Fuzzy Rule base Generation from Numerical Data using Biogeography-based Optimization

S Kumar,
P Bhalla,
A Singh,

Fuzzy rule based systems are one of the very important class of knowledge based systems. The knowledge in a fuzzy system is embedded in the form of a rule base. This short article presents a new approach to rule base extraction from numerical data using Biogeography Based Optimization Approach (BBO). The rule base extraction problem is formulated as the minimization problem. BBO was used to enumerate rules corresponding to each data set. The paper discusses rule extraction for type zero TSK fuzzy systems for battery charger. However, the approach is very powerful computation tool to deal with NP hard problems. The results indicate that the BBO is a very promising optimizing algorithm for evolving fuzzy logic based systems.

Keywords: Fuzzy membership function; Rule generation; Soft computing

INTRODUCTION

The most important area of the application of Fuzzy Set Theory is Fuzzy Rule-Based Systems (FRBSs). These kinds of systems constitute an extension of classical Rule-Based Systems, because they deal with fuzzy rules instead of classical logic rules. An important application of FRBSs is Linguistic Modelling, which in this field may be considered as an approach used to model a system making use of a descriptive language based on Fuzzy Logic with fuzzy predicates, where the interpretability of the obtained model is the main requirement. Thus, the linguistic model consists of a set of linguistic descriptions regarding the behavior of the system being modelled.

In this approach, fuzzy linguistic IF-THEN rules are formulated and a process of fuzzification, inference, and defuzzification leads to the final decision of the system. These rules represent domain knowledge acquired from empirical observations and experience. The use of descriptive linguistic variables in fuzzy rules provides the additional advantage of representing this knowledge in a form that is easy for humans to comprehend and validate.

Traditionally, fuzzy rules have been obtained via discussions with domain experts but this approach has many problems and shortcomings the interviews are generally long, inefficient and frustrating for both the domain experts and knowledge engineers, especially so in domains where experts make decisions based on incomplete or imprecise information. This knowledge acquisition phase is often the main bottleneck within the knowledge engineering process and therefore considerable effort has been expended in designing algorithms that automatically induce fuzzy rules from historical data. With this requirement in mind, a large number of methods have been proposed to automatically generate fuzzy rules from numerical data making use of different soft computing techniques, such as, neural networks, genetic algorithms, swarm intelligence, ant colony optimization etc.

In this paper, a novel way of facing the fuzzy rule base problem in proposed making use of Biogeography based optimization algorithms. To do so, the rule base problem will be formulated as an optimization problem. For complex problems the rule base problem turns out to be NP hard problem. In this paper, the rule base generation for Sugeno (type zero TSK) fuzzy models is discussed.

FUZZY RULE-BASED SYSTEMS

Introduction to Fuzzy Rule-based Systems

A fuzzy rule based systems presents four major modules: fuzzification, inference engine, knowledge base and defuzzification module. The fuzzification module transforms the crisp input(s) into fuzzy values. These values are then processed in fuzzy domain by inference engine based on the knowledge base supplied by the domain expert(s). The knowledge base is composed of the Rule Base (RB), characterizes the control goals and control policy of the domain expert by a set of linguistic control rules, and of the Data Base (DB), containing the term sets and the membership functions defining their semantics. Finally the processed output is transformed from fuzzy domain to crisp domain by defuzzification module. The structure of a linguistic Fuzzy rule-based system is shown in Figure 1.

The structure of a rule base can be stated as follows:

\[ Ri : \text{if } X_1 \text{ is } A_{i1} \ldots X_n \text{ is } A_{in} \text{ then } Y \text{ is } B_j \]
In the case of TSK (Takagi, Sugeno and Kang, 1985) models the consequent is the function of inputs. In this case the rule can be written as

\[ R_i : if \ X_1 \ is \ A_{i1} \ and \ \ldots \ and \ X_n \ is \ A_{in} \ then \ Y = f(x, y) \]

Where \( R_i \) is output fuzzy set in the case of Mamdani type systems. In the case of a Sugeno type system, \( B_j \) is a fuzzy singleton as shown in Figure 1. \( X_1, \ldots, X_n \) and \( Y \) are input and output linguistic variables, respectively, and \( A_{i1}, \ldots, A_{in} \) and \( B_j \) linguistic labels, each one of them having associated a fuzzy set defining its meaning.

**Fuzzy Systems Design**

The process of fuzzy systems design involves the following steps:

1. Identify the input and output variables
2. For these variables, generate membership functions and decide their shapes, such as, triangular, Z-type, S-type etc.
3. Generate rule base for the system
4. Select the type of inference are rule composition operator, implication and aggregation operators.
5. Decide on the defuzzification technique and generate a crisp control action (defuzzification).

For the systems of small complexity Step 1 can be performed by the experts by including all the available inputs. For the systems of higher complexity and it is not possible to take into account all the inputs and one may be constrained to select only those inputs which have significant contribution to the overall output of the system. Some of the suggested procedures in literature are forward selection procedure, backward elimination procedure, best subset method and few other statistical selection procedures. Redundant variables may be removed using exhaustive search technique presented by Shakti, et al. Step 2 can be performed with the help of domain expert(s) if they are available, from the common sense, or from the available numerical data. In case numerical data is available for these variables the membership functions generating using techniques like, FCM, neural networks, GA etc can be used. As for as Step 4, is concerned then may be hundreds of combinations of composition, implication and aggregation operators. For Step 5, a large number of defuzzification techniques are available in the literature. Some of the commonly used defuzzification techniques are, centre of gravity (COG), centre of sum methods, first/last of maxima, Mean of maxima (MOM), BADD, SLIDE, Modified-sliden and RAGE methods.

Step 3 involves the development of rule base. In the case of a knowledge based model development, Step 3 is performed by a knowledge engineer with the help of domain experts whereas in the case of data driven modelling, certain computerized techniques are used to develop the rule base. Rule base generation techniques are clarified into four categories. The first category will be called as classical methods. This consists of methods like the one proposed by Wang and Mendel, the second category consists of all the methods that employ genetic algorithm (GA) based methods. The third category contains neural networks based methods for rule base generation. The methods employing the swarm intelligence techniques (PSO and ACO) will be placed into category four. This paper proposes another technique based on the mathematics of biogeography, ie, biogeography based optimization (BBO) algorithm. This technique is placed in category of evolutionary algorithms.

**Rule base Generation Problem Formulation**

Figure 2 represent TSK type zero fuzzy systems. It is clear from figure that such systems consists of four major modules, ie, fuzzifier, rule composition module (fuzzy MIN operators), implication module (multipliers in this case), and defuzzification module.

The overall computed output, in the case of a Sugeno type system, can be written as follows:

\[
\text{Computed output} = \frac{\sum_i W_i C_i}{\sum W_i}
\]

The number of fuzzy rules can be defined as below:

\[
R = \prod_{j=1}^{p} m_j
\]
But these R rules are due to combinations of membership functions of various inputs and these are incomplete as knowledge only about antecedent part are available and consequents are yet unknown. Because for any set of inputs $W_i$ are easily computed by fuzzifier and rule composing modules, the right hand side of output expression (1) can be evaluated if we could choose the proper values for $C_i$. For a given data set of a system, $W_i$ are known. Find the appropriate values of $C_i$ such that, the difference between the computed output and the actual output as given in data is minimum.

$$C_{\text{computed}} = \frac{W_1 C_1 + W_2 C_2 + \ldots + W_R C_R}{W_1 + W_2 + \ldots + W_R}$$ (2)

The computed output are compared with actual output as given in data set and find the error. Let the error be defined as follows:

$$\text{error} E = \text{actual output} - \text{computed output}$$

Now the whole problem of rule base generation boils down to a minimization problem as stated below:

Minimize objective function $E$

$$E = O_{\text{actual}} - O_{\text{computed}}$$

Subject to the constraint that $C_i \in \{\text{specified set of consequents}\}$ (3)

Any minimization technique may not be applicable if the problem is very complex. BBO algorithm are applied to evaluate rule base.

**BIOGEOGRAPHY BASED OPTIMIZATION ALGORITHM FOR RULE BASE GENERATION**

Biogeography-based optimization is a population-based evolutionary algorithm that is based on the mathematics of biogeography. Biogeography is the study of the geographical distribution of biological organisms. In BBO, problem solutions are represented as islands and the sharing of features between solutions is represented as emigration and immigration. An island is any habitat that is geographically isolated from other habitats.

Biogeography-based optimization was first presented in Simon and is an example of how a natural process can be modelled to solve general optimization problems. This is similar to what has occurred in the past few decades with genetic algorithms, neural networks, ant colony optimization, particle swarm optimization, and other areas of computer intelligence. Biogeography is nature’s way of distributing species, and is analogous to general problem solving. Suppose that there are some problem and that a certain number of candidate solutions are there. A good solution is analogous to an island with a high HSI (Habitat suitability index), and a poor solution is like an island with a low HSI. Features that correlate with HSI include factors such as rainfall, diversity of vegetation, diversity of topographic features, land area, and temperature. The variables that characterize habitability are called suitability index variables (SIVs). High HSI solutions are more likely to share their features with other solutions, and HSI solutions are more likely to accept shared features from other solutions. As with every other evolutionary algorithm, each solution might also have some probability of mutation, although mutation is not an essential feature of BBO.

**BBO Algorithm**

1. Initialize the BBO parameters: maximum species count $s_{\text{max}}$, the maximum migration rates $E$ and $I$, the maximum mutation rate $m_{\text{max}}$, an elitism parameter and number of iterations.

2. Initialize a random set of habitats, each habitat corresponding to a potential solution to the given problem.

3. For each habitat, map the Habitat Suitability Index to the number of species $S$, the immigration rate $\lambda_s$ and the emigration rate $\mu_s$.

4. Probabilistically use immigration and emigration to modify each non-elite habitat, then recompute each HSI.

5. For each habitat, update the probability of its species count given by equation (4). Then, mutate each non-elite habitat based on its probability and recompute each HSI.

6. Is acceptable solution found? If yes then go to Step 8.

7. Number of iterations over? If no then go to Step 3 for the next iteration.

8. Stop

**Rule base Generation through BBO**

Here, the algorithm that extracts rules from data set are discussed.

**Step 1: Divide Input and Output Spaces into Fuzzy Regions:**

For each of the input and output variable its range or interval from starting value to end value is identified. Each of the
input and output spaces are partitioned into K regions (membership functions). This algorithm considers only triangular membership functions. However other membership function shapes can be considered. The number K can be different for each of the variables. Any of the methods can be used to cluster the input and output data. Neural networks, GAs and Fuzzy C-means clustering (FCM) etc. can be used to partition input and output variable ranges.

**Step 2 : For Each Training Example Considering the Input Values Generate Fuzzy Rule(s)**

For the steps which follow, let \( x_1 \) and \( x_2 \) be the inputs and \( y \) be the output variable. In order to implement this step, that are followed as:

a) Determine the membership grade \( \mu_i \) of a given input \( x_i \) into the different fuzzy membership functions.

b) Enumerate the composed values or depth of firing \( W_i \).

c) Call BBO algorithm to choose appropriate value of \( C_i \) from the specified set, such that, \( \text{error}_i = (\text{actual output} - \text{computed output}) \) is minimized.

d) Enumerating the consequents completes the rule for this set of input values

e) Create a combined fuzzy rule base.

f) Rule Reduction Technique: Assign a degree to each rule as follows:

\[
D(\text{rule}) = \mu_A(x_1) \times \mu_B(x_2) \quad \text{(product of membership grade of input \( x_1 \) in fuzzy set \( A \) and membership grade of \( x_2 \) in fuzzy set \( B \)).}
\]

At this point if an expert is available and he assigns his degree of belief in the correctness of particular data set then that degree of belief ‘\( m \)’ must be multiplied with above expression. This will help to choose an appropriate rule when conflicts like two rules with same antecedent part but different consequent result. Then, the rule that has maximum degree are chosen. In case two rules are exactly similar then discard the one with lower degree. After discarding the redundant and contradicting rules, the complete set of rules that constitute the final compact rule base for the system are obtained.

**APPLICATION TO BATTERY CHARGER**

A simple system of fuzzy battery charger is taken. Let the temperature range is 0° to 50° (in terms of corresponding voltage) and the temperature gradient as obtained by taking time derivative of the conditioned signal as obtained from temperature sensor, varied from 0 to 1 mV/10 s. The input and output variables identified for rapid Ni-Cd battery charger along with their universes of discourse are listed in Table 1.

Using Fuzzy Modified C-means clustering (FCM), the temperature range is fuzzified into three membership functions and temperature gradient into two membership functions. Same labels for these membership functions are chosen as shown in Figure 3. Because there are three membership functions for the input ‘temperature’ and there are two inputs for the second input, ie, ‘temperature gradient’ there can be at most \( 3 \times 2 = 6 \) combinations or antecedent parts of the rules as given below. The rule base is yet incomplete as for each rule the consequent need to be found out. From the given dataset of Table 1 it is found that there are only five consequents that form the set of consequents from where one particular element is chosen as the consequent for a particular rule. The specified set of consequents in this case are \( C_1 = \text{trickle} = 0.1 \, \text{A}, C_2 = \text{low} = 1 \, \text{A}, C_3 = \text{medium} = 2 \, \text{A}, C_4 = \text{high} = 3 \, \text{A} \) and, \( C_5 = \text{ultrafast} \).

**Table 1 Input and output variables for rapid Ni-Cd battery charger along with their universes of discourse**

<table>
<thead>
<tr>
<th>Input variables</th>
<th>Minimum value</th>
<th>Maximum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature (( T )),°C</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>Temperature gradient (dT/dt),°C/s</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Output variable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Charging rate (( C_r )), A</td>
<td>0</td>
<td>8C</td>
</tr>
</tbody>
</table>

**Figure 3 Sugeno type fuzzy model for battery charger**
The parameters of antecedent and consequents are chosen in such a way so as to fulfill condition given by expression (3).

As is evident from Figure 3, degree of compatibility of any input data set to rule represented by \( W_i \) can be easily computed using the following formula

\[
W_i = \min(\mu_{\text{LOW}}(\text{temperature}), \, \mu_{\text{LOW}}(\text{temp_grad}))
\]

This way all the \( W_i \) are evaluated, the right hand side of output expression (1) can be evaluated if the proper values for \( C_j \in \{ \text{ULTRAFAST, MED, LOW, HIGH, TRICKLE} \} \) can be chosen.

The BBO algorithm is implemented in MATLAB to select the values of consequents to satisfy the equation (3). It was observed that the algorithm was successfully able to generate the required rule base for the FLS. With the application of rule reduction algorithm as given in [III- B - Step 2-(e)] BBO leads to the following rule set of only four rules:

Rule#1 : If temperature is LOW then \( C_j \) is ULTRA_FAST

Rule#2 : If temperature is MEDIUM and Temp_Grad is LOW then \( C_j \) is HIGH

Rule#3 : If temperature is MEDIUM and Temp_Grad is HIGH then \( C_j \) is MEDIUM

Rule#4 : If temperature is HIGH then \( C_j \) is TRICKLE

It has to be stated clearly, that this rule set is the best result obtained during a few runs of the BBO and it could not always be reached. It is very easy to interpret and has the nice property that the more relevant input 'temperature' is used for all rules, while the less relevant input 'temperature gradient' only appears in two rules in combination with the other input.

**CONCLUSIONS**

Rule extraction from numerical data is a high computational complexity problem. BBO was applied to successfully extract rule base from numerical data. All rules were extracted from the given data for a TSK type zero system. The method appears to be very efficient. Its performance is compared with other optimization approaches, i.e. Ant Colony optimization (ACO), Particle swarm Optimization (PSO), and Genetic Algorithms (GA) based optimization techniques.

BBO appears to be most efficient.

**REFERENCES**


