Automatic Identification of Fuzzy Models with Modified Gustafson-Kessel Clustering and Least Squares Optimization Methods

Grzegorz Glowaty

AGH University of Science and Technology, Department of Computer Science,
Al. Mickiewicza 30, 30-059 Krakow, Poland
glowaty@agh.edu.pl

Abstract. An automated method to generate fuzzy rules and membership functions from a set of sample data is presented. Our method is based on clustering and uses a modified version of Gustafson-Kessel algorithm. The aim is to divide a product space into set of clusters for which the systems exhibits behavior close to linear. For each of the clusters we produce a fuzzy rule and generate a set of membership functions for the rule antecedent with use of an approach based on curve fitting. Weighted linear least-squares regression is used to obtain consequent functions for TSK-models.

Key words: fuzzy modeling, fuzzy clustering, Gustafson-Kessel algorithm.

1 Introduction

Fuzzy models have proven to be effective function approximators. They are also easy interpretable because they are composed of human readable rules. Those rules can be used to understand a nature of a modeled system. This huge advantage of fuzzy modeling above many other modeling techniques motivates researchers to work on automatic methods of fuzzy modeling as they eventually allow for easy generation of human readable interpretation of the system.

In this paper we focus on a fuzzy model generation with use of fuzzy data clustering. First, we provide a general idea of application of clustering in fuzzy model identification. We propose modifications to Gustafson-Kessel fuzzy clustering algorithm with a purpose of producing clusters more suitable for usage in fuzzy model. Then we show how to convert those clusters to TSK fuzzy models. At the end of this work models produced with the described method are compared with models produced by other classical fuzzy modeling approaches.

2 Clustering in Fuzzy Model Identification

Fuzzy rules introduce a natural partition of the system space. Antecedents of the rules introduce a partition of the input space. This partition defines a set of regions in which particular rules apply. General idea behind the use of clustering techniques in the

M. Bubak et al. (Eds.): ICCS 2008, Part I, LNCS 5101, pp. 883–892, 2008.
© Springer-Verlag Berlin Heidelberg 2008

fuzzy model identification [6, 8, 10] is that if we are able to find groups of sample data that exhibit similar behavior in a given area of a system space then we should be able to divide the problem of modeling into several smaller subspaces. In each of these subspaces we create a fuzzy rule that mimics approximated system's behavior in this area. Fuzzy clustering methods not only find cluster centers, but also assign membership degree of each of the samples to each of the clusters. We use this information in generation of fuzzy rules. We modify Gustafson-Kessel fuzzy clustering algorithm [9] and use it as the basis for our approach.

3 Finding Clusters

3.1 Desired Cluster Properties

The objective of our method is to create fuzzy rules for which antecedents are decomposed into set of predicates for each of the variables of the input domain. This kind of model provides the best interpretability of produced rules. There are approaches [6] that use *n-1* dimensional fuzzy sets as membership functions in rule antecedents (where *n* is the number of dimensions of the product space). However, those models are harder to interpret. In the best performing of the methods presented in [8], Gath and Geva clustering algorithm is used and a transformation of input variables is applied. The goal of the transformation is to leverage clusters as if they were parallel to the axes of the space. That also reduces readability of the rules.

In order to derive fuzzy model from a set of fuzzy clusters in the product (input-output) space $X_1 \times \ldots \times X_{n-1} \times X_n$ (where X_n is the output domain) a projection of each of the clusters onto each of the input space axes is obtained. Fuzzy clusters being results of the most of the fuzzy clustering algorithms are of the shape of sphere or hyper-ellipsoid. In case of spheres it is easy to obtain a "projection" of a cluster on an axis without a loose of information, however in case of hyper-ellipsoids the more axes of the ellipsoid are parallel to the axes of the space, the more information is preserved. Some of the approaches are based on this observation and look for the clusters that have all of their axes parallel to the axes of the space [10].

For the TSK fuzzy models the consequent of the fuzzy rule may be a linear function of input variables. In this case there is no need of projecting the cluster onto the output axis. With this in mind we propose a modified version of Gustafson-Kessel algorithm that finds clusters that are easily projected onto the input space, and not necessarily parallel to the output axis.

3.2 Gustafson-Kessel Algorithm

Let us assume a set of *N* samples in the *n* dimensional space. The target is to find *K* fuzzy clusters, such that:

$$\forall i \in \{1,\ldots,N\} \sum_{k=1}^{K} \mu_{k,i} = 1, \tag{1}$$

where $\mu_{k,i}$ is a membership degree of sample *i* to cluster *k*. Gustafson-Kessel algorithm finds clusters by minimizing the following function:

$$J_{X,m}(U,V)=\sum_{i=1}^{N}\sum_{k=1}^{K}\mu_{k,i}^{m}D_{A_k}^{2}(x_i-v_k), \tag{2}$$

where U is a set of membership degrees μ, V is a set of cluster centers v, m is fuzziness factor (usually a value close to 2), X is a set of N samples x, and $D_{A_k}^2$ is a norm induced by matrix A_k. Every cluster has its own norm inducing matrix

$$A_k=[\sigma_k \det(F_k)]^{\frac{1}{n-1}}F_k^{-1}, \tag{3}$$

where F is a fuzzy covariance matrix defined as follows:

$$F_k=\frac{\sum_{i=1}^{N}\mu_{k,i}^{m}(x_i-v_k)(x_i-v_k)^T}{\sum_{i=1}^{N}\mu_{k,i}^{m}}. \tag{4}$$

Parameter σ_k in (3) was introduced as a cluster capacity so the objective function minimization is not trivial process of minimizing all values of matrix A. Usually for Gustafson-Kessel algorithm destination capacity of 1 for each of the clusters is assumed. Norm $D_{A_k}^2$ induced by matrix A_k is calculated in the following way:

$$D_{A_k}^{2}(\mathrm{x})=(\mathrm{v_k}-x)^T A_k(v_k-x). \tag{5}$$

Given the membership degrees centers of the clusters are calculated as the weighted mean value of all membership degrees:

$$v_k=\frac{\sum_{i=1}^{N}\mu_{k,i}^{m}x_i}{\sum_{i=1}^{N}\mu_{k,i}^{m}}. \tag{6}$$

On the other hand, given the cluster center and the norm inducing matrix it is possible to induce desired membership degrees of the samples in the following way:

$$\mu_{k,i}=\frac{\frac{1}{D_{A_k}^2(x_i)}}{\sum_{j=1}^{K}D_{A_j}^2(x_i)}. \tag{7}$$

Gustafson-Kessel algorithm minimizes function given by (2) by iterative execution of the following steps:

1. Initialize U with random membership degrees
2. Calculate centers of clusters with (6)
3. Calculate new membership degrees with (7)
4. Calculate fuzzy covariance matrices using (4)

5. Calculate norms induced by those matrices using (3) and (5).
6. If membership degrees have changed more in this iteration than assumed termination value proceed to step 2.

In [10] a modification of this algorithm was proposed to restrict it to find clusters that are parallel to all the axes of the input-output space. In this method, we propose modification that results in finding clusters parallel to input space axes, and not necessarily parallel to the output axis. Gustafson-Kessel algorithm needs the number of clusters as the input parameter. We identify several models with different numbers of clusters and chose the best one according to the testing set error.

3.3 Modification of Gustafson-Kessel Algorithm to Obtain Desired Clusters

Clusters that are parallel to one of the axes tend to have significant non-zero variance along this axis and values of all covariances of this axis variable close to zero. As it was noticed in [10] a desired covariance matrix for clusters parallel to the axes is a diagonal matrix. In this work, however, we are looking for a wider class of clusters, namely clusters that are parallel only to the input-space axes.

To achieve this, we lessen a restriction on the covariance of the output variable, but still do not want to introduce any covariance between input variables. This leads to clusters induced by covariance matrix of a form (8)

$$F_0 = \begin{bmatrix} c_1 & 0 & \dots & 0 & 0 \\ 0 & c_2 & \dots & 0 & 0 \\ \dots & \dots & \dots & 0 & y_i \\ 0 & 0 & 0 & c_{n-1} & 0 \\ 0 & 0 & y_i & 0 & c_n \end{bmatrix}. \tag{8}$$

If F is a fuzzy covariance matrix, F_0 is a matrix that was created from F by putting 0 everywhere except for diagonal, and a single place not on a diagonal in last row and last column. Question to be answered is whether such a matrix is a valid covariance matrix. This is important because the covariance matrix needs to be positive semi-definite so it has positive determinant and the norm inducing matrix A obtained by (3) exists in $\Re^{n\times n}$. It is easy to show that in general (if more original elements were preserved) such matrix may not be a covariance matrix. However, restricting values in a way shown in (8) leads to covariance matrix in all cases.

Theorem 1. Let F be a covariance matrix. Matrix F_0 as in (8) created from F has properties of a covariance matrix.

Proof
Let us consider only two variables: *i-th* and *n-th* and their covariance matrix

$$F_{in} = \begin{bmatrix} c_i & y_i \\ y_i & c_n \end{bmatrix}. \tag{9}$$

From properties of covariance matrices we have:

$$\det(F_{in}) \geq 0 \Rightarrow c_i c_n - y_i^2 \geq 0 . \tag{10}$$

For F_0 to be a covariance matrix it is sufficient to be a symmetric positive semi-definite. A sufficient condition for a matrix to be positive semi-definite is that all determinants of the leading minors of the matrix are non-negative. Let F_0^j be a *j-th* leading minor of F_0. From definition (8) and form a fact that diagonal contains only non-negative numbers (variances) for all *n-1* leading minors:

$$\forall j = 1...n-1 : \det(F_0^j) = c_1 \cdot ... \cdot c_{n-1} \geq 0 . \tag{11}$$

The value of last minor's determinant (of whole matrix):

$$\begin{aligned} \det(F_0) &= c_1 \cdot ... \cdot c_n - c_1 \cdot ... \cdot c_{i-1} y_i \, c_{i+1} \cdot ... \cdot c_{n-1} y , \\ \det(F_0) &= c_1 \cdot ... \cdot c_{i-1} c_{i+1} \cdot ... \cdot c_{n-1} (c_i c_n - y_i^2) . \end{aligned} \tag{12}$$

From (10) and (12) we conclude $\det(F_0) \geq 0$, so the matrix (8) is positive semi-definite symmetric matrix.

It is worth noting that condition stated by the above theorem does not hold in a general case when we leave more than one non-diagonal non-zero value in the last row and column. We modify the Gustafson-Kessel algorithm so it finds clusters that have covariance matrices of a form (8) meaning that their axis are not necessarily parallel to the output axis. We do this by introduction of a step 4a to the algorithm:

4a. Convert covariance matrix to a form (8) by preserving only the largest covariance value in the last row/column.

The intuition for this approach is that we would like to preserve the most significant relation in the shape of the obtained cluster. Because of the conclusions of Theorem 1 all calculations performed in next steps of the algorithm may succeed.

4 Converting Clusters to Fuzzy Rules

Having obtained a set of fuzzy clusters canters and a set of norms induced for those clusters the task is to create membership functions of rules' antecedents. In this example we use asymmetric Gaussian type of membership functions but it must be noted that any classical type of membership functions would fit our method. The membership function is based on 4 parameters determining peak point and shape of left and right sides of the curve

$$f_{\sigma_1, c_1, \sigma_2, c_2}(x) = \begin{cases} e^{\frac{-(x-c_1)^2}{2\sigma_1^2}}, x < c_1 \\ e^{\frac{-(x-c_2)^2}{2\sigma_2^2}}, x > c_2 \\ 1, x \geq c_1 \wedge x \leq c_2 \end{cases} . \tag{13}$$

Some authors suggest projecting clusters onto each of the axes using fuzzy projection techniques [1, 10]. Curve fitting technique is applied to adjust membership function parameters so the membership degree of fulfillment of premise of the rule corresponding to a given cluster reflects the membership degree of the measured samples to that cluster. In TSK model we assume prod-type AND operator for rule premise. It should be noted that our technique applies also to different types of operators (e.g. min). The degree of fulfillment of a rule *j* is calculated as follows:

$$d_j(x) = \prod_{i=1}^{n-1} f^{(i)}(x^{(i)}), \tag{14}$$

where $f^{(i)}(x^{(i)})$ is a value of *i-th* function in a form of (13) for *i-th* coordinate of vector *x*.

We employ non-linear least squares optimization to obtain parameters of $f^{(i)}$. Objective function under minimization for rule *i* is given by (15):

$$e(\Sigma^{(i)}, C^{(i)}) = \sum_{j=1}^{N} (\mu_{i,j} - d_i(x))^2, \tag{15}$$

where $\Sigma^{(i)}, C^{(i)}$ are sets of parameters of membership functions used in rule *i* and $\mu_{i,j}$ is a membership degree of sample *j* to cluster *i.* If Jacobian of the destination function is analytically available we may use it in the calculations (this applies for standard Gaussian membership functions). In all other cases we may calculate Jacobian approximation using finite differences. In this work we used a subspace trust region approach [3] available in Matlab Optimization Toolbox, but also other least squares curve fitting methods could be applied. Numerical gradient based methods have a risk of converging to local minimum. In this case, however, we have a good starting point for the minimization. We can use cluster centers coordinates as initial values for the *C* parameters. Having cluster center defined as a very close to optimal we have less chance of converging to some local minima. Also we can calculate good initial guess value for Σ parameters, such that neighboring membership functions overlap. However, experiments have shown that this calculation is not necessary, as function converges without it.

5 Determination of Rule Consequents

Clusters detect grouping of sample data, which due to the construction of the covariance matrix may be approximated with a linear function. We use weighted least squares linear regression to identify parameters of the output function for a rule. Philosophy for use of weighted method is that samples with little membership degrees in a given rule are likely be evaluated by other rules so their output should not influence the output function to a big extent. And contrary, samples with high membership

degrees to a rule is evaluated primarily by this rule so they should have a big impact on the output function. Given the output function for a rule i we can formulate an error function for linear regression as shown below:

$$g_i(x^{(1)},...,x^{(n-1)}) = a_0 + \sum_{j=1}^{n-1} a_j x^{(j)} \quad ,$$
$$e_i(A) = \sum_{j=1}^{N} \mu_{i,j}[x_j^{(n)} - g_i(x_j^{(1)},...,x_j^{(n-1)})]^2 \quad . \tag{16}$$

6 Experimental Results

6.1 Box-Jenkins Gas Furnace

The input data [11] is a series of pairs <*u(t),y(t)*> where *u(t)* is a rate of flow of gas into furnace, and *y(t)* is a CO_2 concentration at the time *t*. With use of the method described in [1] we conclude that the output *y(t)* can be predicted with use of 3 variables: *y(t-1), u(t-4)* and *u(t-3)*. Variable *u(t-3)* does not significantly improve the performance of the model, while adding computation complexity. As in [6] we use only *y(t-1), u(t-4)* variables. As the learning data set we chose the first half of the samples, second half is used to calculate an approximation error.

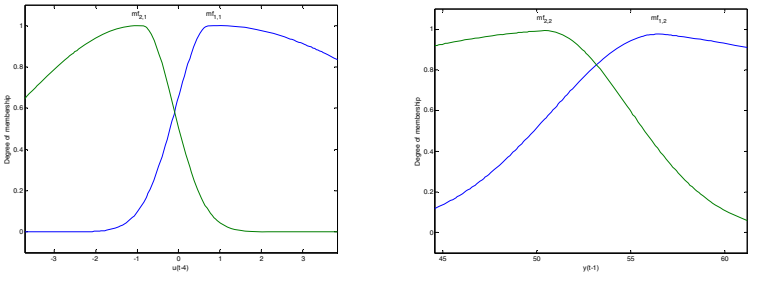

Fig. 1. Membership functions of input variables obtained for gas furnace problem

Membership functions that were obtained with our approach are depicted on Fig. 1. Resulting TSK rule base:

$$\begin{aligned} &\text{IF } u(t-4) \text{ IS } mf_{1,1} \text{ AND } y(t-1) \text{ IS } mf_{1,2} \text{ THEN } y = -1.38u(t-4) + 0.51y(t-1) + 25.78 \\ &\text{IF } u(t-4) \text{ IS } mf_{2,1} \text{ AND } y(t-1) \text{ IS } mf_{2,2} \text{ THEN } y = -1.39u(t-4) + 0.54y(t-1) + 23.86 \ . \end{aligned} \tag{17}$$

Root mean square error (RMSE) for testing set approximation with rule base (17) is 0.391. Table 1 compares our result with results obtained with different methods summarized in [6].

Table 1. Comparison of RMSE for gas furnace problem

Method	Num. of inputs	Num. of rules	RMSE
Pedrycz (84)	2	81	0.565
Xu (87)	2	25	0.572
Sugeno (91)	6	2	0.261
Sugeno (93)	3	6	0.435
Wang (96)	2	5	0.397
Delgado(99)	2	2	0.396
Rantala(02)	4	5	0.358
This method	**2**	**2**	**0.391**

As it can be seen our method provides good approximation accuracy with simple model. Delgado [6] model provides similar accuracy, but they use input membership functions in product space, hence not achieving that interpretability as our model. Wang [12] provides also similar accuracy model with significantly bigger number of rules. Sugeno [13] model providing the best accuracy uses significantly bigger input information so these two methods can not be directly compared with this example.

6.2 Non-linear Function Identification

As another benchmark we use a non-linear function with two input variables:

$$z = (1 + x^{-2} + y^{-1.5})^2, 1 \le x, y \le 5. \tag{18}$$

We use 50 random samples for learning and 200 other random samples for the system performance evaluation once it is learned. We determined a number of clusters to be 4. Figure 2 presents actual function vs. our modeling result. As we can see, our approximation does not perform well on the boundaries of the system space. This is due to the fact that very little random samples for learning were selected on the boundary.

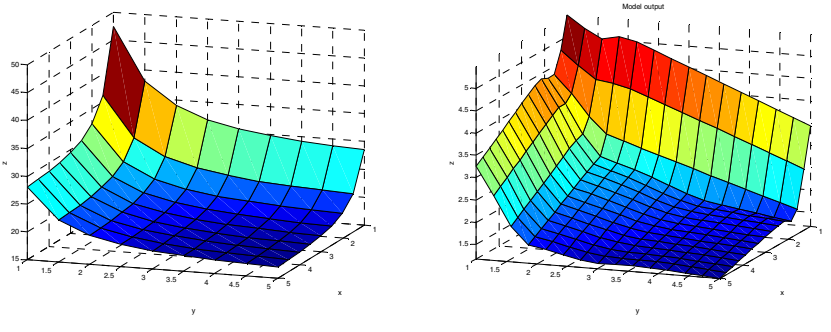

Fig. 2. Original function and its fuzzy model

Table 2. Comparison of RMSE for non linear function (18)

Method	Num. of rules	RMSE
Wang (96)	6	0.281
Delgado(99)	2	0.266
This method	**2**	**0.233**

Table 2 presents RMSE of our approach compared with other results from the literature.

6.3 Miles Per Gallon (MPG) Prediction

We run the test against standard miles per gallon prediction data set [14]. We divided the data set into two equal subsets and performed learning on one of them and measured the RMSE on the other half of the data. We selected 5 inputs for our model (displacement, horsepower, weight, acceleration and year) and 4 rules. Table below shows comparison of our result with other approaches found in the literature. It must be noted, that difference in MPG prediction are so small that can be due to the selection of random learning and testing sets. Our model is more complicated than Babuska [8] model but provides more interpretability as the other method uses transformation of input variables for the rules. Optimized ANFIS provides similar results to our method but with more complex underlying model.

Table 3. Comparison of RMSE for MPG approximation

Method	Inputs	Rules	Training RMSE	Testing RMSE
Jang (96) (linear reg.)	6	-	3.45	3.44
Babuska (02)	5	2	2.72	2.85
ANFIS	5	6	2.48	2.85
This method	**5**	**4**	**2.76**	**2.84**

7 Conclusions

We have shown that existing clustering based approaches to fuzzy modeling may still be improved. By modification of clustering algorithm in use we are able to obtain accurate fuzzy models and still preserve the interpretability. Additionally, it has been proven that curve fitting techniques combined with linear regression methods are a valid approach to convert clusters into fuzzy rules.

As numerical results show, our method provides satisfactory results very often delivering a simpler model than other approaches. Moreover, the method is extensible enough and can be easily adopted to find membership functions of other types than Gaussian. It also can be subject to later optimization, giving a very good basis for optimization starting point. Optimization of the model obtained with this method is in scope for our future work in this area.

References

1. Sugeno, M., Yasukawa, T.: A Fuzzy-Logic-Based Approach to Qualitative Modeling. IEEE Trans. on Fuzzy Systems 1(1), 7–31 (1993)
2. Jang, J.S.R.: ANFIS: Adaptive-network-based fuzzy inference systems. IEEE Trans. on System, Man and Cybernetics 23(3), 665–685 (1993)
3. Coleman, T.F., Li, Y.: An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds. SIAM Journal on Optimization 6, 418–445 (1996)
4. Rantala, J., Koivisto, H.: Optimized Subtractive Clustering for Neuro-Fuzzy Models. In: 3rd WSEAS International Conference on Fuzzy Sets and Fuzzy Systems (2002)
5. Wang, W., Zhang, Y.: On fuzzy cluster validity indices. Fuzzy Sets and Systems 158, 2095–2117 (2007)
6. Gomez-Skarmeta, A.F., Delgado, M., Vila, M.A.: About the use of fuzzy clustering techniques for fuzzy model identification. Fuzzy Sets and Systems 106, 179–188 (1999)
7. Parekh, G., Keller, J.M.: Learning the Fuzzy Connectives of a Multilayer Network Using Particle Swarm Optimization. In: IEEE Symposium on Foundations of Computational Intelligence, pp. 591–596 (2007)
8. Abonyi, J., Babuska, R., Szeifert, F.: Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models. IEEE Trans. on Systems, Man and Cybernetics 32(5), 612–621 (2002)
9. Gustafson, E.E., Kessel, W.C.: Fuzzy clustering with a fuzzy covariance matrix. In: Proc. of the IEEE Conference on Decision and Control, pp. 761–766 (1979)
10. Klawonn, F., Kruse, R.: Constructing a fuzzy controller from data. Fuzzy Sets and Systems 85, 177–193 (1997)
11. Box, G.E.P., Jenkins, G.M.: Time Series Analysis, Forecasting and Control. Holden Day, San Francisco (1970)
12. Langari, R., Wang, L.: Complex systems modeling via fuzzy logic. IEEE Trans. on Systems, Man, and Cybernetics 26(1), 100–106 (1996)
13. Sugeno, M., Tanaka, K.: Successive identification of a fuzzy model and its applications to prediction of a complex system. Fuzzy Sets and Systems 42(3), 315–334 (1991)
14. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2007)