Solving the Sales Prediction Problem with Fuzzy Evolving Methods

Dejan Dovžan  
Faculty of Electrical Engineering  
University of Ljubljana  
Tržaska 25  
Email: dejan.dovzan@fe.uni-lj.si

Vito Logar  
Faculty of Electrical Engineering  
University of Ljubljana  
Tržaska 25  
Email: vito.logar@fe.uni-lj.si

Igor Škrjanc  
Faculty of Electrical Engineering  
University of Ljubljana  
Tržaska 25  
Email: igor.skrjanc@fe.uni-lj.si

Abstract—This paper presents the solving of the petrol sales volume estimation problem given by the Task Force on Competitions, Fuzzy Systems Technical Committee IEEE Computational Intelligence Society. The solution using eTS, SAFIS and eFuMo method is presented. The results are compared to linear ARX model identified using RLS method. Results using static and prediction model are given. The paper also presents parts of new eFuMo method, which is based on recursive Gustafson-Kessel clustering.

Keywords: recursive Gustafson-Kessel clustering, evolving fuzzy model, eFuMo.

I. INTRODUCTION

Takagi-Sugeno fuzzy models are a powerful practical engineering tool for the modeling and control of complex systems. They expand and generalize the well known concept of gain scheduling. T-S models utilize the idea of linearization in a fuzzily defined region of the state space. Due to the fuzzy regions (clusters), the nonlinear system is decomposed into a multi-model structure consisting of linear models [1]. This enables the T-S fuzzy model to approximate virtually any nonlinear system within a required accuracy, provided that enough regions are given [2].

To build the T-S model the structure and the parameters of the local models must be identified [3]. Structure identification includes an estimation of the cluster centers (antecedent parameters), which is usually done by fuzzy clustering. Then for each cluster the local model’s parameters are estimated, which is usually done with the least-squares method [4].

The identification can be made off-line or on-line. On-line learning of fuzzy model has made significant progress in last few decades. Many methods already exist. Some of them are based on fuzzy logic (eTS [5], eTS [6], [7], simple_TS [8], +eTS [9], FLEXFIS [10], FLEXFIS+ [11]), others use neural networks, which realize the behavior of fuzzy model (EFuNN [12], [13], DENFIS [14], ANFIS [15], GANFIS [16], SOFNN [17], SAFIS [18], SCFNNS [19], NFCN [20], D-FNN [21], GD-FNN [22], SONFNN [23], NeuroFAST [24], RAN [25], ESOM [26], Neural gas [27], EFNF [28], GAP-RBF [29]).

The fuzzy model, that is usually used by the methods is first order Takagi-Sugeno (ANFIS, SONFNN, D-FNN, GD-FNN, DENFIS, eTS, NeuroFAST, SOFNN ...), zero order Takagi-Sugeno (SCFNNS, SAFIS, GAP-RBF, EFuNN) or generalized fuzzy model (GANFIS).

The methods differ also in the ability of adapting the fuzzy model and its structure. Some of the methods require an initial fuzzy model structure, which is then adapted. The adaptation includes only the adaptation of consequent and premise parameters (adaptive methods [16],[15]). Some of the methods include a mechanism for adding new clusters to the model structure (incremental methods [5]). Recently proposed methods include also mechanisms for merging, removing and splitting clusters (evolving methods).

In this paper solving of the petrol sales volume prediction is discussed. The learning data was provided by the Task Force on Competitions, Fuzzy Systems Technical Committee IEEE Computational Intelligence Society and are real historic data. For solving the problem the eTS and SAFIS methods are considered. Their results are compared to linear ARX model, obtained by LSE and to results obtained by eFuMo toolbox. The eFuMo (evolving fuzzy model) is our humble contribution to evolving methods. It is based on recursive fuzzy c-means and Gustafson-Kessel (GK) clustering methods [30],[31],[32]. Evolving nature was achieved by adding the evolving mechanisms (adding, merging,...) to recursive clustering.

The paper is organized in the following fashion. First a description of the eFuMo method is given. Because of the paper page limit only mechanisms that were used for solving this problem are presented. Next the problem is shortly discussed and solution is proposed. At the end results and conclusion are given.

II. EVOLVING FUZZY MODEL

As stated above, the evolving fuzzy model method is our humble contribution to a set of evolving methods. It is included in a eFuMo toolbox, which is an in-house developed toolbox for Matlab environment. The method is based on c-means and GK recursive clustering proposed in [30]. The method includes adding, merging, deleting and splitting mechanisms. The parameters of local linear models are identified by fuzzy least squares. The method considers two different vectors. One is clustering vector, for clustering (x_c), and the other is regression vector, for local models’ parameters estimation (x_k). In most cases these two vectors can be the same.

To cluster the input-output space, positions of cluster centers and variances of the data around the centers should be
calculated. Using the fuzzy c-means recursive algorithm this can be done by the following equations. First the change of centers positions is calculated as:
\[
\Delta \mathbf{v}_i(k+1) = \left( \frac{\mu_{ik+1}}{s_i(k+1)} \right)^\eta \left( \mathbf{x}_f(k+1) - \mathbf{v}_i(k) \right),
\]
where \( \mathbf{v}_i(k) \) is the center position at previous sample, \( \mathbf{x}_f(k+1) \) is the current clustering data vector and \( s_i(k+1) \) is the sum of past membership degrees, calculated as:
\[
s_i(k+1) = \gamma_o s_i(k) + \mu_{ik+1}^\eta.
\]
The initial \( s_i(0) \) is usually set to one. With initialization of \( s_i(0) \) to higher value slows down the initial speed of adaptation. The \( \gamma_o \) is the forgetting factor. The centers are then moved for
\[
\mathbf{v}_i(k+1) = \mathbf{v}_i(k) + \Delta \mathbf{v}_i(k+1) \quad i = 1, \ldots, c.
\]
The index \( i \) is the index of the cluster and \( c \) is the number of clusters. Membership \( \mu_{ik+1} \) of the current clustering vector \( \mathbf{x}_f \) is calculated as:
\[
\mu_{ik+1} = \frac{1}{\sum_{j=1}^{c} \left( \frac{d_{ik+1}}{d_{jk+1}} \right)^\frac{2\eta}{p}},
\]
where \( d_{ik+1} \) is the distance between clustering vector and the \( i \)-th cluster and \( \eta \) is the fuzziness (in most cases \( \eta = 2 \)). The distance is defined as:
\[
d^2_{ik+1} = (\mathbf{x}_f(k+1) - \mathbf{v}_i(k))^T \mathbf{A}_i (\mathbf{x}_f(k+1) - \mathbf{v}_i(k)),
\]
where \( \mathbf{A}_i \) depends on a distance that we use. For Euclidean distance \( \mathbf{A}_i \) is unit matrix, using Mahanalobis distance \( \mathbf{A}_i \) is equal to inverse fuzzy covariance matrix. For the distance used with GK clustering algorithm the \( \mathbf{A}_i \) is equal to fuzzy inverse matrix and determinant:
\[
\mathbf{A}_i = [\rho_i \ \det(\mathbf{F}_i)]^{1/p} \mathbf{F}_i^{-1},
\]
where \( \rho_i \) is usually set to one and \( p \) is the number of variables (number of elements of \( \mathbf{x}_f \)).

The fuzzy covariance is calculated by the following equation:
\[
\mathbf{F}_i(k+1) = \gamma_c \frac{s_i(k)}{s_i(k+1)} \mathbf{F}_i(k) + \frac{\mu_{ik+1}^\eta}{s_i(k+1)} \mathbf{D}_{F_i}
\]
\[
\mathbf{D}_{F_i} = (\mathbf{x}_f(k+1) - \mathbf{v}_i(k+1))^T (\mathbf{x}_f(k+1) - \mathbf{v}_i(k+1)).
\]
When using GK clustering algorithm recursive inverse fuzzy covariance matrix and recursive determinant calculation is needed. The equation for the inverse fuzzy variance matrix is following:
\[
\left[\mathbf{F}_i(k+1)\right]^{-1} = \frac{1}{\gamma_c} \frac{s_i(k+1)}{s_i(k)} \left[\mathbf{F}_i(k)\right]^{-1} - \frac{\mathbf{B}}{C}
\]
\[
\mathbf{B} = \left[\mathbf{F}_i(k)\right]^{-1} \mathbf{D}_{F_i} \left[\mathbf{F}_i(k)\right]^{-1}
\]
\[
C = \gamma_c \frac{s_i(k)}{\mu_{ik+1}^\eta} + \sum_{k=1}^{n} \left[\mathbf{F}_i(k)\right]^{-1} d_{F_i}
\]
\[
d_{F_i} = \mathbf{x}_f(k+1) - \mathbf{v}_i(k+1).
\]
The equation for recursive determinant calculation is given as:
\[
\det (\mathbf{F}_i(k+1)) = \left( \gamma_c \frac{s_i(k)}{s_i(k+1)} \right)^z \det (\mathbf{F}_i(k))(1 + A)
\]
\[
A = \frac{1}{\gamma_c} \frac{s_i(k)}{s_i(k+1)} \left[\mathbf{F}_i(k)\right]^{-1} d_{F_i},
\]
where \( z \) is the number of elements of the clustering vector. More detailed derivation of these equations can be found in [32].

Once the clusters are updated, the fuzzy recursive least squares are used to update the local linear models’ parameters. Different algorithms are proposed in [33, 5, 28, 34]. They are based on weighted recursive least squares. The equation for adaptation based on [34] are:
\[
\psi_i(k+1) = \beta_i [x_{k+1}(k+1), \ldots, x_{e_i}(k+1), 1]^T
\]
\[
y_i(k+1) = \beta_i x_{k+1}(k+1) \quad i = 1, 2, \ldots, c.
\]
\[
\mathbf{P}_i(k+1) = \frac{1}{\lambda_c + \psi_i^T(k+1) \psi_i(k+1)} \lambda_c \psi_i(k+1) \psi_i(k+1)^T
\]
\[
\theta_i(k+1) = \theta_i(k) + \mathbf{P}_i(k) \psi_i(k+1) [y_i(k+1) - \psi_i^T(k+1) \theta_i(k)]
\]
Parameters of \( i \)-th local model are denoted as \( \theta_i \), forgetting factor is denoted with \( \lambda_c \) and \( \beta_i \) denotes membership degree of current clustering vector to the \( i \)-th cluster. For \( \beta_i \) the membership degrees, calculated at cluster’s center update can be used. But in general they do not have a smooth transition between clusters. When identifying a process with a smooth non-linearity it’s better to recalculate the membership degrees using Gaussian functions. One can use radial base functions:
\[
\mu_{ij} = e^{-\frac{d_{ij}^2}{2\sigma_{im\sigma_{ij}}}},
\]
where \( d_{ij} \) is the distance between \( j \)-th component of \( i \)-th cluster and \( j \)-th component of clustering vector:
\[
d_{ij}^2 = (x_{f_j} - v_{ij})^2,
\]
\( \sigma_{ij}^2 \) is the \( j \)-th diagonal element of fuzzy covariance matrix and \( \eta_m \) is the overlapping factor. Membership degree \( \beta_i \) is calculated as a product of component membership degrees:
\[
\beta_i = \prod_{j=1}^{z} \mu_{ij},
\]
where \( z \) is the number of components of clustering vector \( \mathbf{x}_f \). More accurate membership degrees are obtained using elliptical functions:
\[
\beta_i = e^{-\frac{d_{ij}^2}{2\sigma_{im\sigma_{ij}}}},
\]
where \( D_i \) is calculated as:
\[
D_i^2 = (\mathbf{x}_f - \mathbf{v}_i)^T \mathbf{F}_i^{-1} (\mathbf{x}_f - \mathbf{v}_i).
\]
Note that these membership degrees \( \beta_i \) should be normalized.

The above equations represent the adaptation algorithm of eFuMo method. To achieve the evolving nature of the method,
mechanisms for adding, removing, splitting and merging of clusters must be included. The method also employs a condition for preventing moving of clusters to which membership degree of the current clustering vector is lower than a certain threshold. This threshold is set somewhere between 0.1 and 0.3. The evolving mechanisms are shortly described in the following subsections.

A. Adding the clusters

Adding is usually based on some distance measure of the current clustering vector to existing clusters or on membership degrees. If a current clustering vector has lower membership degree than a predefined threshold a new cluster is added. The eFuMo adding criterion is based on a normalized distances between the current clustering vector and existing clusters. Two different distances are considered. One is the Mahalanobis normalized distance and other is the normalized Euclidean distance. The second one is calculated for each component of clustering vector. For the Mahalanobis normalized distance the normalization vector is first defined. The vector consist of diagonal elements of fuzzy covariance matrix:

\[ d_{f\text{norm}} = \left[ \sqrt{F_{jj1}} \sqrt{F_{jj2}} \ldots \sqrt{F_{jjn}} \right]^T, \]

(22)
The normalized distance is calculated as:

\[ d_{\text{norm}} = \frac{(x_f(k) - v_i)^T F_{ij}^{-1} (x_f(k) - v_i))^{0.5}}{k_n (d_{f\text{norm}} F_{ij}^{-1} d_{f\text{norm}})^{0.5}}, \]

(23)
where \( k_n \) is user defined constant, normally set to \( 2\sqrt{P} \). \( i \) is the index of the closest cluster. If the distance under Eq. 23 is greater than one, a new cluster can be added. The Euclidean normalized distance is defined as:

\[ d_{ij\text{norm}} = \frac{d_{ij}}{k_n \sqrt{F_{jjij}}}, \]

\[ d_{ij} = |x_f(k) - v_{ij}|, \]

(24)
where \( i \) is the index of the closest cluster and \( j \) is the index of the component. Again the distance must be greater than one for adding the cluster. In this case there are several possibilities of defining the condition for adding. Either all component distances must satisfy the condition, either only one component distance satisfies the condition, or a certain number of component distances satisfy the condition.

To prevent the adding of the clusters based on outliers, additional condition was introduced as in [35]. For the cluster to be added at least \( N \) consequent samples must satisfy the condition for adding. The number \( N \) is normally set from 3 to 10, depending on data.

1) Initialization of new cluster: The new cluster center is positioned at the current clustering vector. The fuzzy covariance matrix is initialized as diagonal matrix where the distances to closest cluster are considered. The diagonal elements are defined as:

\[ f_{\text{new,ij}} = -\frac{d_{ij}^2}{2p_m \ln(\epsilon_\beta)}, \]

(25)
where \( \epsilon_\beta \) is a user defined constant, normally set to 0.15. If the distance \( d_{ij} \) is smaller than standard deviation (\( \sqrt{F_{jjij}} \)), then a diagonal element is equal to a diagonal element of the closest cluster (\( f_{\text{new,ij}} = f_{ij} \)). In the same manner a initial fuzzy covariance for the first cluster can be set considering the expected number of clusters and data range. Next the parameters of new local model can be initialized using weighted mean:

\[ \theta_{i+1} = \frac{\sum_{i=1}^c \omega_{ij} \theta_{ij}}{\sum_{i=1}^c \omega_{ij}}, \]

(26)
where \( i \) is the index of cluster and \( j \) is the parameter index. Weights \( \omega_{ij} \) can be equal to normalized membership degrees to other clusters, or equal to normalized membership degrees to other clusters combined with variance of parameters:

\[ \omega_{ij} = \frac{1}{\sigma^2_{ij}}, \]

(27)
where \( \sigma^2_{ij} \) is the \( j \)-th diagonal element of least squares covariance matrix of \( i \)-th cluster.

B. Removing the clusters

Removing is in our case based on cluster’s support [9]. Support \( N_{si} \) is defined as the number of samples that have the maximal membership to the certain cluster. The cluster is removed if in certain user defined time (\( k_{\text{delay}} \)) does not receive a certain number of support samples. The cluster can also be deleted if its support, normalized with their age (\( S_{ni} \)), is lower than a fraction of mean. Age \( a_i \) is defined as a number of samples from the creation \( k_i \) and current sample \( k \):

\[ a_i = k - k_i, \]

(28)
\[ S_{ni} = \frac{N_{si}}{a_i}, \]

(29)
The condition for removing can be written as:

IF \( S_{ni} < \varepsilon \text{ mean}(S_n) \)
OR (\( N_{si} < S_{\text{thr}} \) AND \( k < k_i + k_{\text{delay}} \))
THEN remove \( i \)-th cluster.

C. Merging the clusters

Merging algorithm is meant to merge the clusters that are either close or have the same local model’s parameters. The eFuMo method considers merging based on membership degrees of clusters to each other [28], and merging based on correlation method [11].

Basic merging is based on a normalized Mahalanobis distance:

\[ d_{kj\text{norm}} = \frac{(v_k - v_i)^T F_{ij}^{-1} (v_k - v_i))^{0.5}}{(d_{f\text{norm}} F_{ij}^{-1} d_{f\text{norm}})^{0.5}}. \]

(31)
Clusters are merged if the distances \( d_{kj\text{norm}} \) and \( d_{ki\text{norm}} \) are below the threshold \( k_{d} \) and the ratio between distances is lower than the threshold \( k_{r} \).

IF \( d_{kj\text{norm}} < k_d \) AND \( d_{ki\text{norm}} < k_d \) AND \( |1 - \frac{d_{kj\text{norm}}}{d_{ki\text{norm}}}| < k_r \)
THEN merge \( i \)-th and \( k \)-th cluster.

(32)
Threshold $k_d$ is defined based on the allowed maximal membership degree $\varepsilon_\beta$ of $k$-th cluster to $i$-th cluster and vice versa.

$$k_d = \sqrt{-2\eta_m \ln(\varepsilon_{\beta m})}.$$  \hfill (33)

The $k_r$ should be set to around 0.1 and $\varepsilon_{\beta m}$ to around 0.8. This rule merges only close clusters. In general, the clusters can be further apart, but still have the same shape and local model’s parameters. To merge this clusters, a correlation between membership degrees to clusters (clusters’ firing levels) ($C_{beta_{ik}}$) are monitored as in [11]. If the correlation is higher than a threshold value, the clusters are considered for merging. This condition is combined with the condition of angles between parameters ($\phi_{ik}$) [11] and clusters’ support from the time of last change in the number of clusters ($N_{sl_i}$). Each time some of the evolving mechanisms changes the number of clusters the correlation value $C_{beta_{ik}}$ and support $N_{sl_i}$ are reset to zero. Clusters can also be merged if the angle between parameters are small and there are no other clusters between them.

The initialization of new cluster position and variance is done as suggested in [28].

D. Splitting the clusters

Splitting is currently meant to fine tune the evolving fuzzy model. It can add clusters in input-output space, where the output model error is higher than predefined threshold. The eFuMo method tracks the mean error for each cluster. At each sample, if the sample does not satisfy the distance condition for adding, the output error of the current model is calculated. Then the model error is divided among the clusters, depending on the membership degrees of the current clustering vector. If the error of one of the clusters exceeds the defined threshold, this cluster is split. The parameters of a new and old local model stay the same. Centers of new and old cluster are positioned based on a fuzzy covariance matrix. The new positions are calculated as:

$$v_{i1} = v_i + df_{i,\text{norm}}$$ \hfill (34)

$$v_{i2} = v_i + df_{i,\text{norm}}$$

This approach of initialization is presented also in [36]. Initialization can also be based on eigenvectors of fuzzy covariance matrix [37]. The fuzzy covariance matrix is divided by two. The error threshold is time decaying. The user specifies the maximal and the minimal value and decay constant for the threshold - the same as in [21]. The error is set to zero when number of clusters changes.

The eFuMo method also considers a general time delay for the evolving mechanisms. The delay is user specified. If change in number of clusters occurs, the evolving mechanisms are stopped for this specified delay.

III. PROBLEM AND SOLUTION

The goal of the competition is to solve the petrol volume sales estimation. The given data were station’s own price of the petrol, competitors’ prices and the day and the month of the data sample. The given data included 25 different petrol stations, where the daily prices and sales of petrol for about one year were provided.

First step in solving the problem was defining the least squares regressor and clustering vector. Based on our observations, the petrol sales is day dependent and relative price dependent. In general the consumption of gas doesn’t lower much with the price raise, because we highly depend on our car and therefore the petrol. This assumption probably doesn’t hold for a long period of time. If the prices will constantly grow, people will search for alternative sources of energy/feul. However since we are modeling petrol sales for one year period, the above assumption seems reasonable. Decision on which petrol station we buy gas is probably made by comparing the minimum price of petrol to the petrol price at each station. The decision is then made based on this difference and local position of the gas station. Based on that the regression vector for least squares was formed as:

$$x_k(k) = [d(k), \Delta p_{min}(k-1), \Delta p(k), p_{own}(k)]^T,$$ \hfill (35)

where, $d$ is the dummy vector for days. The vector has seven variables, each representing one day in a week. If the current data sample is from Monday, then the first element of vector $d$ is one and others are zero. The $\Delta p_{min}$ is the difference between the minimal price and the price of own petrol station and $\Delta p$ is the difference between competitive prices and our own price ($p_{own}$). The output of the model is volume of sale $v_s$.

The variables differ in their maximal and minimal values. For clustering and evolving methods it is recommended that the variables are normalized. Usually this is done by using variance and mean value of the variable or minimal and maximal value of the variable. This is usually not possible when an online identification is made, since there is little prior knowledge of the system. However, one can estimate minimal and maximal value. In order to come as close as possible to a real life scenario, we decided on a fixed normalization of variables, meaning that all data sets were normalized with the same value as:

$$x_{\text{norm}} = k \cdot x - n.$$ \hfill (36)

Normalization constant for variables are given in Table I.

<table>
<thead>
<tr>
<th>Variable</th>
<th>$k$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta p_{min}$</td>
<td>7.5</td>
<td>0.9</td>
</tr>
<tr>
<td>$\Delta p$</td>
<td>5</td>
<td>0.5</td>
</tr>
<tr>
<td>$p_{own}$</td>
<td>3</td>
<td>-2.7</td>
</tr>
<tr>
<td>$v_s$</td>
<td>8000</td>
<td>1</td>
</tr>
</tbody>
</table>

Next step was applying the eFuMo method on the data sets using vector under Eq. 35 for clustering and least squares regressor. Using Mahanalobis distance for adding we found that very few clusters were added (one or two on average). This can be due the fact that either adding distance is not adequate, either clustering vector is not adequate or the system can actually be approximated with a linear model.
Results of the modeling are shown in the next section. The problem was solved using eTS method, SAFIS method, eFuMo method and LSE method (for linear model identification). The results are compared. It should be noted that SAFIS was programmed from an article, therefore there could be some mistakes in the program. Also both methods were tuned as proposed in papers. Since we are no expert for this two methods we are convinced that they can give better results as presented in this paper.

IV. RESULTS

As mentioned above the parameters of eTS method were set as in [5] and of SAFIS method as in [31]. For the eFuMo method Gustafson-Kessel on-line clustering was chosen. The clustering vector was chosen as:

\[ x_f(k) = [\Delta p_{\min}(k-1), p_{own}(k), v_s(k-1)]^T. \]  

The condition for adding the clusters was based on component distances. The condition for adding is satisfied if at least one component distance satisfies the condition under Eq. 24. A new cluster is added if three consequent samples satisfy the adding condition. The threshold for merging was set to \( k_r = 0.1 \) and \( \varepsilon_{\beta_M} = 0.8 \). The correlation threshold was 0.96 and angle threshold was 5 degrees. The removing of the cluster was done when its support was less than 1 percent of the mean support. Maximal threshold for splitting was set to 1 and minimal to 0.1 with the decay factor 0.99. The threshold for stopping the adaptation of cluster centers was set to 0.3. The initial value of \( P_i \) was a unit matrix and the value of \( s_i \) was 40. The forgetting factors were set to one. Local fuzzy least squares were used for linear parameter identification [5].

The mean absolute error for learning data sets and number of generated clusters are given in Tables II, III, IV, V, VI. Under eTS\(_g\) are the results, given by eTS method, using global fuzzy weighted least squares. Results using local fuzzy weighted least squares are under eTS\(_l\). This algorithm was also used with the eFuMo method (eFuMo\(_l\)).

The mean of MAE for all data sets are given in Table VII. It can be seen that the eFuMo method can approximate sales volume at least as good as eTS method. With the eFuMo method also different algorithms for linear parameters identification were tested. The results are in Table IX, where
the mean values of MAE are given. The tested algorithms were global weight least squares (eFuMo) and local weight least squares (eFuMo) \[5\], weight least squares presented in \[28\] (eFuMo), in \[31\] (eFuMo) and in \[33\] (eFuMo).

The obtained results with evolving methods are not as good as we expected. Partly because of outliers in data sets (the MAE was calculated including the outliers) and partly because of small numbers of data (evolving methods usually need a larger set of data to properly identify the fuzzy model). The accuracy of the fuzzy model can be further improved using one step prediction model. The results for the one step prediction model are given in Table IX.

In Figure 4 the sales volume and its estimation using static models for the third data set is given. In Figure 5 normalized inputs for static fuzzy model are presented. The clusters for the same set are shown in Figures 1, 2 and 3. The clusters were generated with eFuMo method.

In Figure 6 an error histogram for the third data set is given. Not considering a few large error amplitudes, which are the results of outliers in real data, the error distribution can be defined as the Gaussian. From this we can conclude that the model structure is adequate. It was noticed that the sales volume data have some spikes. Usually somewhere between the 70th and the 80th sample. The spikes can not be predicted using given prices. They are probably date dependent (probably some holidays). This assumption is however not certain. In order to be sure that the assumption holds we would need data for at least three years.

Adding the dummy variable for month and competitive prices to the regression vector, the accuracy of prediction model can be further improved for about one percent. However, the fuzzy model in this case is less transparent and more comprehensive.

V. CONCLUSION

The paper describes solving of the volume sales estimation problem. Fuzzy models identified with evolving methods are compared to linear models identified with LSE method. The idea behind the solution was to use as few inputs as possible to make the fuzzy model transparent. The same method with the same setting was used on all data sets. Only one sweep through data was made. The results show that the given data can be adequately approximated with a linear model. With the use of the fuzzy model not much is gained on accuracy. In our case the eFuMo method performed better than other methods. However, it can not be said that eFuMo method is better than others, since other methods were not tuned by an expert. The working data sets in the presented problem were quite small. In general, the eFuMo method is meant for larger data sets.

<table>
<thead>
<tr>
<th>method</th>
<th>mean MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>eTS₁</td>
<td>793.82</td>
</tr>
<tr>
<td>eTS₂</td>
<td>858.22</td>
</tr>
<tr>
<td>SAFIS</td>
<td>973.61</td>
</tr>
<tr>
<td>LSE</td>
<td>790.76</td>
</tr>
<tr>
<td>eFuMo</td>
<td>777.97</td>
</tr>
</tbody>
</table>

TABLE VII
MEAN OF MAE FOR ALL DATA SETS WITH THE STATIC MODEL.

<table>
<thead>
<tr>
<th>method</th>
<th>mean MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>eFuMo₀₀</td>
<td>792.38</td>
</tr>
<tr>
<td>eFuMo₀₁</td>
<td>777.97</td>
</tr>
<tr>
<td>eFuMo₀₀₀</td>
<td>791.31</td>
</tr>
<tr>
<td>eFuMo₀₀₀₀</td>
<td>785.04</td>
</tr>
<tr>
<td>eFuMo₀₀₀₀₀</td>
<td>870.29</td>
</tr>
</tbody>
</table>

TABLE VIII
MEAN OF MAE FOR ALL DATA SETS FOR STATIC MODEL WITH DIFFERENT WEIGHTED LEAST SQUARES ADAPTATION.

In Figure 1 the cluster positions for the third data set are given. In Figure 2 normalized inputs for static fuzzy model are presented. The clusters for the same set are shown in Figures 1, 2 and 3. The clusters were generated with eFuMo method.

<table>
<thead>
<tr>
<th>method</th>
<th>mean MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>eFuMo₀₀</td>
<td>738.93</td>
</tr>
<tr>
<td>eFuMo₀₁</td>
<td>720.81</td>
</tr>
<tr>
<td>eFuMo₀₀₀</td>
<td>734.48</td>
</tr>
<tr>
<td>eFuMo₀₀₀₀</td>
<td>725.68</td>
</tr>
<tr>
<td>eFuMo₀₀₀₀₀</td>
<td>816.49</td>
</tr>
</tbody>
</table>

TABLE IX
MEAN OF MAE FOR ALL DATA SETS FOR PREDICTION MODEL WITH DIFFERENT LEAST SQUARES ADAPTATION.
However, the results show that it could also perform quite well on smaller data sets using proper settings. The eFuMo toolbox is currently in developer’s version. We still need to make some tests and experiments to evaluate the performance of the method. The user friendly final version of the toolbox should be available for download at the end of this summer. The toolbox will be downloadable from our web site http://msc.fe.uni-lj.si/ with help files and settings suggestions.

REFERENCES

Fig. 6. Error histogram.

1993.
generalized adaptive neuro-fuzzy inference systems,” IEEE Trans. on
creating self-organizing fuzzy neural networks,” Neural Networks,
[18] H. J. Rong, N. Sundararajan, G. B. Huang, and P. Saratchandran,
“Sequential adaptive fuzzy inference system (SAFIS) for nonlinear
system identification and prediction,” Fuzzy Sets and Syst., vol. 157,
network speed controller for permanent-magnet synchronous motor
of fuzzy rules by generalized dynamicc fuzzy neural networks,” IEEE
inference network and its applications,” IEEE Trans. on Fuzzy Syst.,
ART-based structure and parameter learning TSK model,” IEEE Trans.
learning, data analysis and modeling,” in Proc. ICANN 2000 Neural Net-
works Neural Computing New Challenges Perspectives New Millenium,
clustering as a basis for evolving neuro-fuzzy modeling,” Evolving
[29] G.-B. Huang, P. Saratchandran, and N. Sundararajan, “A recursive growing
and pruning rbf (gap-rbf) algorithm for function approximations,”
in Proceedings of The Fourth International Conference on Control and
Automation (ICCA03), Montreal, Canada, June 2003, pp. 10 – 12.
an adaptive fuzzy model of a hybrid semi-batch reactor,” Control
[31] ——, “Recursive fuzzy c-means clustering for recursive fuzzy identifi-
159 – 169, 2011.
[32] ——, “Recursive clustering based on a gustafsonkessel algorithm,”
[33] J. de Jesús Rubio, EVOLVING INTELLIGENT SYSTEMS Methodology
and Applications. Hoboken, New Jersey: John Wiley & Sons, 2010,
ch. STABILITY ANALYSIS FOR AN ONLINE EVOLVING NEURO-
FUZZY RECURRENT NETWORK, pp. 173–199.
[34] S. Blažič, I. Škrjanc, S. Gerkšič, G. Dolanc, S. Strmčnik, M. B. Hadjiski,
and A. Stathaki, “Online fuzzy identification for an intelligent controller
based on a simple platform,” Engineering Applications of Artificial
dynamic version of fuzzy k-nearest neighbours to monitor evolving
cepts and Applications,” ser. Studies in Fuzziness and Soft Computing
Series, J. Kacprzyk, Ed. Berlin Heidelberg: Springer-Verlag, January
[37] I. Škrjanc, B. Hartmann, O. Banfer, O. Nelles, A. Sokja, and L. Teslic,
“Supervised hierarchical clustering in fuzzy model identification,” Fuzzy