Rule reduction in a fuzzy model of a chain system

Izabela Rejer

University of Szczecin, Faculty of Economics and Management
Mickiewicza 64/66, 71-101 Szczecin, Poland
i_rejer@uoo.univ.szczecin.pl

The aim of this article is to present a method which can be used for eliminating unnecessary rules from a rule base of a fuzzy model built for a system of a chain data distribution. The article presents both – a basic version of the method, possible to use in 2-input systems and its extended version, possible to use in multi-input systems. The application of the proposed method will be illustrated via a real economic system of an unemployment rate in Poland in years 1992-1999.

Introduction

A large number of models built via soft-computing techniques are regarded as black-box models, it is as models which internal processes are hardly understandable for a human. Very often the only criterion which is taken into consideration in the process of building such models is an accuracy criterion. In situations when a model is built for automatic performing repeatable actions or calculations, such approach is often justifiable. However, when a model is built in order to support a human in the decision process or in order to explain the behavior of a real system, the accuracy criterion is not sufficient. A model which is constructed in order to support a human in his work should be not only precise but also highly intelligible. Since the interpretation difficulties increase with the model complexity, the model created to support a human should be as simple as possible.

Fuzzy and neuro-fuzzy models are very useful when a real system is to be described in the form of rules understandable for a human. Unfortunately both types of models in their classic form are very prone to the problem known as *curse of dimensionality*, which causes that even when a very small number of input variables described by a very small number of membership functions are used in the model, the resulting rule base can be too complex to be understandable for a human [6].

The *curse of dimensionality* problem is a very serious one but it does not discriminate the fuzzy and neuro-fuzzy models as useful tools for a decision process. The fact is that very often the rule base of a fuzzy model contains a lot of unnecessary rules which can be eliminated without the loss of the model precision. Obviously the problem is how to find out which rules of a fuzzy model are unnecessary ones. Theoretically the solution is simple – it should be enough to reveal which rules are not covered by any data point. In practice, however, such straightforward approach can have a very serious negative consequence – it can result in a non-continuous fuzzy model. Since a well design fuzzy model should be a continuous one, another approaches to eliminate unnecessary rule should be applied.

The aim of this article is to introduce a method which can be used for identifying unnecessary rules of a fuzzy model but which does not leave the empty regions inside the model. The method in its classic form is dedicated for a specific class of systems - chain systems.

A chain system is a system which is characterised by a chain data distribution in a whole domain. Due to this feature the chain system can be described not only by a surface model (e.g. fuzzy model) but also by a parametric curve which shows the very centre of the data distribution and which can be the base for determining the most dense region of data points. The knowledge of the region containing the vast majority of data points can be used to eliminate rules which are placed outside this region. With this approach it is impossible to obtain a non-continuous model because the chain model is always a continuous one.

The application of the method proposed in this article will be illustrated via a real economic system of an unemployment rate in Poland in years 1992-1999. The output variable of this system is *unemployment rate* and the input variables are: *money supply*, *number of inhabitants* and *dollar's rate of exchange*. The data for the survey was provided by the Polish Statistic Department.

The article consists of three main sections. First section addresses two issues – building a parametric curve model and establishing the width of the region surrounding this model containing vast majority of training data. Second section explains the basis of the proposed method and presents its application in a 2-input system and the last one – extends the method for multi-input tasks.

Parametric curve model as a tool for establishing the data region

As it was stated in introduction the proposed method of rule reduction is designed for chain systems. Hence, the first issue which should be addressed here is how to find out whether a system has a chain or a surface nature.

The main feature of a multi-dimensional chain system is that its decomposition into one-dimensional subspaces (showing the behaviour of each system variable in regard to a parameter t - indicating the approximated data sequence) gives a set of tight chain dependences. Therefore, in order to verify whether a system is of a chain profile, the reverse analysis should be performed. Tight chain dependencies, visible on all two-dimensional graphs presenting the behaviour of system variables in regard to t parameter, will indicate the chain profile of the whole system. Sometimes there can be a problem with establishing the approximated data sequence. This problem, however, does not exist when time series systems are under consideration. In this class of systems the parameter t can be interpreted as time variable, explicitly or implicitly given in the data set.

Figure 1 presents data distributions of four variables of the unemployment rate system, described in introduction, in regard to parameter t (which in this case can be interpreted as time). As it can be observed in the figure, all four variables are of a strict chain distribution which indicates that the whole system has also a chain profile.

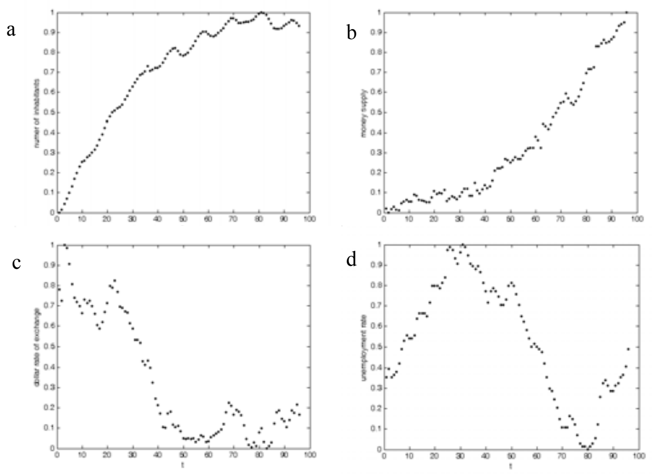

Fig. 1. Time series of the variables of the system of unemployment rate in Poland in years 1992-1999: a) money supply, b) number of inhabitants, c) dollar's rate of exchange, d) unemployment rate

The main benefit of the chain system is that it can be described not only by a surface model but also by a parametric curve [7]. The main idea of the parametric curve modelling method is to build a set of two-dimensional models, where each model describes the behaviour of one variable (input or output) in regard to the known parameter *t*. These two-dimensional models can be created with many different mathematic techniques e.g. non-linear neural networks, polynomial regression, splines, etc. Two-dimensional models, built with one of the mentioned techniques, are then assembled together in order to create a multi-dimensional model describing the input-output mapping in the whole space (eq. 1) [2].

$$\begin{cases} x_1 = f_1(t) \\ x_2 = f_2(t) \\ \dots\dots\dots\dots\dots \\ x_n = f_n(t) \end{cases} . \qquad (1)$$

In order to illustrate the process of parametric curve modelling, the 3D chain model of the unemployment rate was created (input variables: *money supply* and *number of inhabitants*). The neural networks of following parameters were used to build two-dimensional time series models of all system variables [3][5]:

- flow of signals: one-way,
- architecture of connections between layers: all to all,
- hidden layers: 1 hidden layer with suitable number of sigmoid neurons (5 for variable *number of inhabitants*, 3 for variable *money supply*, 4 *for variable unemployment rate*),
- output layer: 1 linear neuron,

- training method: backpropagation algorithm with momentum and changing learning rates,
- training aim: minimize mean absolute error (MAE),
- training time: 20000 epoch,
- testing method: visual control.

Models built with neural networks described above are shown in Fig. 2 (a, b, c). By assembling together all three models, the parametric curve model was created (fig. 2d). The approximated accuracy of this model was calculated using MAE (eq. 2) [1] and was equal to 2.28% [8].

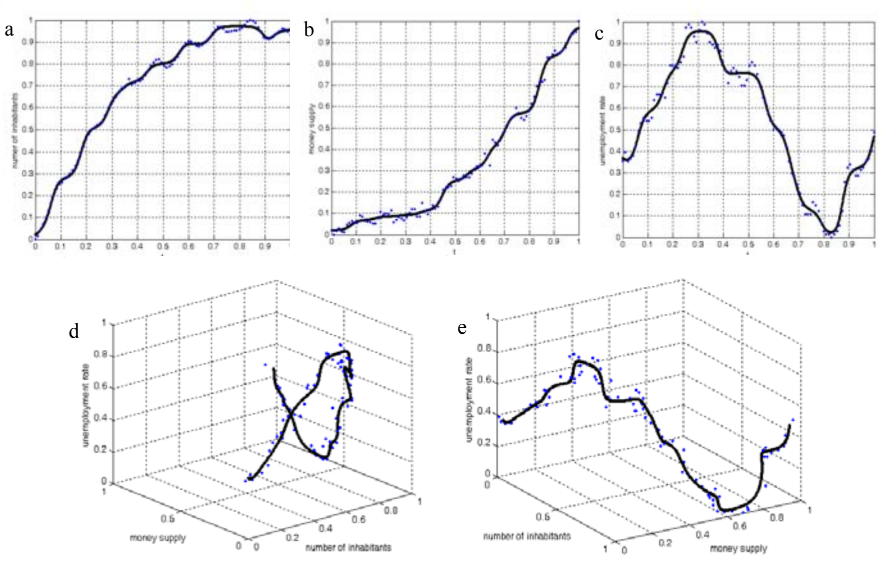

Fig. 2. Unemployment rate system: a, b, c) Two-dimensional time series models; d, e) Three-dimensional parametric curve model

$$MAE = \frac{\sum_{i=1}^{n} z_i^* - z_i}{n}, \qquad (2)$$

where: z_i^* – empirical values, z_i – theoretical values, n – number of data points.

Two facts concerning systems of a chain data distribution are very helpful in a process of establishing the region covering most data points of the analysed system:

1. The parametric curve model shows the very centre of the data distribution in a multi-dimensional space.
2. Most data points of the analysed system lay in a very close distance from the chain model.

Obviously, the knowledge of the centre of data points is not sufficient to establish the region covering most data points of the analysed system. The second point which has to be addressed is a width of this region in a multi-dimensional space. Assuming that the width of this region is the same in each direction, it can be calculated as a radius of a hypertube surrounding the chain model in a multi-dimensional space. In order to establish the radius of this hypertube, the distribution of the absolute distances between each training data point and the chain model, calculated in a multi-dimensional space, are considered. The radius is calculated on the base of three quartiles of this distribution (first, second and third) (eq. 3) [8].

$$R_h = Q3 + \min(Q1; Q2 - Q1; Q3 - Q2), \qquad (\mathbf{3})$$

where: Q1 - first quartile (which indicates the region surrounding the chain model covered by 25% of data points), Q2 - second quartile (which indicates the region surrounding the chain model covered by 50% of data points), Q3 – third quartile (which indicates the region surrounding the chain model covered by 75% of data points), R_h - radius of the hypertube.

Figure 3 presents a hypertube surrounding the chain model of unemployment rate built earlier in this section. The radius of the hypertube, calculated on the basis of eq. 3, was equal to 0.0387. Figure 3a presents the hypertube in the whole system space and fig. 3b – presents its projection on the input space. Obviously for rule reduction algorithm the shape of the hypertube in the input space will be essential.

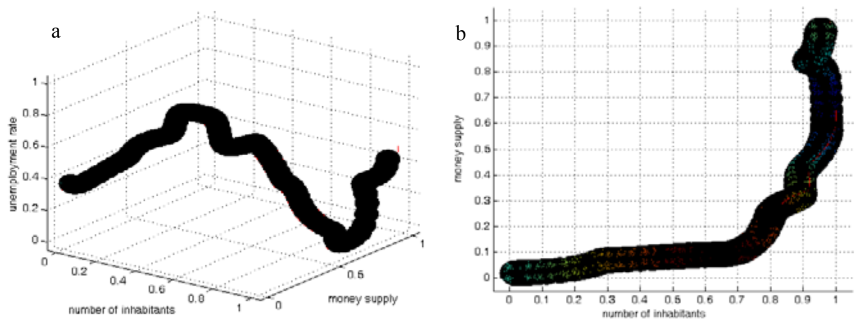

Fig. 3. Hypertube surrounding the chain model of an unemployment rate: a) in the whole system space, b) in the input space

Algorithm for rule reduction in a two-input system

The proposed method of rule reduction will be discussed on the example of a neuro-fuzzy model of unemployment rate of two input variables: *number of inhabitants* and *money supply*. The parameters of the model are as follows:

– input membership functions - asymmetrical triangular functions, 5 functions per each input variable,
– output membership functions – 25 singletons,
– inference method – MIN-MAX,
– deffuzification method – height method,
– training method – backpropagation algorithm with momentum,

– training time: 1000 epoch,

Figure 4 illustrates the surface of the neuro-fuzzy model of above parameters (fig. 4a) and its rule net in an input space (fig. 4b). The MAE of the model is equal to 3.90%.

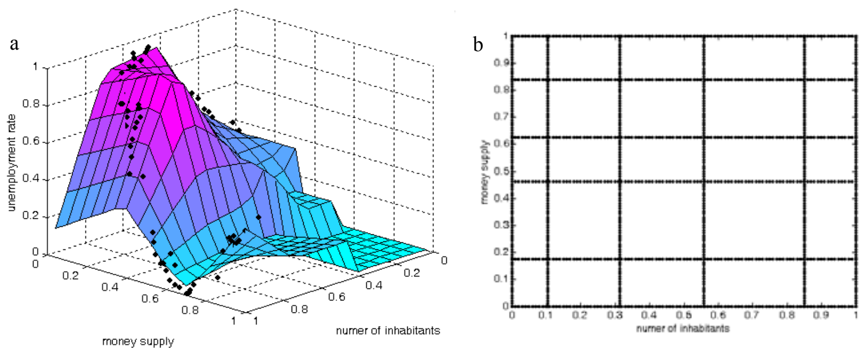

Fig. 4. Neuro-fuzzy model of unemployment rate: a) model surface, b) rule net

In case of systems described by two input variables the proposed algorithm for rule reduction is quite straightforward and can be described in the following two steps:

– Step 1 - Identify rules which overlap the hypertube in the input space.
– Step 2 - Identify rules which have to be added to the rule base (created in the first step) in order to fulfil the requirements of the inference process.

Step 1

The aim of this step is to reveal rules which cover all data points lying inside the hypertube. It is of no importance whether a rule cover a large number of data points or only one data point – it is essential to find out all rules overlapping the hypertube. Obviously in order to fulfil this task the boundaries of the hypertube have to be found. As it was stated in previous section the hypertube surrounds the chain model in equal distance in each point of each dimension. Hence, there is no straight mathematic transformation of the chain model equation which could be used to establish the hypetube boundaries. Therefore, in order to find out these boundaries the chain model has to be equally sampled in a large number of points (for the model described in this section in 1000 points) and per each point the positions of two points orthogonal to the chain model, lying in a distance R_h have to be calculated (fig. 5a). As a result two sets of points pointing the two hypertube boundaries are obtained (fig. 5b).

After determining the hypertube boundaries, all rules which do not overlap the hypertube are temporary eliminated from the rule base. Figure 6a shows the rules which were left in the discussed neuro-fuzzy model after performing the first step of the proposed algorithm. As it can be noticed more than 50% of rules were temporary eliminated.

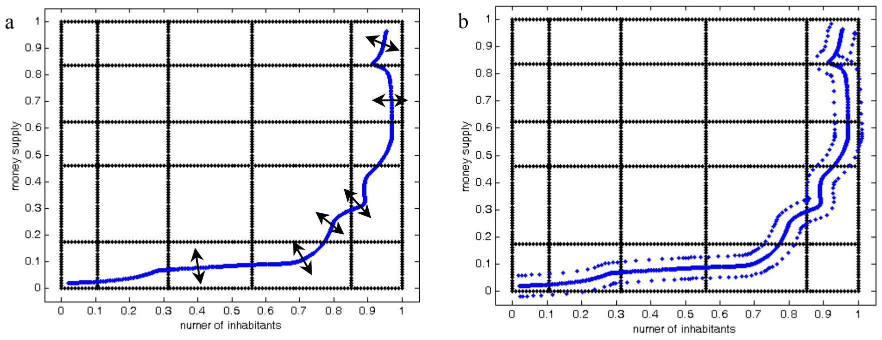

Fig. 5. Process of determining the boundaries of the hypertube surrounding the chain model

Step 2

Obviously the algorithm cannot be stopped at this point because the fuzzy model obtained after the first step is an incomplete one (its error – 5.34% - is almost 50% bigger than the error of the original model – 3.90%). This incompleteness is a result of a way of constructing the rule net of a fuzzy model. The rule net intersections are located in points where the membership functions are equal to 0.5. That means that for some data points, lying in the region of influence of the outer rules, the maximum membership degree to all fuzzy sets will not be equal to 1 but will be smaller than 1. Of course such situations disable the proper inference process and should be avoided.

In order to eliminate mentioned problem, the rule base created in the first step has to be extended. The rules which should be added to the model to ensure its completeness are the neighbours of these outer rules which fulfil the following condition: the peak of the membership function contained in the rule premise lies inside the hypertube. The process of adding rules is performed separately for each dimension and for each rule. For most rules only one neighbour has to be added. Only rules which overlap both hypertube boundaries (and fulfil presented condition) require sometimes adding of two neighbours.

Figure 6b presents the rules which were added to the rule base created in the first step of the algorithm in order to ensure the completeness of the fuzzy model. As it can be noticed 6 rules were added to the rule base (one of the outer rules did not need adding a neighbour because the condition concerning the membership function peak was not fulfilled).

The MAE of the fuzzy model containing the rule base from fig. 6b was equal to 3.91%. This error value (almost equal to the value of the original model – 3.90%) proves that all important rules were left in the rule base and that the whole model can be regarded as a complete one. The very slight difference in both models errors (0.01%) was caused by leaving some data points (possible outliers) outside the hypertube boundaries.

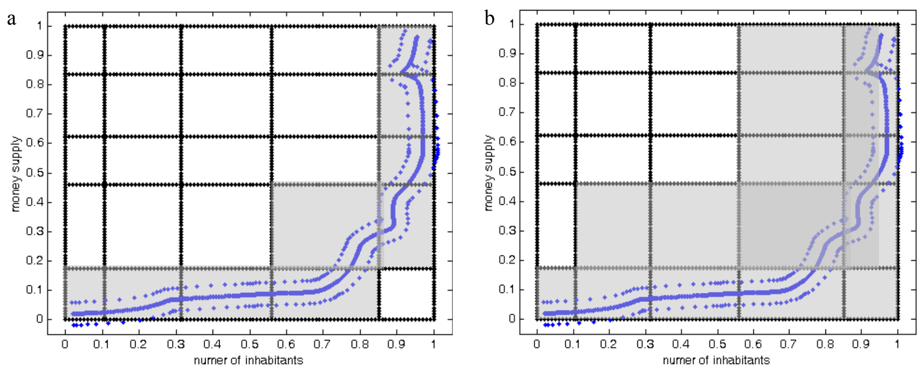

Fig. 6. a) Rules left in the rule base after completing the first step of the algorithm, b) Rules added to the rule base in the second step of the algorithm

The application of the proposed method for rule reduction allowed to eliminate 10 of 25 rules contained in the rule base of the unemployment rate model. That means that 40% of rules were identified as unnecessary ones and eliminated from the model. This result is a significant one but far more spectaculars results can be achieved when systems of more input variables are regarded.

Algorithm for rule reduction in a multi-input system

The previous section presented the way of applying the proposed method of rule reduction in a 3D system. Obviously, the question is how to transform the algorithm to the form which could be applied in multi-input systems. Since there is no need to perform the algorithm for the whole input space at once, the transition is quite straightforward. The only thing which has to be done is to divide the input space of the analysed system into 2-input subspaces and perform both algorithm steps separately for all subspaces. Why it is enough to analyse 2D subspaces of the whole input space? The answer is quite simple. A rule overlaps a hypertube in the whole input space only when it overlaps it in all 2D subspaces. When a rule overlaps a hypertube only in some 2D subspaces that means the hypertube does not intersect the rule in the whole space (for example the hypertube lies over or under the rule hypercube).

To be more specific the algorithm of rule reduction in case of a multi-input system can be described as follows:

1. Find all 2D subspaces of the whole input space of the analysed system
2. For each 2D subspace - identify rules which overlap the hypertube
3. Create a rule base composed of rules which overlap the hypertube in all 2D subspaces
4. For each 2D subspace – identify rules which should be added to the rule base in order to fulfil the requirements of the inference process
5. Extend the rule base created in the third step by adding rules which were chosen in the fourth step in all 2D subspaces

The algorithm performance will be presented via a neuro-fuzzy model of unemployment rate of three input variables: *number of inhabitants*, *money supply* and *dollar's rate of exchange*. The parameters of the model were almost the same as described in previous section. The only difference was in the number of input membership functions which was equal to 6 (per each input). The model error, after completing the learning process, was equal to 2.63%. Before starting the algorithm a chain model of the analysed system was constructed (MAE = 3.80%) and the hypertube radius for the input space was established ($R_h = 0.087$).

The algorithm results were as follows:

- after completing the first three steps of the algorithm the rule base of the original fuzzy model was reduced from 216 to 37 rules (MAE = 4.42%),
- after completing next two steps the rule base was extended to 79 rules and the MAE was reduced to 2.68%.

As it can be noticed the application of the proposed method for rule reduction in a three-input system allowed to eliminate 137 of 216 rules contained in the rule base of the fuzzy model. That means that 63% of rules were identified as unnecessary ones and eliminated from the model.

Conclusion

The aim of this article was to present a method which can be used for rule reduction of a fuzzy model. The main advantages of the proposed method are as follows:

- The reduction rate is a significant one (in presented applications it was equal to 40% and 63%) which is very important not only when a model is used in its software version as a tool supporting a human but also when a model is a base for hardware implementation.
- The resulting models have almost the same rate of precision as the non-reduced ones. That means the algorithm eliminates rules which are really unnecessary and do not take part in the inference process.
- The models obtained after applying the proposed method are continuous ones.

Although, the practical application of the proposed method was illustrated via a time series system, its application is absolutely not limited to this class of systems. This is due to the fact that chain models, used in the method, can be constructed not only for time series systems but also for any kind of chain systems in which the data sequence can be established. Moreover, to apply the proposed method in practice, there is no need to construct the chain model of the analysed system – it is enough to establish the possible interpolation region of the fuzzy model with any available technique [4]. That means the application of the method is not limited to chain systems but it can be also used in surface systems. Obviously, since the surface systems in general need far more rules than chain ones, the reduction rate in surface systems is much smaller.

References

[1] Aczel, A. D.: Complete Business Statistics, Richard D. Irwin Inc., Sydney (1993)
[2] Bronsztejn I.N., Siemiendiajew K.A., Musiol G., Mühlig H.: Modern Compendium of Mathematic, Polish Science Publisher Warsaw (2004)
[3] Demuth, H., Beale, M.: Neural Network Toolbox User's Guide, The Math Works Inc., Natick MA USA (2000)
[4] Klesk P.: The method of setting suitable extrapolation capabilities for neuro–fuzzy models of multi–dimensional systems", PhD Thesis, Technical University of Szczecin, 2005.
[5] Masters, T.: Practical Neural Networks Recipes in C++, Academic Press Inc (1993)
[6] Piegat, A.: Fuzzy Modeling and Control, Physica-Verlag, New York (1999)
[7] Piegat A., Rejer I. Mikołajczyk M.: Application of Neural Networks in Chain Curve Modelling, Artificial Intelligence and Soft Computing ICAISC 2006 - Lectures Notes in Artificial Intelligence, Springer-Verlag, Berlin (2006) (in publishing)
[8] Rejer I, Mikołajczyk M.: A Hypertube as a Possible Interpolation Region of a Neural Model, Artificial Intelligence and Soft Computing ICAISC 2006 - Lectures Notes in Artificial Intelligence, Springer-Verlag, Berlin (2006) (in publishing)