ELICITING TRANSPARENT FUZZY MODEL USING DIFFERENTIAL EVOLUTION

M. Eftekhari, M. Karimi, S.D. Katebi, A. H. Jahanniri

School of Engineering, Shiraz University, Shiraz, Iran

Tel:+98(711)6271747   Fax:+98(711)6271747  E-mail: katebi@shirazu.ac.ir

Keywords: Differential evolution, Eliciting, Fuzzy Model, Clustering.

Abstract

In this paper a new technique for eliciting a fuzzy inference system (FIS) from data for nonlinear systems is proposed. The strategy is conducted in two phases: in the first one, subtractive clustering is applied to extract a set of fuzzy rules, in the second phase, the generated fuzzy rule base is refined and redundant rules are removed on the basis of an interpretability measure. Finally, centres and widths of the membership functions are tuned by means differential evolution. Case study is presented to illustrate the efficiency and accuracy of the proposed approach.

1 Introduction

In the absence of an accurate first-principle model, which is very difficult to obtain in practice, it is necessary to resort to artificial intelligent techniques, which permit the modelling of a system based on experts’ knowledge as well as on operating data. From the set of techniques available, fuzzy modelling appears to be the most adequate approach, since it contributes to interpretability by allowing a system to be represented by linguistic rules. Clustering - in the input-output space - allows one to obtain a finite set of linguistic rules.

One of the most important goals for developing a fuzzy model is to attain an interpretable system through the easily understandable fuzzy rules for modelling unknown nonlinear systems. Transparency is a measure of the human linguistic interpretability of the rules in a fuzzy model. In many engineering applications transparency is a very important property, since it allows the transformation of data (information) into (human) knowledge. The traditional learning methods for extracting fuzzy rules often suffer from the lack of interpretability, generation of unnecessary fuzzy rules and highly similar membership function for the related fuzzy variables. Thus, there are two types of fuzzy systems namely approximative and descriptive. The former emphasizes on the accuracy of the model and the latter is more interpretable. Recent studies have been focused on improving interpretability while maintaining a rational degree of accuracy [1-3].

Several investigations about the interpretability of fuzzy classification problems have recently been performed [4-6]. Also many studies regarding to the transparency have been reported in the fields of decision making, prediction of process behaviour, data mining and others [7-9].

An automatic data-driven based method for constructing the initial fuzzy models is Chiu’s subtractive clustering [10]. This method has the advantage of avoiding the explosion of the rule base, a problem known as the “curse of dimensionality.” Therefore, clustering based methods would be preferred to the grid partitioning techniques. In this contribution the subtractive clustering technique is used for creating an initial FIS.

Evolutionary algorithms (EAs) have been widely used for eliciting fuzzy models owing to their ability in searching for optimal solutions in an irregular and high dimensional solution space [3,5]. EAs and other stochastic search techniques seem to be a promising alternative to traditional techniques. A particularly promising EA paradigm for continuous search space is the so called differential evolution algorithm, which has gained increasing attention [11]. Recent studies in the field of parameter identification show that DE algorithm outperforms the previously known algorithms [12]. Additionally, DE is much simpler to implement and it requires less parameter tuning. In this work DE is used for adjusting the parameter of the refined FIS model.

2 Fuzzy modelling

The aim of constructing a FIS is to obtain a set of fuzzy rules that describe the system behaviour as accurately as possible, given a set of operating data and if available an initial set of linguistic rules collected from experts. In this paper, a general fuzzy Auto Regressive Exogenous structure is assumed to describe the system. In such a structure, the system is represented by a set of Takagi-Sugeno type rules such as,

\[
\begin{align*}
R: \text{if } & y_1(k) \text{ is } A_{y_1} \text{ and } \ldots \text{ and } y_n(k) \text{ is } A_{y_n} \text{ and } u_1(k-d_1) \text{ is } B_{u_1} \text{ and } \ldots \text{ and } u_m(k-d_m-1) \text{ is } B_{u_m}, \\
\text{then } & y_1(k+1) = f_1(u_1(k-d_1), \ldots, u_m(k-d_m-1), y_1(k), \ldots, y_n(k)) \text{ and } \ldots \text{ and } y_n(k+1) = f_n(u_1(k-d_1), \ldots, u_m(k-d_m-1), y_1(k), \ldots, y_n(k))
\end{align*}
\]  

(1)
where \( q \) denotes the number of system outputs, \( p \) represents the number of inputs and the parameters \( n_1, ..., n_q, m_1, ..., m_p, d_1, ..., d_p \) are related to the system's order and discrete time. \( A_{ji} \) and \( B_{ji} \) are linguistic values for input variables, given by their membership function: 
\[
\mu_{A_{ji}}, \mu_{B_{ji}}, \ i, j = 1, 2, ..., R.
\]
The parameters \( n_1, ..., n_q, m_1, ..., m_p, d_1, ..., d_p \) must be suitably chosen. This may be accomplished on the basis of domain knowledge. Assuming these values are appropriately chosen, the next phase is to obtain a set of rules of type (1) and also refining the rule base and adjusting the parameters of the membership functions using data collected from the system in the following form:
\[
X = [\theta(1) ... \theta(N - 1)]^T, \psi = [\psi(1), \psi(N - 1)]^T
\]
(2)
where \( N \) is the number of data samples available for the identification purposes and \( \theta \) is the regression vector.

3 Subtractive clustering

In order to obtain a set of \( R \) rules and avoiding the problems inherent in grid partitioning based clustering techniques, (i.e. rule base explosion), subtractive clustering is applied [10]. This technique is employed since it allows a scatter input-output space partitioning.

Subtractive clustering is, essentially, a modified form of the Mountain Method. Thus, let \( Z \) be the data set obtained by concatenation of the sets \( X \) and \( \psi \) given by the expressions (2). Assuming that all the data points are normalized in each dimension, the data set \( Z \) is bounded by a hypercube. In the algorithm, each point is seen as a potential cluster centre, for which some measure of potential is assigned according to equation (3):
\[
\psi_i = \sum_{j=1}^{N} e^{-|z_j - z_i|^2} \gamma^2
\]
where \( \alpha = 4/r_n^2 \) and \( r > 0 \) defines the neighbourhood radius for each cluster centre. Thus, the potential associated with each cluster depends on its distance to all the points, leading to clusters with high potential where neighbourhood is dense. After calculating potential for each point, the one with higher potential is selected as the first cluster centre. Let \( z_{i}^\ast \) be the centre of the first group and \( \psi_i^\ast \) its potential. Then the potential for each \( z_{i}^\ast \) is reduced according to equation (4), specially for the points closer to the centre of the cluster:
\[
\psi_i = \psi_i^\ast e^{-|z_j - z_i|^2} \gamma^2
\]
where \( z_j^\ast \) be the location of the first cluster center and \( \psi_j^\ast \) be its potential value. Also \( \beta = 4/r_n^2 \) and \( r_n > 0 \) represent the radius of the neighbourhood for which significant potential reduction will occur. The radius for reduction of potential should be to some extend higher than the neighbourhood radius to avoid closely spaced clusters. Typically, \( r_n = 1.25r_p \). Since the points closer to the cluster centre will have their potential strongly reduced, the probability for those points to be chosen as the next cluster is lower. This procedure (selecting centres and reducing potential) is carried out iteratively until stopping criteria is satisfied. Additionally two threshold levels are defined, one above which a point is selected for a cluster centre and the other below which a point is rejected.

By the end of clustering, a set of fuzzy rules will be obtained. Each cluster represents a rule. However, since the clustering is carried out in a multidimensional space, the related fuzzy sets must be obtained. As each axis refers to a variable, the centres of the member ship functions are obtained by projecting the centre of each cluster in the corresponding axis. The widths are obtained on the basis of the radius \( r_n \).

4 Differential evolution

Recently proposed Differential Evolution (DE) [11] is a search algorithm that is increasingly becoming popular among researchers. It deals with a real coded population, instead of a binary or a Gray representation, and devises its own crossover and mutation in the real space. Differential Evolution, like Genetic Algorithms, belongs to the family of Evolutionary Computation and uses the concept of fitness in the same sense as in Canonical Genetic Algorithms (CGA). However, there are some major philosophical differences between CGA and DE. Although DE uses a population based computing strategy, unlike CGA, here a real parameter representation is used and an individual is formed by a vector array of all the variables in the problem. DE uses both crossover and mutation. However, both operations need to be redefined in the DE context. DE attempts to create \( p^\ast \), a mutated form of any individual \( p \), using the vector difference of randomly picked individuals.

There are several variants of the original DE, the particular one used in this work is the simple algorithm proposed by [11] in which the vector difference of two randomly selected individual \( p^\ast \) and \( p^\prime \) are used such that:
\[
p^\ast = p + \gamma(p^\prime - p^\ast)
\]
(5)
where \( \gamma \) is a user-supplied scaling factor, usually kept between 0 and 2. In DE this operation is known as mutating with vector differentials. Next, the crossover is applied between any individual member of the population \( p^\ast \) and the mutated vector \( p^\prime \), which is done essentially by swapping the vector elements in the corresponding location. Similar to CGA this is also done probabilistically and the decision of doing (or not doing) crossover is monitored by a crossover rate \( \eta \), \( 0 \leq \eta \leq 1 \). The new vector \( p^\prime \), produced this way, is known as the trial vector. It has to be made sure that the trial vector inherits at least one variable from the mutated vector \( p^\prime \), so that it does not become an exact replica of the original parent vector. In DE the trial vector is allowed to pass on to the next generation if and only if, its fitness is higher than that of its parent vector \( p^\ast \). Otherwise, the parent vector proceeds to the next generation. The crossover and mutation procedures are more elaborated in Figure (1).
Figure 1. Cross over and mutation in DE

Differential evolution, as described above, is a greedy scheme, because one of the parents is always compete against its own offspring for a berth in the next generation. It is known to converge very fast for many applications and some of its recent applications in the control design and analysis related problems.

5 Interpretability measure

Similar fuzzy sets represent almost the same region in the universe of discourse of a fuzzy variable; i.e., they describe the same concept. Therefore, the first step in attaining model interpretability consists of finding and merging similar membership functions. Structural learning by means of subtractive clustering technique leads to a set of initial membership functions, some of which may be highly similar, resulting in some unnecessary and redundant rules. Thus, the model will lack transparency and it seems useful to merge similar membership functions.

Let A and B be two fuzzy sets. The measure of similarity of these sets may vary from 0, indicating completely distinct to 1 indicating completely similar. The most common similarity measure for fuzzy sets given in the literature [13] is based on the intersection and union operations and is defined as follows:

\[ S(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \]  

(6)

Where \( S \) is the similarity measure and \(| \cdot |\) indicates the size of a set, and intersection and union operators are denoted by \( \cap \) and \( \cup \) respectively [13]. The implementation of this measure in a continuous universe of discourse will prove computationally intensive, particularly for Gaussian MF which are produced by subtractive clustering. Therefore, some simplification techniques for calculating the similarity of fuzzy sets have been suggested, for example, triangular membership function with centre \( c \) and width \( \sigma \sqrt{\pi} \) is utilized by [14] for similarity evaluation. This method is adapted in this work for calculation of the similarity between two fuzzy sets.

If the similarity measure for two fuzzy sets is greater than a predefined threshold, both sets are replaced by a new fuzzy set having parameters given by the following equations

\[ c_{new} = \frac{\sigma_1 c_1 + \sigma_2 c_2}{\sigma_1 + \sigma_2} \]  

(7)

\[ \sigma_{new} = \frac{\sigma_1 + \sigma_2}{2} \]  

(8)

Consider a system with \( n \) inputs and one output and assuming that the three following rules have been generated by employing subtractive clustering.

\[ R_1 : \text{if } x_1 \text{ is } mf_1 \text{ and } \ldots \text{ and } x_n \text{ is } mf_n \text{ then } k(x_1, x_2, \ldots, x_n) \]  

\[ R_2 : \text{if } x_1 \text{ is } mf_1' \text{ and } \ldots \text{ and } x_n \text{ is } mf_n' \text{ then } k(x_1, x_2, \ldots, x_n) \]  

\[ R_3 : \text{if } x_1 \text{ is } mf_1'' \text{ and } \ldots \text{ and } x_n \text{ is } mf_n'' \text{ then } k(x_1, x_2, \ldots, x_n) \]  

(9)

\[ k(x_1, x_2, \ldots, x_n) = a_{11}x_1 + a_{21}x_2 + \ldots + a_{n1}x_n + a_{nl} \]  

(10)

Where \( mf_j \) indicates the \( j \)-th MF on the \( i \)-th input. The antecedent of \( j \)-th rule is constituted from the conjunction (and) of \( j \)-th MFs on each of \( n \) inputs. \( a_{ij}, a_{2j}, \ldots, a_{nj}, a_{nj} \) are the output parameters of the Takagi-Sugeno (TS) FIS. If the third MF of each input \( mf_j \) variable merges with the other MFs on that input then the third rule constructing from the conjunction of the third MFs will be omitted. In this manner redundant rules may be removed from the rule base during the MF merging process. The procedure of membership function merging, one pair per iteration, continues until no more pairs satisfy the merging threshold. The membership function created after merging is a merging candidate in the following iterations.

6 Generating interpretable fuzzy models

The proposed procedure for generating a comprehensible fuzzy inference system is shown in Figure (2).

![Diagram](image)

Figure 2. Procedure for Generating Interpretable FIS

The subtractive clustering method is used for creating an initial FIS model. The FIS is refined and redundant rules are removed by the procedure described above, the DE is utilized for parameter tuning.
After terminating the above procedure, the simplified FIS are finally tuned once more by DE in order to obtain an appropriate degree of suitable accuracy (i.e. acceptable Mean Square Error (MSE)). A threshold may be imposed on the required degree of accuracy which can result in the termination of the simplifying procedure and consequently the whole procedure.

7 Simulation results

The parameters for various algorithms are initialized as follows. The population size and generation numbers are set to $ps = 50$ and $G = 50$ respectively. For DE parameters, the cross over probability, $pc = 0.8$, the differential scaling factor $\gamma = 0.2$, and the radius of subtractive clustering algorithm is set to 0.8. The other parameters are taken as their default values.

The process is a Continuous Stirring Tank Reactor (CSTR), where the reaction is exothermic and the concentration is controlled by regulating the coolant flow. Schematic diagram of continues stirring tank reactor is shown in Figure (3).

In figure (3), $F_0, C_{A0}$ and $T_0$ are feed flow rate of mass, inlet feed concentration and inlet feed temperature respectively and $F_i, T_i$ are inlet coolant mass flow and temperature, $V, T_c, C_A, F$ and $T_j$ are reactor volume, temperature, concentration, output feed flow rate and output coolant temperature. In this case, the input is coolant flow to jacket and the outputs are product concentration and temperature. The numbers of data taken from [15] as public domain bench mark is 7500 and the sampling time is 6 seconds. Seventy percent of the data pairs are selected randomly for training and the rest is used for test.

Figure (4) shows the final rule produced after the simplification, representing the CSTR model. Figure (5) shows the corresponding MFs of the antecedent for the resulting rule.

The solid curve in figure (6) depicts the actual and the dotted curve represents the resulting behaviour of the obtained model for the first output in response to a PRBS input. Also the MSEs for the trained and the tested data are given as, (MSE_train=1.071e-005, MSE_Test=6.5058e-008). Similar responses for the second output, $y_2(t)$ is given in the figure (7) and the degree of accuracy measured by the MSE for this output is calculated as, MSE_train =0.3028 MSE_Test =0.0033.

As can be seen from figures (6 and 7), the model obtained by the proposed approach is described by the minimum number of rule i.e. one rule. Further, it is quite accurate and compares well with the actual model given by the bench mark data set.

If $(u(t-1)$ is Middle) and $(u(t)$ is Middle) and $(y_1(t-2)$ is Big) and $(y_1(t-1)$ is Middle) and $(y_2(t-2)$ is Middle) and $(y_2(t-1)$ is Middle) then $y_1(t)= 0.0519u(t-1)-0.0003u(t)-0.8121y_1(t-2)+1.7471y_1(t-1)\cdot 0.0226$ and $y_2(t) = -0.1239u(t-1)+0.0060u(t)-0.6339y_2(t-2)+1.4975y_2(t-1)+0.0228$

Figure 4. Extracted rule for CSTR system

Figure 5. MFs of antecedent of the above rule

Figure 6. the first actual and model output

Figure 7. the second actual and model output

8 Conclusion

An iterative algorithm with two steps is employed to produce a simplified and an optimum FIS in terms of number of fuzzy sets and number of rules. In the first step, the parameters of the model are adjusted by utilizing DE. In the second step, similar membership functions are merged. These
two steps are performed iteratively until no more pairs of membership function satisfy the merging criterion. Subtractive clustering as an automatic data-driven method is utilized for constructing the initial FIS. The capability of DE for tuning a FIS of the TS type is very promising. Results obtained illustrate that the proposed approach is efficient in terms of accuracy and transparency of the model. Moreover, it is easy to implement, having only a few control parameters which remain fixed throughout the entire optimization procedure and uses DE as a parallel direct search technique.

References


