CISUC – CENTRE OF INFORMATICS AND SYSTEMS OF

UNIVERSITY OF COIMBRA

Adaptive Computation Group

CLASSE Project

eFSLab

Project Report

Lara Cristina Cordeiro Aires

António Dourado

Coimbra, November 2008

UNIAO EUROPEIA
FSE/FEDER

POS_CONHECIMENTO
Programa Operacional Sociedade do Conhecimento

Supported by FCP Project POSC/EIA/58162/2004

ABSTRACT

A software lab is presented to support the development of fuzzy systems from data (data-driven approach) avoiding redundancy and unnecessary complexity in the obtained membership functions, in order to give some semantic meaning to the results. On-line mechanisms for merging membership functions and rule base simplification are implemented improving interpretability and transparency of the produced fuzzy models, allowing the minimization of redundancy and complexity of the models during their development, contributing to the transparency of the obtained rules. It allows to create TS model (order zero and one) and transform the TS order 0 models into Mamdani models, in a simple and friendly way, accessible to less expert users.

The application, developed in Matlab environment, will be public under GNU license.

Taking into account all its capabilities, rise the idea of implementing this application in a fuzzy systems creation lab,

eFSLab arise – A friendly-user tool for creating fuzzy systems with many capabilities, both for their use in scientific projects, both in teaching the fuzzy subject.

TABLE OF CONTENTS

List of Figures

LIST OF TABLES

List of Symbols, Abbreviations and Nomenclature

1. INTRODUCTION

Data-driven methodologies for automatic generation of computational models are probably one of the most important tools still to be developed to use the immense quantity of information available nowadays [1] in the information systems. In most applications (finance, busyness, industry, medicine, etc.) these methodologies should be iterative, to process the data as it is being reported in real time, and transparent, building a linguistic model clearly interpretable by humans. The eTS - Evolving Takagi Sugeno Systems is one of those methodologies [2] [3].

Traditionally the most important property of a fuzzy system has been its accuracy in representing the real system (for simulation, prediction, decision making, etc.).However the obtained fuzzy systems become frequently without practical utility because it is impossible to give some semantic meaning to its rules due to fuzzy sets superposition, rules sometimes redundant, sometimes contradictory, frequently with high complexity. In recent years interpretability has been considered as the key feature of fuzzy models [4] [5] [6] [7] and can be pursuit by rule base simplification and reduction. There is actually a considerable activity concerning this challenging problem. Several perspectives are being developed, for example by fuzzy set merging using entropy measures [8], by genetic optimization [9][10][11][12], multiobjective evolutionary algorithms [13] [14] [15], by Radial Basis Function Networks [16].

A pruning technique, based on similarity measures, is used here to reduce the degree of redundancy and unnecessary complexity arriving in the automated building of fuzzy rules. This improves the human semantic interpretability and as a consequence the usefulness of the results, allowing the merging of compatible fuzzy sets and possibly reduction of the number of rules and features. The fuzzy system is based on an improved version of the eTS algorithm of [3] and strengthening its capability to spread rules over all the reachable state space [17]. Pruning of the rules is based on fusion of the antecedents fuzzy sets, depending on a similarity threshold, and rule elimination based on similar antecedents. The advantage of similarity measures, principally the geometric ones, relies in its simplicity, from the computational point of view, that is explored in this work.

2. eFSLab

2.1. evolving_TS Algorithm Development

In *eFSLab* was implemented the *evolving_TS* algorithm. This was based on Angelov approach for on-line learning of Takagi-Sugeno (TS) fuzzy models.

2.2. eFSLab Interface

eFSLab was developed in Matlab 7.6.0 (R2008a) and uses the Fuzzy Logic Toolbox 2.2.6.

It was tried to produce an interface as complete as possible, where the user could set the great majority of parameters needed to create a Takagi-Sugeno fuzzy system and transform it into a Mamdani system, in a simple way.

So, *eFSLab* is the main interface in which Takagi-Sugeno (TS) fuzzy systems are created and *SugenoToMamdani* is an *eFSLab* attached interface in which it is possible to transform a TS fuzzy system created in a Mamadani one.

eFSLab and *SugenoToMamdani* are represented in Fig. 1 and 2, respectively.

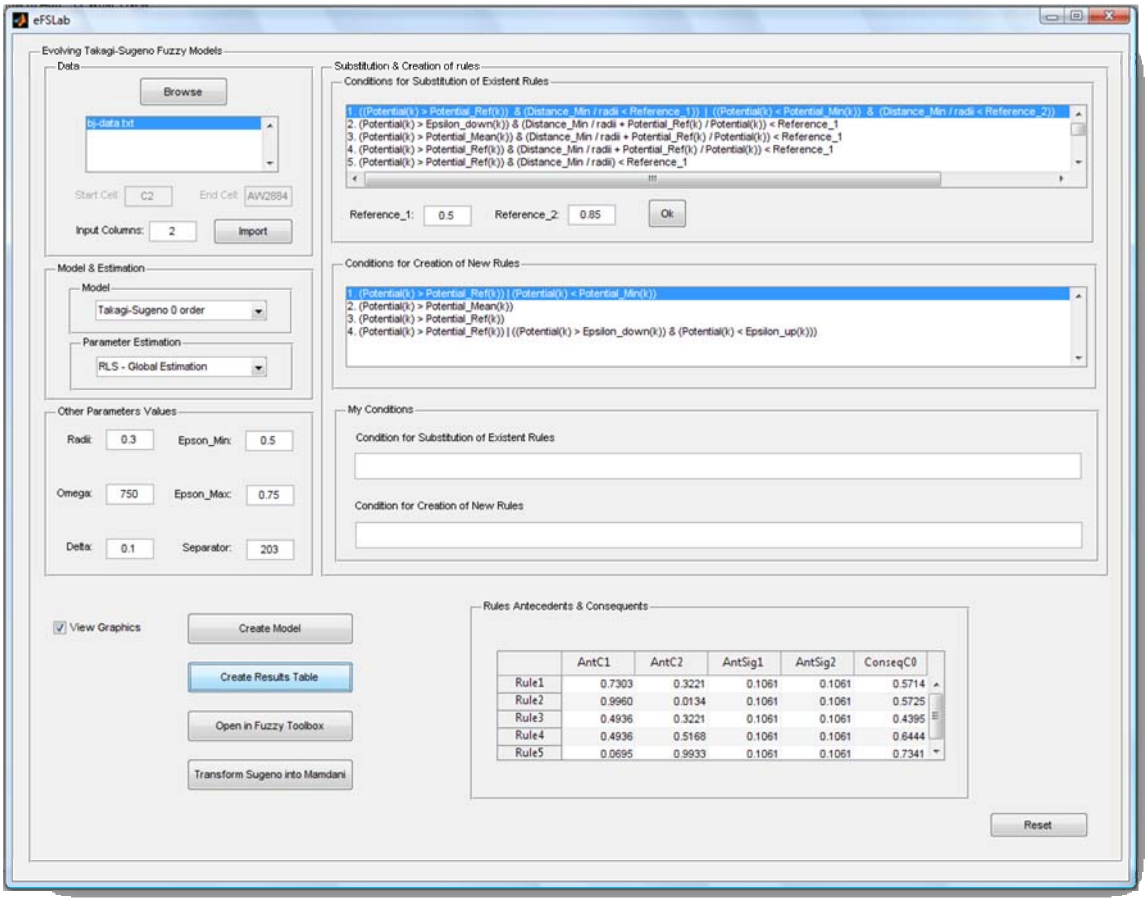


Figure 1: *eFSLab* interface.

1

4

2

5

3

6

7

8

9

10

11

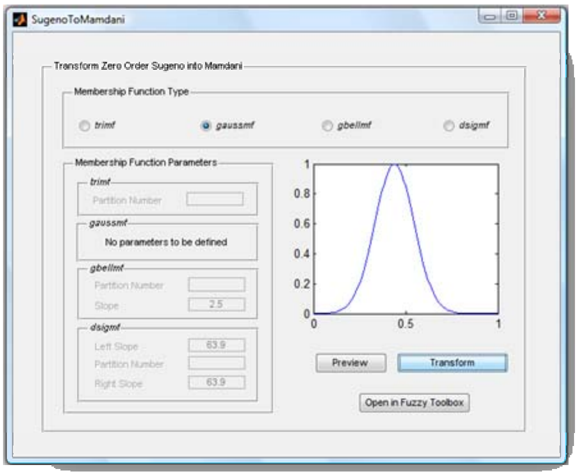
Figure 2: *SugenoToMamdani* interface.

Below, there is a brief description of both interfaces, accordingly with the numeration in Fig. 1 and 2.

1. *Data:* Reads data from an Excel or Text file.

2. *Model & Estimation:* Provides the possibility of choose the Takagi-Sugeno model type of the fuzzy system to be performed and the method of recursive estimation of the consequent parameters.

3. *Other*



Parameters Values: Setting of more parameters to perform the fuzzy system.

4. *Conditions for Substitution of Existent Rules:* Provides the possibility of choose the condition for substitution of existent rules during the fuzzy system creation process.

5. *Conditions for Creation of New Rules:* Provides the possibility of choose the condition for creation of new rules during the fuzzy system creation process.

6. *My Conditions:* Provides the possibility to define your own conditions for substitution of existent rules and creation of new rules during the fuzzy system creation process.

7. *Create Model/View Graphics:* Push the button *Create Model* to create the fuzzy system and activate the check box *View Graphics* if you want to view graphical information during the fuzzy system creation process.

8. *Create Results Table:* Create a table to show antecedents and consequents for each rule created in *Rules Antecedents & Consequents* panel.

9. *Open in Fuzzy Toolbox:* Open the fuzzy system created in Fuzzy Toolbox.

10. *Transform Sugeno into Mamdani:* Provides the possibility of transform the created Takagi-Sugeno fuzzy system into a Mamdani fuzzy system. Opens a new window where the necessary parameters could be set.

11. *Reset:* Restarts all components and variables.

12. *Membership Function Type:* Provides the possibility of choose the membership function type for the creation of the new Mamdani fuzzy system.

13. *Membership Function Parameters:* Provides the possibility of set other parameters to define the selected membership function for the creation of the new Mamdani fuzzy system.

14. *Preview:* Provides the possibility of preview the shape of the defined membership function.

15. *Transform:* Transforms the Sugeno fuzzy system into Mamdani system accordingly with the set parameters.

16. *Open in Fuzzy Toolbox:* Open the fuzzy system created in Fuzzy Toolbox.

2.3. How to work with eFSLab

2.3.1. Data Import

To create a Takagi-Sugeno fuzzy system, the user must define input and output data. So, the first step to follow in the application is to import the data, in the *Data* panel. For such, the user must choose a file by pressing the *Browse* button. Data could be read from a Text or an Excel file.

2.3.1.1. Text File Reading

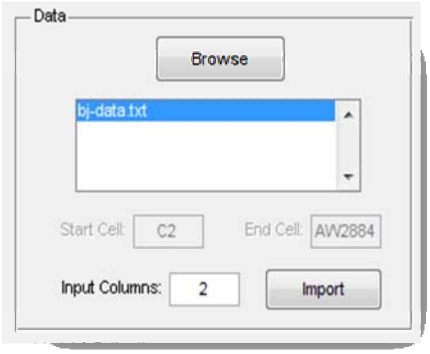


Figure 3: Reading a Text file in *eFSLab.*

In the case of a Text file, after select the file to use, the user has to set the number of input columns in the *Input Columns* field. If the number of inputs set were *n,* it is assumed that the inputs correspond to the first *n* columns of the data imported. It is also assumed that the output data corresponds to the last column of imported data.

In this way, the Text file must have a well defined structure, without headers and columns and rows names, and were the last column must contain the output data.

To import the data the user has to press the *Import* button.

2.3.1.2. Excel File Reading

Figure 4: Reading an Excel file in *eFSLab*.

In the case of an Excel file, after select the file to use, the user has to set the start and end cell in *Start Cell* and *End Cell* fields. The *Start Cell* is the first cell with data and the *End Cell* is the bottom right cell with data. The user must select only data, without headers and columns and rows names. As in the case of the text file, the *Input Columns* field must also be set, being assumed again that output data corresponds to the last data column.

In this way, the Excel file must have a well defined structure in which the last column must contain the output data.

To import the data the user has to press the *Import* button.

2.3.2. Takagi-Sugeno and Recursive Estimation Models

2.3.2.1. Takagi-Sugeno Model

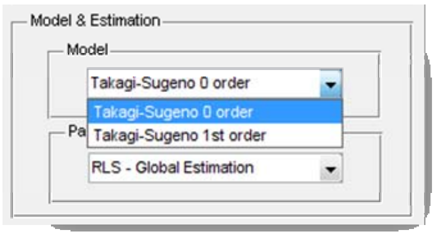


Figure 5: Setting Takagi-Sugeno model in *eFSLab*.

In *Model* drop-down list the user can choose the Takagi-Sugeno type to perform the fuzzy system.

The options are:

- *Takagi-Sugeno 0 order* → for each rule created there are only one consequent;
- *Takagi-Sugeno 1st order* → for each rule created there are four consequents;

2.3.2.2. Recursive Estimation Model

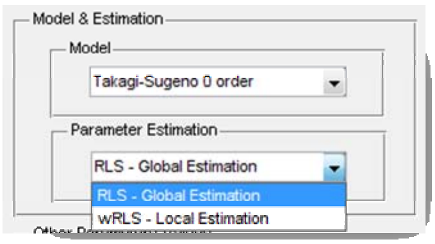


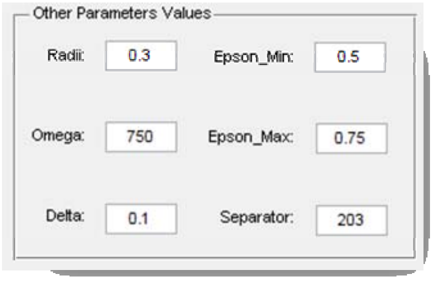
Figure 6: Setting Parameter Estimation model in *eFSLab*.

In *Parameter Estimation* drop-down list the user can choose consequent parameters recursive estimation model to perform the fuzzy system.

The options are:

- *RLS – Global Estimation* → the cost function is minimized, which guarantees globally optimal values of the parameters;
- *wRLS – Local Estimation* → the locally weighted cost function is minimized and locally meaningful parameters are obtained.

2.3.3. Other



Parameters Values

Figure 7: Setting Other Parameters Values in *eFSLab*.

This panel allows setting important parameters to the process of creation of the fuzzy system.

These parameters are:

- *Radii* → Is a positive constant, which defines the spread of the antecedent and the zone of influence of the i^{th} model. Affects the number of rules and consequently the performance and complexity of the models. In general, as the constant *Radii* increases the number of rules created decreases. Its default value is 0.3.

- *Omega* → Has influence on the estimation of the consequence parameters. A small value for *Omega* means that we have some confidence in the initialization parameters of the new rule consequents. A bigger value expresses less confidence in the initialization and inherently it is given a better adaptation capability to the method. Its default value is 750.

- *Delta* → Membership functions overlapping degree. Its value is usually between 0.05 and 0.2. In this case, the default value is 0.1.

- *Epson_Min* → This variable is used in subtractive clustering process, to set a boundary condition – lower threshold – which is defined by the expression *Epson_Min*$*P^{ref}$, where P^{ref} is the maximal potential called reference potential. Its default value is 0.5.

- *Epson_Max* → This variable is used in subtractive clustering process, to set a boundary condition – upper threshold – which is defined by the expression *Epson_Max*$*P^{ref}$, where P^{ref} is the maximal potential called reference potential. Its default value is 0.75.

- *Separator* → This variable allows to user defines test and training data groups inside the input data group. In this way, the *Separator* is the number of the line that the user selects to divide the input data group. Its default value is determined assuming that the training group has 70% of the data of the input data group. So this default value varies with the selected file.

2.3.4. Conditions for Substitution and Creation of Rules

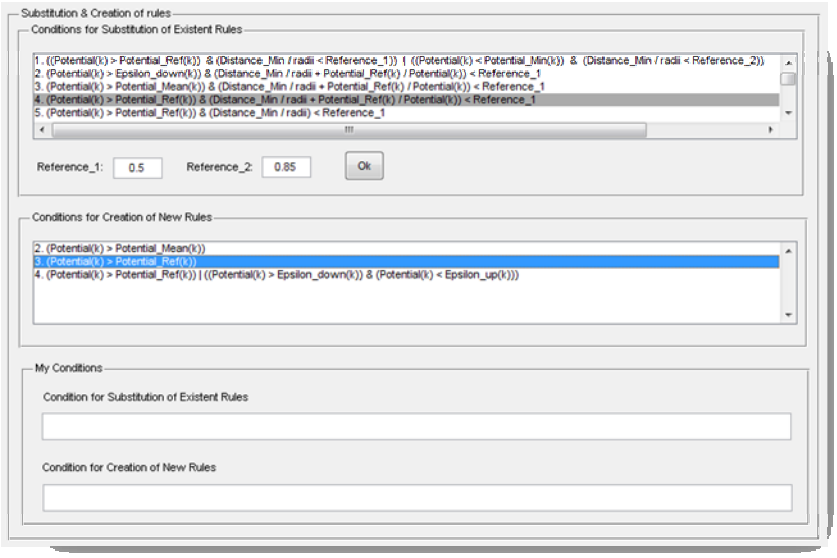
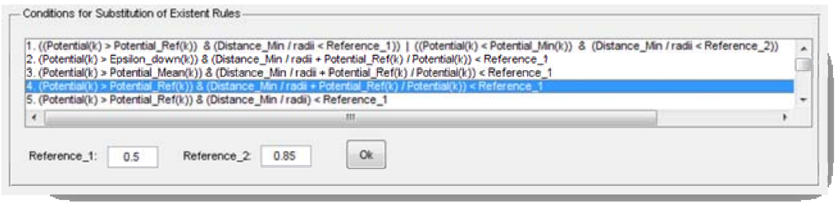


Figure 8: Setting Conditions for Substitution and creation of rules in *eFSLab*.

In this panel the user can chose one of the provided conditions for substitution of existent rules and its correspondent condition for creation of new rules or write his conditions.



2.3.4.1. Conditions for Substitution of Existent Rules

Figure 9: Setting Conditions for Substitution of existent rules in *eFSLab*.

Here the user can chose the condition that allows substituting existent rules by comparison between the potential of the new data sample and the potential of existing rule centers.

After choosing the condition, the user can also set Reference_1 and Reference_2 values.

To finish, the user must push *Ok* button in order to show in *Conditions for Creation of New Rules* list box only options compatible with the substitution condition selected. Compatibility between conditions is presented in the Table 3.

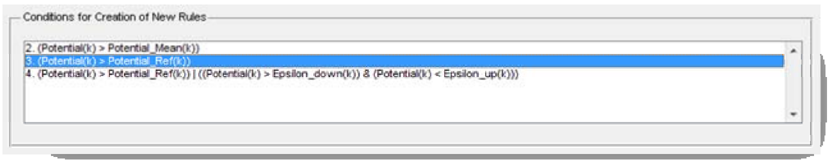
In order to easily understand the meaning of each condition for substitution of rules, these conditions are explained in Table 1.

Conditions	Meaning
1	IF the potential of the new data point is higher than the potential of the existing centres OR lower than the minimum potential of existing centres AND the new data point is close to an old centre (shortest distance is lower than Reference_1 OR Reference_2).
2	IF the potential of the new data point is higher than a threshold Epsilon_down AND the new data point is close to an old centre (sum of the shortest distance between them and its potential is lower than Reference_1).
3	IF the potential of the new data point is higher than the mean potential of the existing centres AND the new data point is close to an old centre (sum of the shortest distance between them and its potentials is lower than Reference_1).

4	IF the potential of the new data point is higher than the potential of the existing centres AND the new data point is close to an old centre (sum of the shortest distance between them and its potential is lower than Reference_1).
5	IF the potential of the new data point is higher than the potential of the existing centres AND the new data point is close to an old centre (shortest distance is lower than Reference_1).
6	IF the potential of the new data point is higher than the potential of the existing centres OR between two thresholds, Epsilon_down and Epsilon_up, AND the new data point is close to an old centre (shortest distance is lower than Reference_1).
7	IF the potential of the new data point is higher than the potential of the existing centres OR lower than the minimum potential of existing centres AND the new data point is close to an old centre (shortest distance is lower than Reference_1).
8	IF the potential of the new data point is higher than the potential of the existing centres AND the new data point is close to an old centre (sum of the shortest distance between them and its potential is lower than Reference_1).
9	IF the potential of the new data point is higher than the potential of the existing centres AND the new data point is close to an old centre (sum of the shortest distance between them and its mean potential is lower than Reference_1).
10	IF the potential of the new data point is higher than the mean potential of the existing centres OR between two thresholds, Epsilon_down and Epsilon_up, AND the new data point is close to an old centre (shortest

	distance is lower than Reference_1).
11	IF the potential of new data point is higher than the potential of the existing centres OR lower than the minimum potential of the existing centres AND the new data point is close to an old centre (sum of the shortest distance between them and its potentials is lower than Reference_1 OR Reference_2).

Table 1: Meaning of each condition for substitution of rules.



2.3.4.2. Conditions for Creation of New Rules

Figure 10: Setting Conditions for creation of new rules in *eFSLab*.

In this step, only options compatible with the selected condition for substitution of existent rules are available.

These conditions allow creating new rules by comparison between the potential of the new data sample and the potential of existing rule centers.

Compatibility between conditions for substitution and creation of rules is presented in the Table 3.

In order to easily understand the meaning of each condition for creation of rules, these conditions are explained in Table 2.

Conditions	Meaning
1	IF the potential of new data point is higher than the potential of existing centres OR lower than the minimum potential of existing centres.
2	IF the potential of new data point is higher than the mean potential of existing centres.
3	IF the potential of new data point is higher than the potential of existing centres.
4	IF the potential of new data point is hogher than the potential of existing centres OR the potential of new data point is between two thresholds, Epsilon_down and Epsilon_up.

Table 2: Meaning of each condition for creation of new rules.

2.3.4.3. Compatibility of Conditions

Compatibility between conditions for substitution and creation of rules is presented in the Table 3.

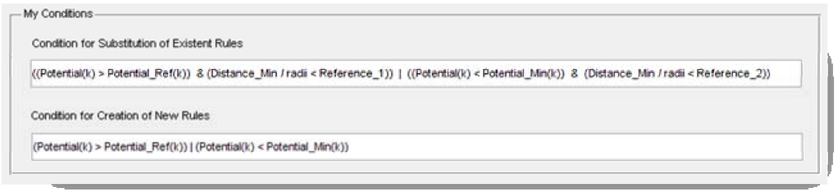
If conditions are not compatible a dimension error occurs and fuzzy system creation process stops.

This is a very important issue mainly if the user chooses to write his conditions in *My Conditions* panel.

		Conditions for Creation of Rules			
		1	2	3	4
Conditions for Substitution of Rules	1	X	X	X	X
	2		X	X	X
	3		X	X	X
	4		X	X	X
	5	X	X	X	X
	6	X	X	X	X
	7	X	X	X	X
	8		X	X	X
	9		X	X	X
	10		X	X	X

	11	X	X	X	X

Table 3: Compatibility between Conditions for substitution and creation of rules.



2.3.4.4. My Conditions

Figure 11: Writing conditions for substitution and creation of rules in *eFSLab*.

In these fields the user can write its conditions for substitution of existent rules and for creation of new rules.

However, to define these conditions the user can only use logical, relational and arithmetic operators like they are represented in Matlab syntax and some variables in a particular way.

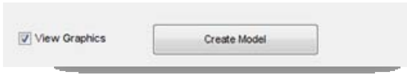
Allowed variables and their meaning are showed in Table 4.

Variable	Meaning
Potential(k)	Potential of the new data point (k)
Potential_Ref(k)	Maximum potential of the existing centers
Potential_Min(k)	Minimum potential of the existing centers
Potential_Mean(k)	Mean potential of the existing centers
radii	Defines the spread of the antecedent and the zone of influence of the i^{th} model
Distance_Min	Minimum distance between the new data point and old centers
Reference_1	General value that can be used as a reference to other variables
Reference_2	General value that can be used as a reference to other variables
Epsilon_down(k)	Potential_Ref(k) will be multiplied by this variable in order to define a minimum potential threshold
Epsilon_up(k)	Potential_Ref(k) will be multiplied by this variable in order to define a maximum potential threshold

Table 4: Variables allowed to write conditions of substitution and creation of rules and their meanings.

To define the conditions, variables must be used in *eFSLab* exactly as they are written in Table 4.

When the user is writing the conditions must pay attention to compatibilities between rules substitution and creation conditions. If both conditions weren't concordant a Matlab error will appear due to matrix dimensions and the process are interrupted.



2.3.5. Creating the Fuzzy Model

Figure 12: Creating Model pushbutton and checkbox for graphics visualization.

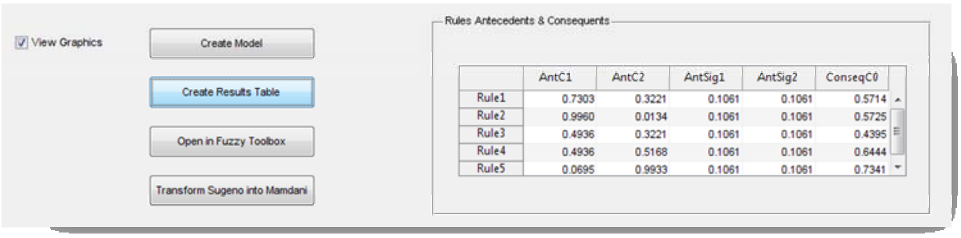
After setting all needed parameters, the user must press *Create Model* button in order to initialize the fuzzy system creation process.

If user wants graphically follow the whole process development, must check *View Graphics* option before press *Create Model* button.

2.3.6. Consult Results

Results of fuzzy system creation process can be accessed by two ways:

- For each system creation is provided a text file named *Diagnosis* which contains information about whole process, values of consequents and antecedents and some performance evaluation results.
- Press *Create Results Table* button and a table with antecedents and consequents values for each rule appear in *Rules Antecedents & Consequents*



panel, like is showed in Figure 13.

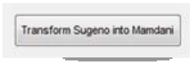
Figure 13: Create a table with rules antecedents and consequents.

2.3.7. Analyzing properties of created fuzzy system

Figure 14: Button to open created fuzzy system in Matlab Fuzzy Toolbox.

Pressing *Open in Fuzzy Toolbox* button, fuzzy system created will be opened in Matlab Fuzzy Toolbox. There, user can analyze fuzzy system properties.

2.3.8. Go to



TransformToMamdani interface

Figure 15: Button to go to *TransformToMamdani* interface.

Pressing *Transform Sugeno into Mamdani* button will be opened another window where the user can set transformation parameters.

If the user intended to transform a first order Takagi-Sugeno system into a Mamdani, it will be advised that Mamdani precision would be worst than Takagi-Sugeno because transformation is made centering the consequent in the corresponding Takagi-Sugeno independent consequent.

2.3.9. Transform Takagi-Sugeno into Mamdani fuzzy model

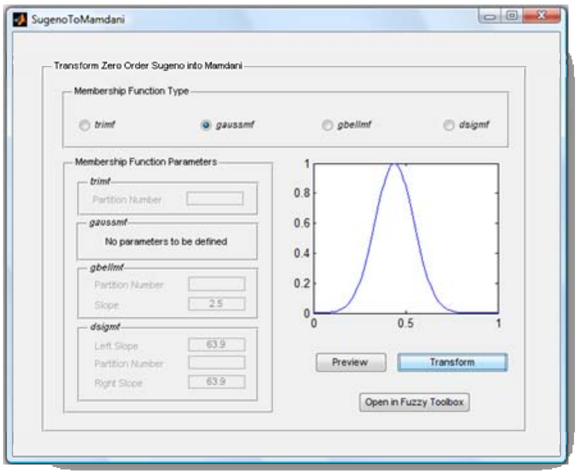
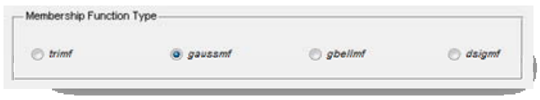


Figure 16: *TransformToMamdani* interface.

Here the user can transforms a Takagi-Sugeno fuzzy system into a Mamdani fuzzy system. To do that, some parameters must be set.



2.3.9.1. Membership function type

Figure 17: Set membership function type.

User must set the membership function type, checking the wanted type.

Options are:

- *trimf* → Triangular membership function;
- *gaussmf* → Gaussian curve membership function;
- *gbellmf* → Generalized bell curve membership function;
- *dsigmf* → Membership function composed of the difference between two sigmoidal membership functions.

For more detailed information about membership function types consult Matlab help.

2.3.9.2. Membership function parameters

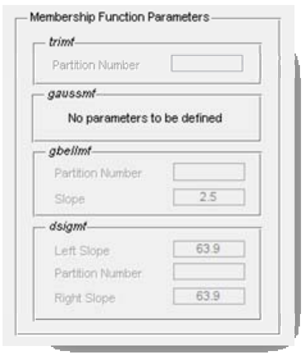


Figure 18: Set membership function parameters.

Parameters that must be set are:

- Partition Number → Number of partitions for fuzzy system;
- Slope →Slope of *gbellmf* membership function;
- Left Slope →Left slope of *dsigm* membership function;
- Right Slope → Right slope of *dsigm* membership function;

For *gaussmf* membership function it is not necessary to set any parameters because function shape and its centers are defined by sigma and consequents of Sugeno system, respectively.

2.3.9.3. Preview membership function shape

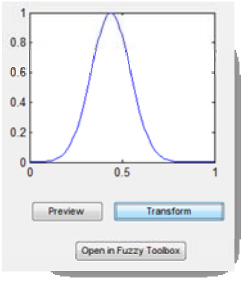
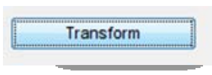


Figure 19: Preview of membership function shape defined.

To pre-visualize the shape of the membership function defines, the user can press *Preview* button and graphic representation appears.



2.3.9.4. Transforming the fuzzy model

Figure 20: Button to initialize Takagi-Sugeno transformation into Mamdani.

To initialize

transformation process user has to press *Transform* button.

2.3.9.5. Analyzing properties of transformed fuzzy system

Figure 21: Button to open transformed fuzzy system in Matlab Fuzzy Toolbox.

Pressing

Open in Fuzzy Toolbox button, fuzzy system transformed will be opened in Matlab Fuzzy Toolbox. There, user can analyze fuzzy system properties.

2.3.10. Reset eFSLab

Figure 22: Reset button.

Pressing *Reset* button, all components and variables are reset and all windows are closed.

3. Conclusions

The sFSLab allows an easy building of evolving fuzzy systems from data. It is of course an on-going work available to all scientific community. It can be downloaded from http://eden.dei.uc.pt/~dourado/eFSLab and is free under GNU license. New features will be introduced in the future and the authors hope that other researchers will contribute to it in such a way that it will become an important tool for all researchers and practitioners working with fuzzy modeling in the data paradigm.

One important progress will be the development of a technique to transform first order TSK models to Mamdani type, finding consequents that will allow similar accuracy of TSK models.

Accomplished objectives

Future work

4. References

[1] Angelov, P. "Evolving Rule-Based Models: A Tool for Design of Flexible Adaptive Systems", Studies in Fuzziness and Soft Computing, vol. 92, SpringerVerlag, 2002

[2] Kasabov, N. and Q. Song, "DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction", IEEE Transactions on Fuzzy Systems, vol. 10, no. 2, pp. 144-154, 2002.

[3] Angelov, P. and D. Filev,"An Approach to Online Identification of Takagi-Sugeno Fuzzy Models", IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 34, no. 1, pp. 484-498, 2004.

[4] Mencar C., G. Castellano and A.M. Fanelli, "Distinguishability quantification of Fuzzy sets", Information Science, 177:130-149, Elsevier, 2007.

[5 Mikut, R., J. Jäkel, L. Gröll "Interpretability issues in data-based learning of fuzzy systems", Fuzzy Sets and Systems , vol. 150, 179–197, 2005

[6] Mencar, C., G. Castellano, A. M. Fanelli, "Some Fundamental Interpretability Issues in Fuzzy Modeling", Proceedings of EUSFLAT, Barcelona, September 7-9, 2005

[7] Setnes, M., "Complexity Reduction in Fuzzy Systems", PhD Thesis, Delft University of Technology, 2001,

[8] Zhou, S.-M. and J. Q. Gan, "Constructing accurate and parsimonious fuzzy models with distinguishable fuzzy sets based on an entropy measure", Fuzzy Sets and Systems vol. 157, 1057 – 1074, 2006.

[9] Broekhoven, E. V., V. Adriaenssens, B. De Baets, "Interpretability-preserving genetic optimization of linguistic terms in fuzzy models for fuzzy ordered classification: An ecological case study", International Journal of Approximate Reasoning Vol. 44 , Is.1, pp. 65-90, 2007.

[10] Alcalá Rafael, J. Alcalá, Francisco Herrera, José Olero Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation, International journal of approximate reasoning (Int. j. approx. reason.) **ISSN** 0888-613X, vol. 44, n° 1 (33 ref.), pp. 45-64 , 2007, Elsevier

[11] Ester Van Broekhoven , Veronique Adriaenssens , Bernard De Baets, Interpretability-preserving genetic optimization of linguistic terms in fuzzy models for fuzzy ordered classification: An ecological case study,2007

[12] Sivapragasam, C., Rule reduction in fuzzy logic for better interpretability in reservoir operation

Hydrological Processes , vol. 21 issue 21, pp 2835-2844 ,J.Wiley and Sons, 2007.

[13] González, J., I.Rojas, H. Pomares, L. J. Herrera, A. Guillén, J. M. Palomares, F. Rojas, "Improving the accuracy while preserving the interpretability of fuzzy function approximators by means of multi-objective evolutionary algorithms", International Journal of Approximate Reasoning ,Volume 44 , Is.1, , pp32-44, 2007.

[14] Gómez-, A.F., F. Jiménez, G. Sánchez, Improving interpretability in approximative fuzzy models via multiobjective evolutionary algorithms, International Journal of Intelligent Systems, Volume 22, Issue 9 , Pages 943 – 969, Wiley Periodicals, 2007.

[15] Ishibuch, H. and Y. Nojima, "Analysis of interpretability-accuracy trade off of fuzzy systems by multiobjective fuzzy genetics-based machine learning", International Journal of Approximate Reasoning, Volume 44 , Is 1, pp 4-31, 2007.

[16] Jin Y. and B. Sendhoff, "Extracting Interpretable Fuzzy Rules from RBF Networks", Neural Processing Letters, Vol.17,nº 2, pp. 149-164(16), Springer, 2003.

[17] Victor , J and Dourado, A., "On line interpretability by rule base simplification and reduction", Proc. Eunite Symposium, Aachen, 2004.

[18] Box, G.; Jenkins, G., 1976, "Time Series Analysis: Forecasting and Control", second ed., Holden-Day, San Francisco, CA.