E\left(\ln\left(\frac{g(X)}{h(X)}\right)\right)$$

From this formula follows that at the optimum position

$$\int g(x) \ln(h(x)) dx \rightarrow \max$$

and

$$E_{\bar{u}}(I\{S(X) \geq \gamma\} \ln(f(x, \bar{u})) \rightarrow \max$$

hold. These formulas represent the cross entropy value. To determine the optimum value of $\bar{u}$from a smaller set of examples, the importance sampling method is applied to calculate the mean value. During the iteration, the $\bar{u}$at level t is calculated as follows:

$$\bar{u}_t = \arg\max_{\bar{u}}\{E_{\bar{u}}(I\{S(X) \geq \gamma\} \ln(f(x, \bar{u}))\}$$

The main steps of the optimization algorithms are

- 1: determine the initial distribution
- 2: generate a sample with the actual $\bar{u}_t$ parameter, where t denotes the iteration level
- 3: compute the $(1-p)$-quantile γ_t from the sample
- 4: if $\gamma_t$does not differ from the values of the previous iteration levels, it is used as optimum value and the algorithm terminates; otherwise go to point 5

– 5: solve the optimization equation to determine the $\bar{u}_{t+1}$ and go beck to step 2

The main idea of the CE method is to refine the samples for calculating the objective function in an iteration loop. The samples are used to estimate the parameters of the probability density function. The optimum of the objective function depends on these parameters and the goal of this stochastic optimization method is to determine the optimum parameter setting in a small number of iterations.

3. Experiments

In the test system, the fuzzy values are given by trapezoid membership functions with three parameters. The random parameters are generated with a normal distribution having $\sigma = 1$ and

$$f(x, \bar{u}) = \frac{1}{\sqrt{\pi}} e^{-(x-u)^2}$$

Using the normal probability density function, we get that

$$E_{\bar{u}}(I\{S(X) \geq \gamma\} \ln(f(x, \bar{u})) \rightarrow \max$$

is met if

$$E_{\bar{u}}(I\{S(X) \geq \gamma\}(c - (x-u)^2) \rightarrow \max$$

holds. Based on these equations, we get that the parameters of the next level are calculated on the following way:

$$u_t = \sum_{i=1}^{N} \frac{I\{S(X_i) \geq \gamma\} \cdot X_{ij}}{I\{S(X_i) \geq \gamma\}}$$

A minimal training data set is used with 21 sample elements. The shape of the training samples is given if Fig 2.

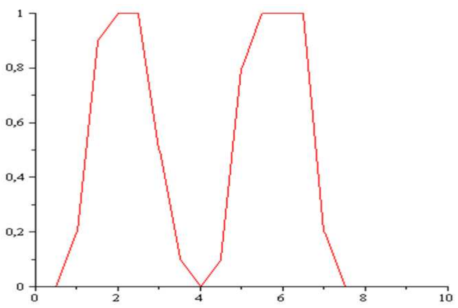

Figure 2: Training sample

In the optimization procedure, the number of approximating values, K is an input parameter. For the case $K = 2$, there are 6 parameters to optimize. To measure the goodness of the approximation, the following distance function was generated:

$$d(T, A) = \sum_{i \in T} \min_{j \in K} \{d_{y_i}(x_i, A_j)\}$$

where

(x_i, y_i): the sample training data pair from T,

A_j: the i-th fuzzy value of the approximation system A,

d_α: the distance defined on the α-cut level.

Using this distance function, the optimization algorithm shows a relative fast convergence to the optimum value as it is shown if Fig 3.

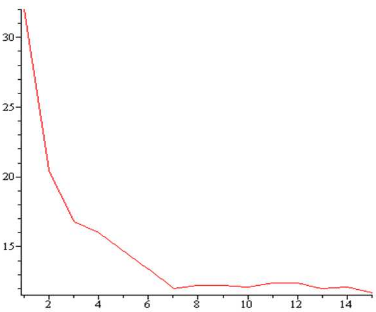

Figure 3: Approximation rate

Reducing the search space only to two parameters (positions of the centers) and setting the radius values to 0.5, the yielded objective function is a function with two parameters. The optimum position is near to the position the general solution but with a higher distance value (as the radiuses are fixed). The surface of this objective function is shown in Fig 4.

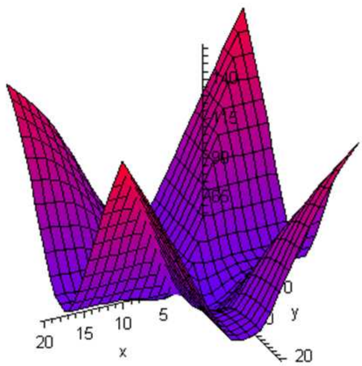

Figure 4: Objective function

4. Summary

The performed analysis shows that the optimization using the cross entropy measure is a good alternative stochastic optimization method. The method provides a very general optimization framework with a relative efficient approximation rate. In the domain of fuzzy control systems, the proposed method can be used to generate the appropriate fuzzy variables matching the experimental training set on control rules. In the next phase of the investigation, the problem domain with several target linguistic variables will be analyzed with the CE optimization method.

References

[1] P. Boer, D. Kroese, S. Mannor, R. Rubinstein: A Tutorial on the Cross-Entropy Method, 2003

[2] G. Castellano, A. Fanelli: A method to construct compact fuzzy systems from input-output data, 1999

[3] Zs. Gera, J. Dombi: Genetic Algorithm with Gradient Based Tuning for Constructing Fuzzy Rules, Publications of ISHRCI, 2005

[4] F.Garrido, R.Ramos: A Methodology for Constructing Fuzzy Rule Based Classification Systems, 1999

[5] F. Herrera, M. Lozano, J. Verdegay: Generating Fuzzy Rules from Examples Using Genetic Algorithms, Proc. of Fifth International Conference of Information Processing and Management of Uncertainty in Knowledge Based Systems, CityplaceParis, 1993

[6] R. Hammell, T. Sudkamp: Learning Fuzzy Rules from Data, RTO Symposium, StateplaceCalifornia, 1998

[7] L. Wang, J. Mendel: Generating Fuzzy Rules by Learning from Examples, IEEE Transactions on Systems, Man and Cybernetics, Vol 22, No 6. 1992, pp 1414-1428.