Procedia Computer Science

Volume 53, 2015, Pages 191–198

2015 INNS Conference on Big Data

Evolving Gustafson-Kessel Possibilistic c-Means Clustering

Igor Škrjanc[1] and Dejan Dovžan[2]

[1] Laboratory of Modelling, Simulation and Control
Faculty of Electrical Engineering, University of Ljubljana
Tržaška 25, SI-1000 Ljubljana, Slovenia igor.skrjanc@fe.uni-lj.si
[2] Laboratory of Modelling, Simulation and Control
Faculty of Electrical Engineering, University of Ljubljana
Tržaška 25, SI-1000 Ljubljana, Slovenia
dejan.dovzan@fe.uni-lj.si

Abstract

This paper presents an idea of evolving Gustafson-Kessel possibilistic c-means clustering (eGKPCM). This approach is extension of well known possiblilistic c-means clustering (PCM) which was proposed to address the drawbacks associated with the constrained membership functions used in fuzzy c-means algorithms (FCM). The idea of possiblistic clustering is appealing when the data samples are highly noisy. The extension to Gustafson-Kessel possibilistic clustering enables us to deal with the clusters of different shapes and the evolving structure enables us to cope with the data structures which vary during the time. The evolving nature of the algorithm makes it also appropriate for dealing with big-data problems. The proposed approach is shown on a simple classification problem of unlabelled data.

Keywords: Big-data clustering, Stream data, Evolving Clustering, eGKPCM, Evolving Classifier.

1 Introduction

The fuzzy c-means clustering (FCM) proposed by Dunn [9] and generalized by Bezdek [5] fails in dealing with strongly noisy data. The reason for that is the concept of membership values which are normalized distances between the current samples and the prototypes/clusters. Meaning that the outliers will also have a membership degree to the closest cluster that is rather high. To avoid this the membership degrees in [15] are interpreted as a relative typicality. The samples which are far from all of the prototypes and could therefore be outliers are accordingly weighted. This means that the membership values for such data samples are low and do not contribute as much as the good data to the prototypes' means and covariance matrices.

To overcome this disadvantage of FCM the possibilistic c-means (PCM) clustering algorithm was introduced by Krishnapuram and Keller [11]. The basic idea is in relaxation of the restriction which is given by the relative typicality (membership degrees) proposed in FCM algorithm. The disadvantages of the algorithm is the sensitivity to the initial values of its parameters and

Selection and peer-review under responsibility of the Scientific Programme Committee of INNS-BigData2015

doi:10.1016/j.procs.2015.07.294

the possible coincidence of several prototypes/clusters in the same location. The solution which avoids the prototypes at the same position is presented in [17]. A hybrid algorithm where both algorithms, FCM and PCM are combined in possiblistic fuzzy c-means clustering (PFCM) is proposed in [15]. This is an attempt to combine both algorithms to estimate the prototypes of data sets as a function of the internal resemblance and external dissimilarity. These algorithms are all based on Euclidian distance and therefore the identified clusters are of hyper-spherical shapes. More flexibility in the sense of shape is given by introducing the Mahalanobis distance into the algorithm as proposed in [14]. This allows detection of the hyper-ellipsoidal form of clusters.

In recent years there has been an increased interest to the on-line nonlinear model identification that combine different approaches from fuzzy logic. The main difference between them is in their clustering algorithms. Most of the on-line clustering algorithms are derived from the off-line clustering algorithms. For example in [10] and in [7] on-line Gustafson-Kessel clustering algorithm is derived, in [4] an on-line version of subtractive clustering technique [6] is derived, in [16] the recursive method based on Gath-Geva clustering algorithm is given and in [13] recursive possibilistic fuzzy modeling approach is given. Most of the on-line clustering algorithms are based on the Euclidian distance [3], [1] making the identified clusters of a hyper-spherical shapes [2]. A step forward was introducing the recursive calculation of fuzzy covariance matrix its inverse and determinant [7], [10], allowing the use of Mahanalobis distance in the recursive and evolving fuzzy model identification and clustering [8]. More on evolving approaches and on-line clustering can be found in [12].

Following the requirements to have the algorithm which could deal with different cluster shape, volume and also with a big-data problems in an on-line manner, the evolving Gustafson-Kessel Possibilistic Fuzzy c-means clustering algorithm (eGKPFCM) is given in our paper. The proposed approach tries to deal with a common issue of the on-line and evolving fuzzy model identification and clustering. Most of the currently available methods are more or less prone to outliers. During the research it was found out that it is almost impossible to do on-line clustering with absolute typicalities without a buffer. Therefore the proposed method, in contrast with above mentioned methods, employs a buffer for atypical samples. New prototypes are then identified from the buffered data.

The paper is organized as follows: after the introduction the Gustafson-Kessel Possibilistic c-Means Clustering algorithm is given. In the next section the Evolving Gustafson-Kessel Possibilistic c-means clustering is presented with and demonstrated on two generic examples. At the end some conclusions are made.

2 Gustafson-Kessel Possibilistic c-Means Clustering algorithm

The main objective in clustering is to find the internal structure of data set to partition this set into c different clusters. The members of these clusters are among each other more similar than in comparison with the elements from the other clusters. The similarity is given by different measures.

The possibilistic clustering aims at minimizing the criterion function

$$I_{PCM} = \sum_{j=1}^{c}\sum_{k=1}^{n}\mu_{jk}^{\eta}d_{jk}^{2} + \sum_{j=1}^{c}\nu_{j}^{2}\sum_{k=1}^{n}(1-\mu_{jk})^{\eta} \tag{1}$$

where c stands for the number of clusters, n is the numbers of data samples, d_{jk}^2 is the distance

measure from the $j-th$ cluster prototype to the $k-th$ observed sample, μ_{jk} is the typicality of the $k-th$ sample regarding the $j-th$ cluster prototype and ν_j stands for the fuzzy variance of the $j-th$ cluster.

By solving the optimization problem using Picard iteration and assuming Euclid distance similarity measure the cluster prototype, with the dimension $1 \times m$, is defined as follows:

$$v_j = \frac{\sum_{k=1}^{n} \mu_{jk}^{\eta} z_k}{\sum_{k=1}^{n} \mu_{jk}^{\eta}} \tag{2}$$

where z_k stands for data sample and μ_{jk} stands for the typicality which is calculated as follows

$$\mu_{jk} = \left(1 + \left(\frac{d_{jk}^2}{\nu_j^2} \right)^{\frac{1}{\eta-1}} \right)^{-1}, \quad \nu_{jk}^2 = \frac{\sum_{k=1}^{n} \mu_{jk}^{\eta} d_{jk}^2}{\sum_{k=1}^{n} \mu_{jk}^{\eta}} \tag{3}$$

where η denotes the fuzziness parameter. The Mahalanobis distance which is further used in the calculation of the typicality is defined as

$$d_{jk}^2 = (z_k - v_j) A_j (z_k - v_j)^T, \quad A_j = (\rho_j \mid F_j \mid)^{\frac{1}{m}} F_j^{-1} \tag{4}$$

where the variable $\rho_j = \mid A_i \mid$ and F_i is the fuzzy covariance matrix of dimension $m \times m$ defined as follows

$$F_j = \frac{\sum_{k=1}^{n} \mu_{jk}^{\eta} (z_k - v_j)^T (z_k - v_j)}{\sum_{k=1}^{n} \mu_{jk}^{\eta}} \tag{5}$$

In our case the typicality variable ν_{jk} (Eq. 3) was fixed to value δ. The δ is in our case a tuning parameter which depends on the expected form of clusters, i.e., if we expect the clusters along the line, or with very big ratio between the maximal and the minimal eigenvalue of the cluster covariance matrix, this factor should be small.

This kind of clustering algorithm may overcome the problems of clustering a noisy data sets, what can happen when using FCM clustering algorithm. Also different shapes of clusters can be detected in the data. GKPCM is on the other side very sensitive to good initialization and it results very often in coincident clusters. This can be an advantage, because we can initialize the algorithm to a larger number of clusters, bigger than the possible number of clusters, and than we can extract the real clusters from that obtained set, where also some degenerated and coincident clusters appears. This idea was used in our approach in the evolving algorithm.

2.1 Problems of cluster number in GKPCM

The problems connected to the cluster number in GKPCM will be discussed next. If we assume the number of clusters in the observed set as c and the number of assumed maximal number of clusters defined as c_{max}, where $c_{max} > c$, than we will get $c_{max} - c$ cluster centers with very special characteristics. Either they will coincide with the centers of the real clusters or they will be the centers of degenerated clusters which appear due to the noise in data set.

To show this problem an example with the data set with two explicit clusters in heavily noisy data space will be used. The data set is generated by two different processes. The first one has $N_1 = 25$ samples (x_{11}, x_{12}) second one has also $N_2 = 25$ (x_{21}, x_{22}):

$$x_{11} = \mathcal{N}(10, 0.75), \; x_{12} = x_{11} + \mathcal{N}(10, 0.25), \quad x_{21} = \mathcal{N}(-4, 0.50), \; x_{22} = \mathcal{N}(4, 0.25) \tag{6}$$

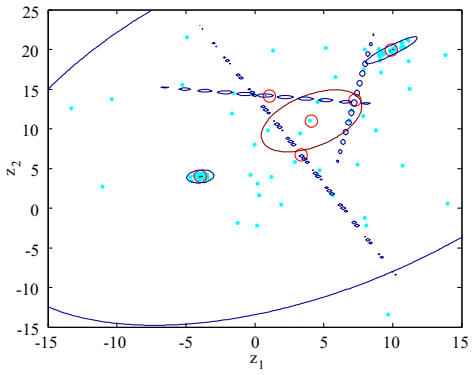

Figure 1: Example of GKPCM clustering without elimination of coincident and degenerated clusters.

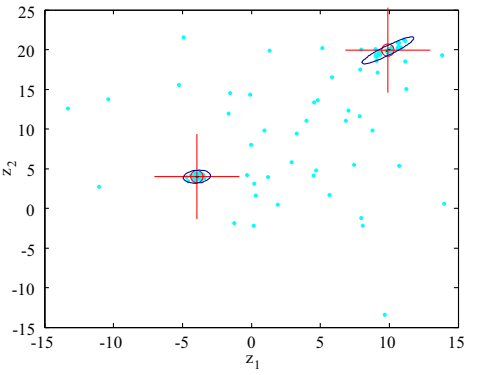

Figure 2: Example of GKPCM clustering with elimination of coincident and degenerated clusters.

The data set is corrupted by strong noise which consists of 50 samples and is described as

$$x_{N1} = \mathcal{N}(5, 8.00)\,,\; x_{N2} = \mathcal{N}(10, 8) \tag{7}$$

All these data samples are put into the data matrix $Z = [z_1\ z_2]$ of dimension 100×2, where $z_1 = \left[x_{11}^T\ x_{21}^T\ x_{N1}^T\right]^T$ and $z_2 = \left[x_{12}^T\ x_{22}^T\ x_{N2}^T\right]^T$. The pre-assumed maximal number of clusters is equal to $c_{max} = 6$, the fuzziness parameter is equal $\eta = 1.5$, the maximal allowed trace of the cluster covariance matrix is defined as $t_{max} = 1.25$ and the cluster shape parameter is equal $\delta = 1$. By using the GKPCM algorithm we get the solution on Fig. 1. The solution is due to the stochastic nature of the algorithm not always the same. In this case 2 cluster centers are obtained at the real clusters positions, $v_{11} = 9.8061$; $v_{12} = 19.8345$ and the second one at $v_{21} = -4.0829$; $v_{22} = 3.9328$. The rest four clusters are in this case degenerated, due to the noisy samples and outliers. In this case the false clusters are detected by the trace of the covariance matrices which are far bigger than the traces of the covariance matrices for the real two clusters. The false clusters can be also detected observing the form of the clusters, which is given by the ratio between the eigenvalues of the covariance matrices or by the number of the data samples in the cluster n_j which should be bigger than pre-defined minimal allowed number n_{min}. In the observed case the differences in trace are significant and are therefore used to detect the unacceptable clusters. The unacceptable clusters are then removed. Fig. 2 shows the two remaining clusters.

The pseudo-code of the GKPCM clustering algorithm is given in Algorithm 1.

3 Evolving Gustafson-Kessel Possibilistic c-means clustering

When dealing with the data stream in identification or in classification problem then the data should be processed in an on-line manner. This means that the centers of prototypes and the covariance matrices of the clusters are updated concurrently by taking into account the current data sample. This data sample can be similar to one of the previous cluster prototypes or it can

Algorithm 1 Pseudo-code of the GKPCM clustering algorithm.

Definition of the maximal number of prototypes c_{max}, the fuzziness parameter η, the termination criteria ϵ, the parameters of the cluster shape δ, the parameters of the accepted maximal trace of the cluster covariance matrix t_{max}, the accepted maximal ratio between the biggest and the smallest eigenvalue γ_{max} and the minimal allowed number of data samples in the cluster n_{min}.
Initialization of the membership degrees for c_{max} clusters by using FCM algorithm and definition of the cluster volume ρ_j.
repeat
  Computation of the cluster centers using Eq. 2,
  Computation of the covariance matrices and the inner distance norm using Eq. 5,
  Computation of the distances using Eq. 4,
  Computation of the absolute typicality using Eq. 3 and ($\nu_{jk} = \delta$),
until $\|\Delta U\|_\infty < \epsilon$, where U is matrix of all typicalities
Elimination of the cluster centers which have:

$$trace\,(F_j) > t_{max}\ or\ \frac{\lambda_{j_{max}}}{\lambda_{j_{min}}} > \gamma_{max}\ or\ n_j < n_{min}, j = 1, ..., c_{max}.$$

be different from all of them. If it is different, than it could be a sample from a new cluster or it could be a noisy data which does not belong to any of the clusters. This implies the solution where the on-line clustering is combined with a batch clustering approach. If the data samples does not belong to any of the clusters, it goes to the buffer and when the buffer is full, the batch version of GKPCM is used to find out if there is any new cluster inside the buffer. If it is, the initial center and initial covariance matrix are added to the previous centers and covariance matrices set. When dealing with strongly noisy environment and having a lot of outliers this can be a good and possible the only robust solution for on-line clustering.

At the beginning of the algorithm the following parameters should be defined: the number of samples in the buffer n_{buf}, the minimal typicality of the data sample μ_{min}, the fuzziness parameter η, the termination criteria ϵ, the parameters of the cluster shape δ, the parameters of the accepted maximal trace of the cluster covariance matrix t_{max} and the accepted maximal ratio between the biggest and the smallest eigenvalue γ_{max},c_{max} is the maximal possible number of cluster centers in the data buffer and the forgetting factor is defined as γ.

In the initialization also the first n_{buf} data samples are used to calculate the initial cluster centers using GKPCM algorithm.

When a new data sample is obtained, the typicalities of the sample to the current set of cluster prototypes are calculated (Eq. 3). If the maximal typicality is bigger than μ_{min} than the sample belongs to the cluster with the biggest typicality. The maximal typicality of the current data sample is defined as $\mu_{pk} = \max_j \mu_{jk}$, where index p defines the cluster with the maximal typicality. If $\mu_{pk} > \mu_{min}$ than the cluster with the index p will be adapted:

$$v_{pk} = v_{p,k-1} + \frac{1}{n_p}\left(z_k - v_{p,k-1}\right),\ F_{pk} = \frac{1}{n_p}C_{pk},\ C_{pk} = \gamma C_{p,k-1} + \left(z_k - v_{p,k-1}\right)^T\left(z_k - v_{pk}\right) \quad (8)$$

where n_p stands for the current number of samples in $p-th$ cluster, v_{pk} and F_{pk} for the center and covariance matrix of $p-th$ cluster.

If the maximal typicality is lower than the threshold value ($\max_j \mu_{jk} < \mu_{min}$), the data sample is put into the buffer. When the data buffer is full (n_{buf}), new clusters are identified by using GKPCM algorithm on the buffer data.

The pseudo-code of the Evolving Gustafson-Kessel Possibilistic c-means clustering algorithm is given in Algorithm 2.

Algorithm 2 Pseudo-code of the eGKPCM clustering algorithm.

Definition of the maximal number of prototypes c_{max}, the fuzziness parameter η, the termination criteria ϵ, the parameter of the cluster shape δ, the parameter of the accepted maximal trace of the cluster covariance matrix t_{max} and the accepted maximal ration between the biggest and the smallest eigenvalue γ_{max}, the minimal typicality of the data sample μ_{min}, the number of required samples in the cluster n_{min}, the number of data samples in the buffer n_{buf} and the forgetting factor is defined as λ.
Detection of the cluster centers from the first n_{buf} data samples using GKPCM and detection of the number of the data samples in each new clusters n_j.
$k \rightarrow n_{buf} + 1$
repeat
  Computation of the absolute typicalities for a new data samples to all cluster.
  if the maximal typicality $\geq$ μ_{min} **then**
    adapt the current cluster centers and the covariance matrices
  else
    put the data sample into the buffer
  end if
  if the number of elements in buffer $\geq$ n_{buf} **then**
    Calculate new cluster centers by GKPCM algorithm and empty the buffer
  end if
  Increment of current sample $k \rightarrow k + 1$
until $k > N$ (the end of the data set)

3.1 An example of eGKPCM clustering algorithm

The evolving algorithm is shown on the data set generated by four processes. The data of these processes form clusters in two dimensional space of different shape, orientation and with the different number of samples. The environment is heavily noisy. At the beginning first two processes are in function and they produce data stream randomly. This means that the data in the stream comes randomly from these two processes and from the noise generator. After sample 300 two more processes start generating the data. The first process (x_{11}, x_{12}) generates $N_1 = 450$, the second process (x_{21}, x_{22}) generates $N_2 = 125$ samples as:

$$x_{11} = \mathcal{N}(3, 0.75),\ x_{12} = x_{11} + \mathcal{N}(3, 0.25),\ x_{21} = \mathcal{N}(-4, 0.50),\ x_{22} = \mathcal{N}(4, 0.25) \quad (9)$$

The third process (x_{31}, x_{32}) generates $N_3 = 75$ and the fourth process (x_{41}, x_{42}) generates $N_4 = 50$ samples as:

$$x_{31} = \mathcal{N}(0, 0.75),\ x_{32} = \mathcal{N}(0, 0.75),\ x_{41} = \mathcal{N}(-4, 0.75),\ x_{42} = -x_{41} + \mathcal{N}(10, 1) \quad (10)$$

The data are corrupted with strong noise which consists of $N_{\mathcal{N}} = 100$ samples and is described as

$$x_N = \mathcal{N}(0, 12.00),\ x_N = \mathcal{N}(0, 18.00) \quad (11)$$

The stream is formed randomly.

The pre-assumed maximal number of clusters is equal to $c_{max} = 2$, the fuzziness parameter is equal $\eta = 1.5$, the maximal allowed trace is defined as $t_{max} = 1.00$, the cluster shape parameter $\delta = 2$, the minimal typicality $\mu_{min} = 0.02$, the forgetting factor $\gamma = 0.996$ and the data buffer is equal to $n_{buf} = 20$. The number of minimal samples in a cluster was set as $n_{min} = 8$. The solution obtained with the GKPCM is shown in Fig. 3 and Fig. 4. In Fig. 3 the results of the initial phase are shown. Figure shows the starting cluster after initialization. The outside

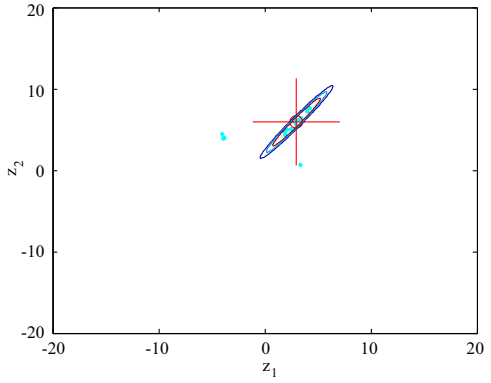

Figure 3: Example of eGKPCM clusters and centers in the initial phase.

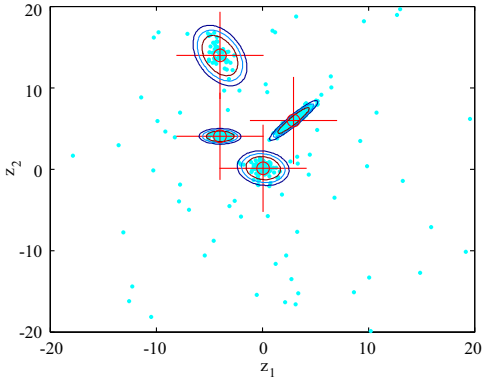

Figure 4: Example of eGKPCM clusters and centers in the final phase.

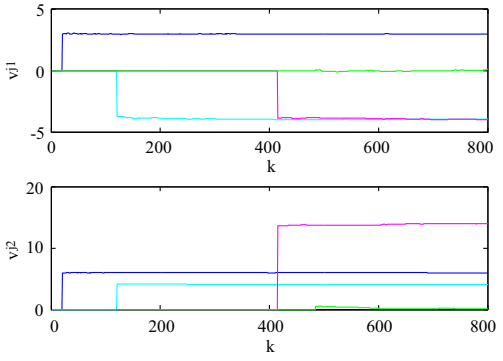

Figure 5: Example of eGKPCM cluster centers evaluation during the learning.

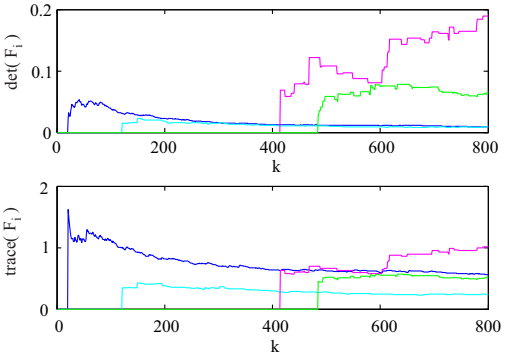

Figure 6: Example of the covariance matrices determinants and traces evaluation during the learning.

contour shows the typicality of 0.01. In Fig. 4 the end result is shown. The adaptation of cluster centers is shown on Fig. 5. The adaptation of the fuzzy covariance matrix is shown on Fig. 6 trough the representation of the determinant and trace. In Fig. 4 the final shape, orientation and position of the clusters is shown and in Fig. 5 the evaluation of cluster centers in both coordinates is presented.

4 Conclusion

In this paper we presented a novel idea for on-line clustering algorithm that is resistant to outliers, noise and can deal with the clusters of different shape, volume, size and can also cope with the big-data problems. The functioning was presented on a generic data example, with a

lot of present noise. The obtained results are very encouraging. Even with a lot of present noise the cluster centers were identified perfectly. We believe that the proposed approach can be very effective for big-data problems and on-line fuzzy model identification. The only disadvantage at the current stage is that algorithm still a few parameters, that should be defined by the user. In further investigation the effects of these parameters on the clustering results will be investigated. We hope to improve degenerative cluster elimination procedure in the sense of making it automatic without user defined thresholds or parameters. The future work will include the testing of the algorithm on real big-data problems and try to see how does the method cope with high dimensionality problems.

References

[1] P. Angelov. *Evolving Takagi-Sugeno Fuzzy Systems from Streaming Data (eTs+).*

[2] P. Angelov, E. Lughofer, and X. Zhou. Evolving fuzzy classifiers using different model architectures. *Fuzzy Sets and Systems*, 159(23):3160 – 3182, 2008.

[3] P. Angelov and X. Zhou. Evolving fuzzy systems from data streams in real-time. In *International Symposium on Evolving Fuzzy Systems 2006*, pages 22–35. IEEE, 2006.

[4] P. P. Angelov and D. P. Filev. An approach to online identification of takagi-sugeno fuzzy models. *IEEE Trans. Syst. Man Cyber. part B*, 34(1):484 – 497, 2004.

[5] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.

[6] S. L. Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intellegent and Fuzzy Systems*, 2:267 – 278, 1994.

[7] D. Dovžan and I. Škrjanc. Recursive clustering based on a gustafson-kessel algorithm. *Evolving Systems*, 2:15 – 24, 2011.

[8] D. Dovžan and I. Škrjanc. Implementation of an evolving fuzzy model (efumo) in a monitoring system for a waste water treatment process. *IEEE Transactions on Fuzzy Systems*, pages 1 – 1, 2014.

[9] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well separated cluster. *Journal of Cybernetics*, 3(3):32 – 57, 1974.

[10] D. Filev and O. Georgieva. *An Extended Version of the Gustafson-Kessel Algorithm for Evolving Data Stream.*

[11] R. Krishnapura and J. M. Keller. Possibilistic approach to clustering. *IEEE Trans. on Fuzzy Systems*, 1(2):98 – 100, 1993.

[12] E. Lughofer. *Evolving Fuzzy Systems - Methodologies, Advanced Concepts and Applications.* Springer, Berlin Heidelberg, 2011.

[13] L. Maciel, F. Gomide, and R. Ballini. Recursive possibilistic fuzzy modeling. In *2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)*, pages 9–16. IEEE, 2014.

[14] B. Ojeda-Magana, R. Ruelas, M. A. Corona-Nakamura, and D. Andina. An improvement to the possibilistic fuzzy c-means clustering algorithm. *Intelligent Automation and Soft Computing*, 20(1):585 – 592, 2006.

[15] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek. A possibilistic fuzzy c-means clustering algorithm. *IEEE Trans. on Fuzzy Systems*, 13(4):517 – 530, 2005.

[16] H. Soleimani-B., C. Lucas, and B. N. Araabi. Recursive gath-geva clustering as a basis for evolving neuro-fuzzy modeling. *Evolving Systems*, 1(1):59 – 71, 2010.

[17] H. Timm, C. Borgelt, C. Doering, and R. Kruse. An extension to possibilistic fuzzy cluster analysis. *Fuzzy Sets and Systems*, 147(1):3 – 16, 2004.