Incremental Fuzzy Rule Base Extension with Optimization[1]

Johanyák, Zsolt Csaba [2] – Kőházi-Kis, Ambrus[3]

A sparse fuzzy rule base offers low complexity and low memory demand for a fuzzy system. Its automatic generation from sample data involves two main tasks, i.e. the structure definition and the parameter identification. In this paper, we present a novel approach that starts with two rules and incrementally creates new rules followed by the identification of their parameters using a direct search method.

1 Introduction

One of the key steps in fuzzy model identification is the creation of the rule base. In several cases there is no human knowledge that could be incorporated in form of predefined linguistic terms and fuzzy rules. Therefore the model is generated automatically from sample data. Most of the known methods create dense rule bases, which can lead to a rule number explosion in case of high number of dimensions and high number of sets/dimension.

The RBE-DLS (rule base extension with direct local search) method ensures a trade-off between the demand on approximation capability of the fuzzy system and the demand on low complexity of the rule base by generating a sparse rule base. It follows the concept of rule base extension [7] and identifies the parameters of the fuzzy system using a direct local search method.

The rest of this paper is organized as follows. Section 2 presents the applicable membership function parameterization approaches. Section 3 introduces the new method after reviewing the concepts of sparse rule bases and rule base extension. Section 4 presents some experimental results applying RBE-DLS and the conclusions are drawn in section 5.

---

[1] Lektorált tanulmány

[2] PhD, főiskolai docens, KF GAMF Kar, Kalmár Sándor Informatikai Intézet, Informatika Szakcsoport

[3] PhD, főiskolai tanár, KF GAMF Kar, Természet- és Műszaki Alaptudományi Intézet, Fizika Szakcsoport

2 Parameterization

Each membership function can be described by a smaller or larger number of parameters depending on the shape type. The piece-wise linear membership functions can be described easily by the position of the break-points. For example in case of a singleton the parameter is the element of the universe of discourse whose membership value is greater than zero; or in case of a triangle shaped normal fuzzy set (e.g. fig. 1) the parameters are the abscissa values of the three vertices

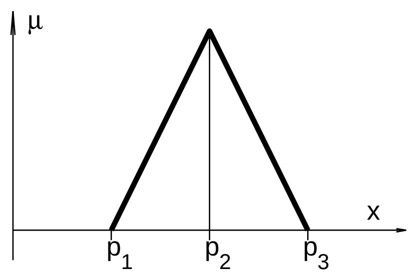

Fig. 1 Parameterization of a triangle shaped membership function

The number of the parameters determines the number of variables whose values are changed in course of the fuzzy model identification, which has a strong effect on the time need of the process. In several cases one might use uniform shaped fuzzy sets in order to reduce the time demand. Thus only one parameter has to be adjusted in case of each linguistic term. This parameter is the position of the set described by the reference point. Usual choices for this task are (fig. 2) abscissa values corresponding to the centre of the core (RP_{CC}, e.g. [1][3][6]), the centre of gravity (RP_{GC}, e.g.[5]) and the centre of the support (RP_{SC}, e.g. [3]).

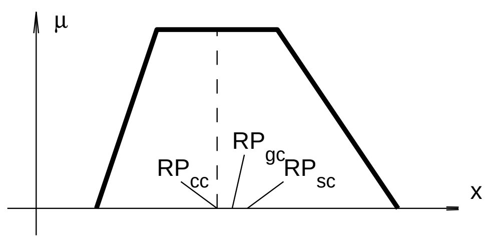

Fig. 2 Usual reference point types

Although the application of the uniform shaped sets reduces the time consumption of the tuning and preserves the good interpretability of the fuzzy rules it also can have a negative side effect by reducing the performance of he fuzzy system (see section 3.5 on details about the performance measurement). Thus the selection of the parameterization is a trade-off between the performance of the system and the cost of the model identification.

3 Model identification from numerical sample data

Fuzzy model identification usually consists of the following two steps.

1. Data preprocessing that could include the identification of the input and output linguistic variables, determination of the lower or upper bounds of each dimension of the input and output universes of discourse, statistical analysis of the data regarding the relevance of the input variables excluding the non relevant ones in order to reduce the complexity of the system.

2. Rule base generation that includes the definition of the input and output partitions as well as the extraction of the rules from the sample data. In course of the rule base generation one can follow two different approaches. The first one divides the task in two separate steps, i.e. the structure definition and the parameter identification (e.g. Precup, Doboli and Preitl [16]; or Botzheim, Hámori and Kóczy [2], or Škrjanc, Blažič and Agamennoni [19]).

   The second approach works incrementally by simultaneously modifying the structure and the parameters, i.e. introducing or eventually eliminating rules and tuning the parameters of the membership functions (e.g. Johanyák and Kovács [7]). This approach also can be applied in case of adaptive fuzzy systems (e.g. Vaščák, and L. Madarász [21])

The method being presented in sections 3.3 and 3.4 covers the second step and follows the second approach.

3.1 Sparse rule base

The rule base of a fuzzy system is categorized as dense (covering) or sparse (non-covering) depending on the coverage of the input space by rules, which is defined by the formula

$$c = \arg\max_{\varepsilon}\Big(\min_{i=1}^{N}\Big\{ \max_{j=1}^{n_i}\Big\{ t\big(A_{i,j}, A_i^*\big) \Big\} \Big\} \geq \varepsilon, \\ \forall A_i^* \subset X_i \Big), \varepsilon \in [0,1], \tag{1}$$

where X_i is the ith dimension of the antecedent space, A_i^* is the fuzzy set describing the observation in the ith antecedent dimension, A_{ij} is the jth linguistic term of the ith antecedent dimension, t is an arbitrary t-norm, n_i is the number of the linguistic terms of the *i*th antecedent dimension, *N* is the number of the antecedent dimensions, and *argmax(.)* calculates the ε value for which the expression in the parentheses takes its maximum. If $c > \varepsilon_0$ the rule base is called ε_0 covering (dense) otherwise it is considered sparse.

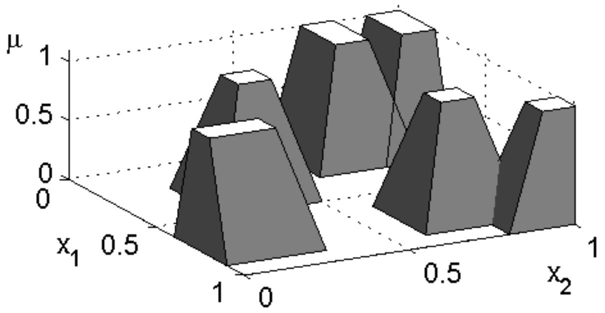

Fig. 3 Sparse antecedent space

If there is no demand on an $\varepsilon>0$ value the rule base is considered sparse when there is at least one possible input value for which the rule base does not contain an applicable rule.

3.2 Fuzzy inference in sparse rule bases

Fuzzy systems applying sparse rule bases have to use approximate inference techniques that can cope with the lack on rules in some regions of the input space. For this task the most used techniques are the fuzzy rule interpolation based ones. They form two main groups based on the key ideas they are using.

The members of the first group, the so called one-step methods determine the conclusion directly from the observation taking into consideration two or more existent rules of the rule base. The methods KH [20], FIVE [10], IMUL [22], IRG [3], and Kovács's method [9] belong to this category.

The members of the second group first produce a new rule in the position of the observation using rule interpolation and next, they determine the conclusion by firing the interpolated rule. Here belong for example the methods GM [1], IGRV [5], LESFRI [6], as well as Chen and Ko's method [4].

3.3 RBE-DLS

The rule base extension using direct local search (RBE-DLS) method aims the generation of a fuzzy rule base from numerical sample data. The data consist of known input and output value pairs. The input could be one- or multidimensional, while the output has to be one-dimensional. In case of a multidimensional output a separate rule base can be generated for each output dimension.

The basic idea of the Rule Base Extension (RBE) is that one creates first an initial rule base and next, one starts an iterative tuning process when beside the adjustment of the values of the known sets' parameters new linguistic terms and rules are introduced into the rule base.

The initial rule base contains only two rules, one describing a maximum point of the output and one describing a minimum point of the output. First one seeks the two extreme output values and a representative data point for each of them. If several data points correspond to an extreme value, one should select the one that is closer to an endpoint of the input domain.

The reference points of the antecedent sets of the first rule will be identical with the corresponding input values of the minimum point. The reference point of the consequent set will be identical with the output value of the minimum point. The shape of the linguistic terms is determined by the default set shape, which is a characteristic feature of the partition. The antecedent and consequent linguistic terms of the second rule are determined in a similar way taking into consideration the maximum point. At this point the system contains two linguistic terms in each dimension.

Having the first two rules determined, next a parameter identification process is started, which iteratively adjusts the values of the linguistic terms' parameters. The details of the applied algorithm are presented in the next section. If the improvement velocity of the fuzzy systems' performance index falls below a specified threshold or even stops after an iteration cycle a new rule is generated. It is because the system tuning reached a local optimum of the performance indicator and the performance cannot improve further by the applied parameter identification algorithm. The new rule introduces additional tuning possibilities. However, in some cases the performance will deteriorate temporarily after the insertion of the new rule into the rule base.

In order to create the new rule, one seeks for the calculated data point, which is the most differing one from its corresponding training point. The input and output values of this training point will be the reference points of the antecedent and consequent sets of the new rule. The shapes of the new linguistic terms are determined using the default shape types of the corresponding partition.

Further on, the last two steps (parameter adjustment and new rule creation) are repeated until the specified iteration number has been reached, or the value of performance index overcomes a prescribed threshold.

3.4 Parameter identification using a direct search method

The aim of the parameter identification process is to improve the model by optimizing its performance evaluated by the so called merit function. In case of fuzzy systems' optimization the merit function is also called performance index (see section 3.5 for its definition). Owing to the nonalalytic behaviour of the merit function one can use effectively only so called direct search methods for optimization.

The best known direct serch method is a so called simplex method invented by Nelder and Mead in 1965 [15] [18]. It is a very effective method in case of a lot of problems having low number of dimensions. It is widely used even nowadays although this method sometimes fails to find the local optimum also for well behaving (two times continuously differentiable) functions [14]. One of the authors tested this algorithm for different dimensional analytic problems of highly reflecting dielectric mirror design [11] starting from a randomly chosen point in the parameter space. This method proved to be statistically stable up to approximately 6-8 dimensions. In higher dimensions the method of Nelder and Mead characteristically loses its way to the local optima. That is this method is not expected to solve our problem in the parameter space with dimension of significantly higher number than 6-8.

Recently direct search methods have been intensively investigated (e.g. in [8]) also in mathematical analysis. A well defined class of direct search algorithms has been identified [8] that can be proved to be convergent for maximization problems with orderly merit functions (that is at least two times differentiable). The generalized algorithm is called Generating Set Search (GSS) method. It must fulfill some conditions for the sake of convergence but there are plenty of features which are freely variable for optimizing the speed of the convergence of the algorithm [8].

We developed a GSS search code in which optional exploratory moves are used to fasten the convergence of the algorithm [11][13]. The convergence of the algorithm is guaranteed by the search on a minimal positive base whose elements form a regular simplex. The elements of this positive base can be obtained by the following recursive formula (for $N = 2, 3, \dots$):

$$A_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad \dots \quad A_N = \begin{pmatrix} 1 & 0 \; \dots \; 0 \\ -\frac{1}{N} & \\ \vdots & \left(\frac{\sqrt{N^2-1}}{N} A_{N-1} \right) \\ -\frac{1}{N} & \end{pmatrix} \tag{2}$$

The algorithm can be described as follows:

Initialization.

Let $f : R^N \to R$ *given.*

Let $x_0 \in R^N$ *be the initial guess.*

Let $\Delta_{tol} > 0$ *be the step-length convergence tolerance.*

Let $\Delta_0 > \Delta_{tol}$ *be the initial value of the step-length control parameter.*

Let G be a positive base, a grenerating set for R^N *given by the recursive formula of eq. (2). (In this case the lower bound of the cosine measure of the generating set:* $\kappa(G) = 1/N$ *.)*

Algorithm. *For each iteration k=0, 1, 2, 3…*

Step 1. *Let* $G_k = \{\Delta_k d \mid d \in G\}$ *be the set of trial steps, and* $H_k = \{s_k + \Delta_k d \mid d \in G\}$ *be the set of explanatory moves. Here* s_k *is a last successful step:* $s_k = x_k - x_{k-1}$*. (but* $s_0 = 0$*)*

Step 2. *If there exists* $d_k \in H_k$ *such that* $f(x_k + d_k) < f(x_k)$ *than set*

- Set $x_{k+1} = x_k + d_k$ *(change the iterate).*

- If $s_k > 12\Delta_k$ *than set* $\Delta_{k+1} = 2\Delta_k$*.*

Step 3. *Otherwise (now* $f(x_k + d_k) \geq f(x_k)$ *for all* $d_k \in H_k$*)*

Let $d_{k\,\max} \in H_k$ *such that* $f(x_k d_{k\,\max}) \geq f(x_k d_k)$ *for all* $d_k \in H_k$*.*

If $f(x_k - d_{k\,\max}) < f(x_k)$ *than*

- Set $x_{k+1} = x_k - d_{k\,\max}$ *(change the iterate),*

- If $s_k > 12\Delta_k$ *than set* $\Delta_{k+1} = 2\Delta_k$*.*

Step 4. *Otherwise if there exists* $d_k \in G_k$ *such that* $f(x_k + d_k) < f(x_k)$ *than*

- Set $x_{k+1} = x_k + d_k$ *(change the iterate) and set* $s_k = d_k$*.*

Step 5. *Otherwise*

Let $d_{k\,\max} \in G_k$ *such that* $f(x_k d_{k\,\max}) \geq f(x_k d_k)$ *for all* $d_k \in G_k$*.*

If $f(x_k - d_{k\,\max}) < f(x_k)$ *than*

- Set $x_{k+1} = x_k - d_{k\,\max}$ *(change the iterate) and set* $s_k = -d_{k\,\max}$*.*

Step 6. *Otherwise do the following:*

- Set $x_{k+1} = x_k$ *(no change the iterate).*

- Set $\Delta_{k+1} = 0.5\Delta_k$ *(contract the step-length control parameter).*

- If $\Delta_{k+1} < \Delta_{\mathrm{tol}}$ *then terminate.*

This algorithm has been implemented in C++ and MATLAB, and it was successfully tested on different problems (e.g. [11] and [12]).

3.5 Performance index

The performance index expresses the quality of the approximation ensured by the fuzzy system using a number that aggregates and evaluates the differences between the prescribed output values and the output values calculated by the fuzzy system. One can choose from several possible performance indices available in the literature (e.g. in [17]). We used the root mean square of the error (RMSE) as performance index owing to its good comprehensibility. and comparability to the range of the output linguistic variable. Its value is calculated by

$$RMSE = \sqrt{\frac{\sum_{j=1}^{M} \left(y_j - \hat{y}_j\right)^2}{M}}, \tag{3}$$

where M is the number of training data points, y_j is the output of the j^{th} data point and $\hat{y}_j$ is the output calculated by the system.

4 Experimental results

For testing purposes we considered first a nonlinear one-dimensional function presented in fig. 4 and we applied the LESFRI fuzzy inference technique. The sample data contained 101 uniformly distributed data points whose abscissa values were in the interval [0,10]. The sample data was split randomly into two sets, one containing 68 data points for training purposes and one containing 33 data points for testing purposes. In order to avoid the overfitting of the fuzzy system to the training data points we also evaluated in course of the parameter optimization the performance of the fuzzy system against the testing data.

Finally, we selected that parameter tuple which ensured a quasi optimal performance in case of the testing data as well. Thus the resulting fuzzy system contained 5 rules, and the value of the performance index was RMSE=0.0967 in case of the testing data and RMSE=0.0771 in case of the testing data.

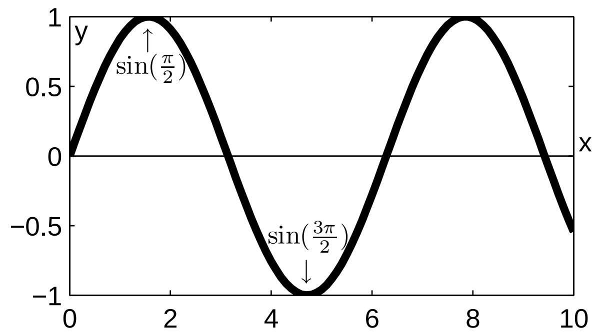

Fig. 4 One-dimensional test function

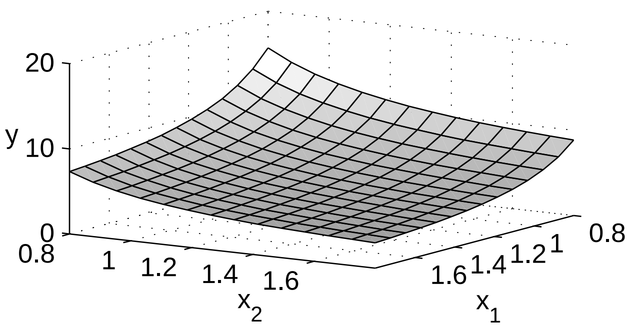

Fig. 5 Two-dimensional test function

Next, we used a two-dimensional nonlinear test function presented in fig. 5. Applying the above described considerations in this case the training data sample contained 130 data points and the testing data sample contained 65 data points. The final parameter tuple was selected based on the same trade-off between the performance of the system against the training and testing data sets as in the case of the first experiment.

The resulting fuzzy system contained 19 rules, and the value of the performance index was RMSE=0.6600 in case of the training data and RMSE=0.6749 in case of the testing data.

5 Conclusions

The experimental results showed that the presented method was able to produce a low complexity sparse rule base in both cases. However, the approximation capability of the resulting systems was slightly worse than the results obtained using the default set shape and hill climbing approach based technique published in [7].

Acknowledgment

This research was supported by Kecskemét College GAMF Faculty grant no: 1KU31, and the National Scientific Research Fund Grant OTKA K77809.

REFERENCES

[1] P. Baranyi, L.T. Kóczy, and T.D. Gedeon, “A Generalized Concept for Fuzzy Rule Interpolation,” in IEEE Transaction on Fuzzy Systems, ISSN 1063-6706, Vol. 12, No. 6, 2004, pp 820-837.

[2] J. Botzheim, B. Hámori, and L.T. Kóczy, “Extracting trapezoidal membership functions of a fuzzy rule system by bacterial algorithm”, in Proceedings of the International Conference 7th Fuzzy Days, Dortmund, Germany, 2001, Lecture Notes in Computer Science, Springer-Verlag, 2206, pp. 218-227.

[3] B. Bouchon-Meunier, C. Marsala, and M. Rifqi, “Interpolative reasoning based on graduality,” in Proceedings of the FUZZ-IEEE'2000, San Antonio, USA, 2000, pp. 483-487.

[4] S.M. Chen and Y.K. Ko, “Fuzzy Interpolative Reasoning for Sparse Fuzzy Rule-Based Systems Based on α-cuts and Transformations techniques,” IEEE Trans. on Fuzzy Systems, Vol. 16, No. 6, 2008, pp. 1626-1648.

[5] Z.H. Huang, and Q. Shen, “Fuzzy interpolation with generalized representative values,” in Proceedings of the UK Workshop on Computational Intelligence, 2004, pp. 161-171.

[6] Z.C. Johanyák and S. Kovács, “Fuzzy Rule Interpolation by the Least Squares Method,” in Proceedings of the 7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI 2006), Budapest, Hungary, 2006, pp. 495-506.

[7] Johanyák, Z. C., Kovács, S.: Sparse Fuzzy System Generation by Rule Base Extension, 11th IEEE International Conference of Intelligent Engineering Systems (IEEE INES 2007), June 29 - July 1, 2007, Budapest, pp. 99-104

[8] T.G. Kolda, R.M. Lewis, V. Torczon, „Optimization by direct search: New perspectives on some classical and modern methods”, SIAM Review, 45, 2003, pp. 385-482.

[9] L. Kovács, “Rule approximation in metric spaces,” Proceedings of 8th IEEE International Symposium on Applied Machine Intelligence and Informatics SAMI 2010, Herl'any, Slovakia, pp. 49-52.

[10] S. Kovács, “Extending the Fuzzy Rule Interpolation "FIVE" by Fuzzy Observation,” Advances in Soft Computing, Computational Intelligence, Theory and Applications, Bernd Reusch (Ed.), Springer Germany, 2006, pp. 485-497.

[11] A. Kőházi-Kis, „A klasszikus Nelder-Mead és egy újonnan kifejlesztett optimumkereső eljárás teljesítményének összehasonlítása”, AGTEDU 2008, Kecskemét, II. kötet, pp. 55-60.

[12] A. Kőházi-Kis, „Longitudinálisan pumpált lézerek optimális gerjesztése”, AGTEDU 2008, Kecskemét, II. kötet, pp. 61-66.

[13] A. Kőházi-Kis, „Praktikus direkt kereső eljárások”, Felsőfokú alapképzésben matematikát, fizikát és informatikás oktatók XXXII. konferenciája, Kecskemét, 2008.

[14] K.I.M. McKinnon, “Convergence of the Nelder-Mead simplex method to a nonstationary point”, SIAM J. Optim, 9, 1998, pp. 148-158.

[15] J.A. Nelder, R.A. Mead, “A simplex method for function minimalization”, Computer Journal, 7, 1965, pp. 308-313.

[16] R.E. Precup, S. Doboli and S. Preitl, “Stability analysis and development of a class of fuzzy systems,” in Engineering Applications of Artificial Intelligence, vol. 13, no. 3, June 2000, pp. 237-247.

[17] R.E. Precup and S. Preitl, “Optimisation criteria in development of fuzzy controllers with dynamics,” Engineering Applications of Artificial Intelligence, vol. 17, no. 6, Sep. 2004, pp. 661-674. 17

[18] W.H Press, S.A. Teulkowsky, W.T. Vetterling, B.P. Flannery, “Numerical Recipies in C, The art of scientific computing”, Cambridge University Press, 1992.

[19] I. Škrjanc, S. Blažič and O. E. Agamennoni, Identification of dynamical systems with a robust interval fuzzy model,” in Automatica, vol. 41, no. 2, Feb. 2005, pp. 327-332.

[20] D. Tikk, I Joó, L T Kóczy, P Várlaki, B Moser and T D Gedeon: Stability of interpolative fuzzy KH controllers,INT J FUZZ SET 125:5-119 (2002).

[21] J. Vaščák, and L. Madarász, “Automatic adaptation of fuzzy controllers”, Acta Polytechnica Hungarica, Journal of Applied Sciences at Budapest Tech, Hungary, Vol. 2, No. 2, 2005, pp. 5-18.

[22] K.W. Wong, D. Tikk, T.D. Gedeon and L.T. Kóczy, “Fuzzy rule interpolation for multidimensional input spaces with applications”. IEEE Trans on Fuzzy Sys. , 13(6), 2005, pp. 809–819.

Fuzzy modell generálása szabálybázis kiterjesztés és lokális közvetlen keresés segítségével

Johanyák Zsolt Csaba – Kőházi-Kiss Ambrus

Összefoglaló

A ritka fuzzy szabálybázisok alkalmazása lehetővé teszi a rájuk épülő fuzzy rendszerek komplexitásának valamint memóriaigényének csökkentését. Mintaadatok alapján történő létrehozásuk két fő feladatot foglal magába, a szerkezet definiálását és a paraméterek beazonosítását.

Cikkünkben egy új módszert ismertetünk ezen feladatok megoldásaként, amely két kezdő szabály létrehozását követően inkrementálisan bővíti a szabálybázist. Minden új szabály beillesztése után a rendszer paramétereinek kvázi-optimális értékeit egy lokális közvetlen keresési eljárás segítségével állapítjuk meg.

Fuzzy Regelbasenherstellung durch RBE und eine direkte Suchmethode

Johanyák, Zsolt Csaba – Kőházi-Kiss, Ambrus

Zusammenfassung

Dünnbesetzte Fuzzy Regelbasen sichern die geringe Komplexität und geringen Speicherbedarf der Fuzzy-Systeme. Ihre Herstellung von Sample-Daten beinhaltet im Wesentlichen zwei Aufgaben, nämlich die Definition der Struktur und die Identifikation der Parameter.

In diesem Beitrag stellen wir eine neuartige Methode, die mit zwei Regeln beginnt und schrittweise schafft neue Regeln, deren Parametern werden durch eine direkte Suchmethode identifiziert.