Clustering as a Tool for Self-generation of Intelligent Systems: A Survey

Rashmi Dutta Baruah, Plamen Angelov

Abstract **- Fuzzy Rule Based (FRB) and Neuro-fuzzy systems are commonly used as a basis for intelligent systems due to their transparent and simple human interpretable structure. One of the crucial steps in designing FRB and neuro-fuzzy systems is to innovate the rule base. Data clustering is one of the approaches that have been applied extensively to automatically generate rules from input-output data. The goal of this paper is to critically review some of the most commonly used as well as recently developed clustering techniques, emphasizing their use in rule base generation. The paper explores the shift from offline clustering techniques to online and finally to evolving techniques that originated due to the current demand of adaptive systems.**

I. Introduction

INTELLIGENT systems differ by other technical, social etc. systems by their ability to learn, reason, make decisions. It is common to represent technical and computer-based systems which has some degree of '*computational intelligence*' by either a Fuzzy Rule Based (FRB) or neural-network systems. FRB systems, in particular has gained the attention of researchers and users due to their specific properties, one being their transparent and human interpretable rule-based structure that is expressive enough to represent even imprecise qualitative knowledge. The core components of such a system are: the rule base and an associated inference process. The rules are of the form: IF *antecedent* THEN *consequent,* and the inference process determines the crisp output for a given input using the rule base.

FRB systems can be broadly classified into two families: Mamdani-type and Sugeno-type. In the Mamdani-type [1], also called linguistic systems, rules are represented as:

$$IF\ x_1\ is\ A_1\ and\ x_2\ is\ A_2\ and\ \ldots\ and\ x_n\ is\ A_m\ THEN\ y\ is\ B \quad (1)$$

where x_i, $i = 1,2, ..,n$ is the input variable, A_j, $j = 1,2, ..m$ and B are linguistic terms (eg. *Small, Large, High, Low* etc.) defined by fuzzy sets, and y is the output associated with the given rule.

The rule structure of the second type, Sugeno-type [2], also called TSK type, is usually given as:

$$IF\ x_1\ is\ A_1\ and\ x_2\ is\ A_2\ and\ \ldots\ and\ x_n\ is\ A_m \quad (2)$$

$$THEN\ y\ is\ a_0 + a_1\, x_{1+\ldots+}\ a_n x_n$$

where x_i's, A_j's, y are input variables, linguistic terms, and output variable associated with the rule respectively, and a_0, $a_{1,\ldots,}a_n$ are consequence parameters.

Thus, in Mamdani-type the consequent of each rule is a fuzzy set whereas in Sugeno-type the consequent is a function of input variables. Due to this difference the inference mechanism of determining the output of the system in both the categories varies somewhat.

The early approaches to the design of the rule base involve representing the knowledge and experience of a human expert, associated with a particular system, in terms of IF-THEN rules. To achieve better system performance another alternative is to use expert knowledge as well as learn from system generated input-output data. This fusion of expert knowledge and data can be done in many ways. For example, one way is to combine linguistic rules from human expert and rules learnt by numerical data [3] and another way is to derive rules from expert knowledge and optimize the parameters (e.g. membership function) using input-output data by applying machine learning techniques [4, 5]. However, recently research in generating fuzzy rules only from input-output data has gained momentum in order to avoid the difficult task of knowledge acquisition [6]; moreover due to technological advancements huge amount of data is easily available.

System modelling requires structure identification and parameter identification. Structure identification deals with determining the input variables, number of rules etc. whereas parameter identification deals with antecedent parameters (membership functions of fuzzy sets) and consequence parameters. A clustering algorithm is mainly applied to structure identification (determining rules) by partitioning the data space. While most of the algorithms assume that input variables are available (based on data and prior knowledge or heuristics), others may optimize iteratively the input variables using various variable selection criteria [7-9]. There are numerous approaches for learning fuzzy rules from data such as, grid based [3, 10], neural network and neuro-fuzzy based [4, 11-13], and evolutionary computation based [14-16]. However, clustering techniques, especially fuzzy clustering, are being used extensively either independently or combined with other techniques for rule generation. Methods based on fuzzy clustering are appealing as there is a close connection between fuzzy clusters and fuzzy rules. For a particular system to be modelled, each IF-THEN rule specifies an area exemplified by a point in the graph of control function that can

Both authors are with the Intelligent Systems Research Laboratory, Department of Communication Systems, InfoLab21, Lancaster University, Lancaster, LA1 4WA, UK; phone +44 (1524) 510525; fax: +44 (1524) 510493; e-mail: r.duttabaruah@lancaster.ac.uk

be identified with the Cartesian product of the membership functions modelling the linguistic terms. For example, point *x* (Fig. 1), is a 'typical' point on the control function of the system, the neighbouring points of x are less 'typical' and so have decreasing membership degree with increasing distance. In the same manner, in a fuzzy cluster (defined in Section 2) *x* can be represented by the cluster centre and the membership degree of neighbouring elements in the cluster decrease with increasing distance to the cluster centre [17, 18].

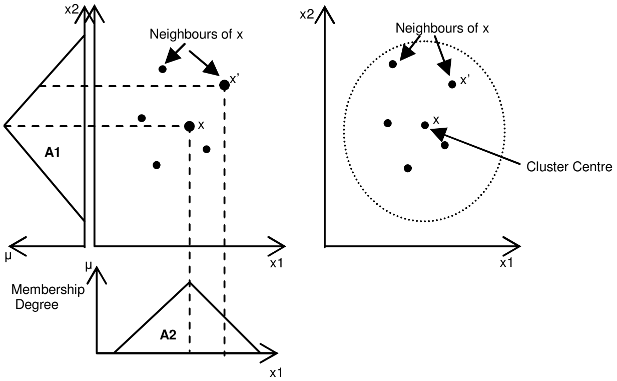

Fig. 1. Correspondence between point x in product space of membership function of input variables and cluster centre.

This paper aims at reviewing clustering techniques and highlighting the pros and cons of each technique in the context of fuzzy rule generation for FRB systems and similarly to neuro-fuzzy systems as examples of *computationally intelligent* systems. The techniques that are considered are categorized here as: i) offline, ii) online, and iii) evolving. The remainder of the paper is organized as follows: section 2 presents an overview of fuzzy clustering and fuzzy rule generation, Section 3, 4 and 5 review the offline, online, and evolving clustering techniques respectively. Finally, section 6 concludes the paper.

II. Fuzzy Clustering and Fuzzy Rule Generation

Given a data set, the aim of clustering is to partition it into different groups (clusters) so that the members in the same group are of similar nature, whereas members of different groups are dissimilar. While clustering, various similarity measures can be considered, one of the most commonly used is distance between data samples. A hard or crisp clustering technique, e.g. k-means [19], assigns a data sample to only one cluster whereas in fuzzy clustering a data sample can belong to all the clusters with certain degree of membership [20].

In order to generate fuzzy rules, fuzzy clustering can be done in input data space only, output data space only or jointly in input-output data space; each of these approaches has its own advantages and disadvantages. After clustering is applied, each cluster induces a rule by projecting the cluster to the respective coordinate space. For rules of Mamdani-type, Sugeno and Yasukawa [21] and Emami et.al. [22] used fuzzy clustering for clustering output data and then projecting the clusters on to the input space in order to define the rule premise (Fig. 2). Babuška and Verbruggen [23] proposed a method to derive a linguistic model from a TSK model. The TSK model is determined by clustering in the input-output space and then the concept of complementary partition is applied to derive the linguistic model. The approach proposed by Salehfar et al. [24] is to apply clustering first to the output space then these clusters are projected on to the input variables. Again clustering is applied to these clusters in the input space to generate several sub-clusters to derive fuzzy rules.

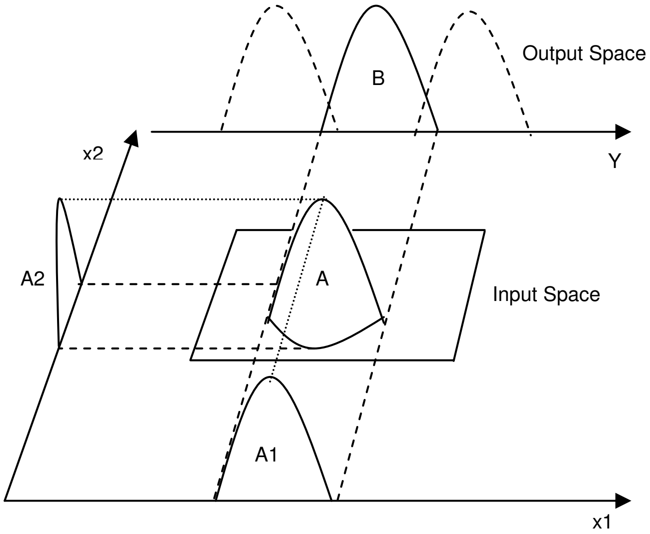

Fig. 2. Fuzzy Clustering in output space and projection on to input space. [21]

In TSK type rules, the common approach is to apply clustering in the input-output data space and projecting the clusters on to the input variables coordinate to determine the premise (membership function) parameter of the rule [25-27] (Fig. 3). The consequent parameters of such rules may be estimated separately by using methods like least squares method.

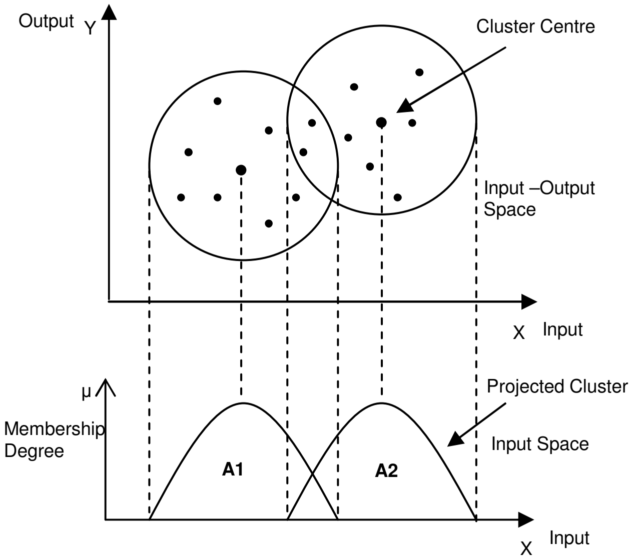

Fig. 3. Fuzzy clusters in input-output space and projection on to input space.

The various clustering techniques used for learning fuzzy rules are categorized here into three categories; i) offline; ii) online, and iii) evolving clustering techniques depending on the mode of feeding data samples to them and the ability of the clustering structure to grow or shrink (to evolve) as opposed to the case when the number of clusters is pre-defined and fixed. Offline methods consider the entire data to be in the memory and perform multiple pass (iterations) over it to get the desired number of clusters (partitions). Such methods are simple and easy to implement compared to other techniques, however they are unable to handle high dimensional data. Further, when used for data partitioning and FRB (or neuro-fuzzy) *computationally intelligent* systems design the resulting rule base is static and the system cannot handle any deviations in the input data which may be due to changes in the operating environment over time. In order to incorporate such changes in the rule-base it is required to re-model the whole system [28]. As the technology has advanced, potentially a huge amount of data with a high data rate can be received from various applications such as packet monitoring in the IP network, real time surveillance systems, and sensor networks. Clustering such form of data, commonly referred as *data stream,* require the algorithms to be fast (non- iterative), memory efficient (need not store previously seen data), adaptive (change the model structure and parameters taking into account data *shift* and *drift*) [29, 30]. Online clustering techniques, as considered here, are algorithms that are incremental or one pass and can handle high dimensional data. However, many online clustering algorithms do still consider the structure of the clusters to be fixed (the number of clusters to be pre-defined) and only change the position of the cluster centres (e.g. Self-organizing maps SOM, Adaptive resonance theory ART [31], etc.). If, in addition to the fact that the data samples are provided in one pass, incrementally, on-line, the cluster structure (number of clusters) can also change (grow or shrink) then we have *evolving* clustering [30]. Thus, evolving clustering, introduced in 2001-2002 [33-36] incorporates the features of online algorithms but in addition have the important property of evolving or adapting the model structure itself, which paves the way for complex systems autonomous structure identification which is a breakthrough in complex (including intelligent) systems design and learning. Later on various applications and developments were reported based on this concept.

III. A Brief Review of the Off-Line Clustering Methods in Relation to FRB Systems Design

Some of the most commonly used clustering techniques for offline FRB system design are: fuzzy c-means, Gustafson-Kessel algorithm, mountain clustering, and subtractive clustering.

Fuzzy c-means (FCM) [37, 38] is adapted from k-means algorithm and is based on minimization of an objective function (equation 3) to obtain optimal number of clusters. A $c\times n$ fuzzy partition matrix U is defined, where n is the number of data points and c, $1<c<n$ is the number of clusters. Each element of U represents the degree of membership (u_{ij}) of a j^{th} data point to i^{th} cluster. The goal is to minimize the squared distance of the data points to their cluster centers given the two conditions (equation 4a and 4b).

$$J(X,U,C)=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^{m}\left\|x_j-c_i\right\|^2, \tag{3}$$

$$\sum_{j=1}^{n}u_{ij}>0,\quad i=1,\dots,c \tag{4a}$$

$$\sum_{i=1}^{c}u_{ij}=1,\quad j=1,\dots,n \tag{4b}$$

X is the data set and C is the set of cluster prototypes (usually cluster centers). The parameter m is called *fuzzifier* exponent, cluster boundaries become softer with higher values of m and vice versa. Before clustering process begins, c, m and U are initialized. A threshold value for terminating condition is also selected. After initialization, the cluster centers (equation 5) and then the partition matrix are updated (equation 6) iteratively. The process continues until the change in the partition matrix at step k and at step $k+1$ is less than a threshold value.

$$c_i=\frac{\sum_{j=1}^{n}u_{ij}^{m}x_j}{\sum_{j=1}^{n}u_{ij}^{m}} \tag{5}$$

$$u_{ij}=\frac{1}{\sum_{k=1}^{c}\left(\frac{\|x_j-c_i\|}{\|x_j-c_k\|}\right)^{\frac{2}{m-1}}} \tag{6}$$

The centre of a cluster is nothing but the mean of all points weighted by their degree of membership to the cluster. The algorithm attempts to move the cluster centers to the proper location within the data set by iteratively updating the cluster centers and the membership degrees. The fuzzy c-means algorithm is simple to implement and has been widely applied in isolation or in combination with other techniques in various domains [39-40]. However, the algorithm is sensitive to initialization of parameters and may get stuck in local minimum [20].

Many variants of fuzzy c-means algorithm have been proposed in the literature by applying various types distance measure other than Euclidean distance. Gustafson-Kessel algorithm (GK) [32] uses Mahalanobis distance as a distance measure between a data point and a cluster centre in order to generate clusters of various size and shapes other than spherical clusters. Each cluster is characterized by cluster centre and covariance matrix (cluster prototype). The distance for i^{th} cluster is given by equation (7), where A_i is the covariance matrix of the cluster.This allows the adaptation of distance norm to the shape of each cluster. Similar to fuzzy c-means algorithm, the cluster centers and the membership grades are updated iteratively in addition to the covariance matrix until change in the partition matrix at step k and at step $k+1$ is less than a threshold value.

$$D_{ij}^2 = (x_j - c_i)^T A_i^{-1} (x_j - c_i) \tag{7}$$

The eigenstructure of the cluster covariance matrix presents the shape and orientation information of the cluster. If the matrix is restricted to diagonal matrix then axis-parallel clusters are generated that are suitable for fuzzy-rule generation [42]. The disadvantage of this algorithm is that it is computationally more intensive compared to fuzzy c-means due to involvement of matrix inverse calculations while updating the covariance matrix. Moreover, it is sensitive to initialization of parameters [20].

Mountain clustering [43, 44] is a simple algorithm that can be used either with fuzzy c-means to generate initial cluster centers or independently to generate approximate cluster centers. Each dimension of the data space is discretized into equidistant points forming a grid. The intersection of the grid lines are called *nodes* and are the potential clusters. A *mountain function* is defined that is related to the density of neighbouring data points and is used to calculate the potential of each grid point (node) to become a cluster centre. The value of the function is high for a node with many neighbouring data points. For all the nodes the mountain function is calculated and the node with highest value is selected as the first cluster centre. To determine the next cluster centre, an amount proportional to the distance of the point to the first cluster centre is subtracted from the current mountain function value of each of the nodes. Thus, the nodes near to the first cluster centre will have higher reduction of their value as compared to the distant nodes. This ensures that nodes closer to the cluster centre are not selected as new cluster centers. Now, the node with the highest remaining mountain function value is chosen as the next cluster centre. This process of selecting cluster centers and subsequently reducing the mountain function value continues until the value of current maximum of the mountain function compared to the first maximum falls below a threshold. The algorithm is simple, however computationally expensive for high dimensional data. Each iteration requires evaluation of $o(n^d)$ nodes where n is the number of grid lines and d is dimension of data space. Further, the generation of number of clusters is sensitive to grid resolution, finer the grid lines more are the potential cluster centers (nodes) i.e. a tradeoff between accuracy and computational complexity. Also, the method needs to predefine certain critical parameters for calculation of mountain function and a threshold value as a terminating criterion.

Subtractive clustering [27, 45] is an improved version of mountain method for cluster estimation. The important difference between the two methods is that, data points are considered as potential clusters instead of grid points in subtractive clustering. This method also assumes that the data points are normalized and bounded by a hypercube. For every data point a *potential* value is calculated and the point with the highest potential value is selected as first cluster centre. The potential value is dependent on the distance of the data point to all other data points, i.e. the larger the number of neighbouring data points the higher is the potential. The neighbourhood of a data point is defined by a constant (radius *r*); data points outside the neighbourhood do not have significant influence on the potential value. Similar to the mountain method, the next step is to reduce the potential of all data points by an amount that is dependent on their distance to the cluster centre. So, the points closer to the cluster centre have less chance to be selected as next cluster centre. Now, the next cluster centre is the point with the remaining maximum potential. Two threshold values are defined that controls the termination of the clustering process. If the ratio of potential (P_k) of the current data point (x) and the potential of the first cluster centre (P_1) is greater than an *upper threshold* value ($P_k / P_1 > uth$) then x is accepted as cluster centre and the process continues. If this ratio is less than a *lower threshold* value ($P_k / P_1 < lth$) then x is rejected and the process terminates. If the ratio lies between the two threshold values then the smallest distance (*minDist*) between the x and existing clusters is determined and the following condition is examined: (i) if sum of the *minDist/r* and P_k / P_1 is greater than or equal to 1 ($minDist/r + P_k/P_1 \geq 1$) then x is set as new cluster centre and the process continues else it is rejected and data point with the next highest potential is selected and tested for above conditions. Although the computational complexity increases linearly with the dimension of the data set, it grows as the square of the number of samples. In most cases it has not been tested on large data sets [46, 47]. Further, this algorithm also needs to predefine certain critical parameters required for potential calculation, neighbourhood definition and threshold values.

IV. A Review on Online Clustering Methods Suitable for Design of FRB Systems

The area of online clustering or data stream clustering itself is an emerging and wide area of research engaging many researchers. The literature provides numerous methods that have been proposed for clustering data streams [48, 49]. In this section a brief review of selected online clustering methods is presented.

Several variations of fuzzy c-means algorithm have been proposed for data streams [50-53]. Both the algorithms streaming fuzzy c means (SFCM) [50] and online fuzzy c means (OFCM) [51] assumes and that the data is arriving and processed in batch i.e. n_1 data points arrive at time T_1, n_2 at time T_2 and so on. At the initial step the number of clusters is predefined. The first batch of data is clustered using FCM and only the weighted cluster centers are stored. The weight of a cluster center is the sum of membership degrees of all data to that cluster. As the next batch of data arrives the cluster centers of previous batch of data are clustered together with this new set of data. The weights are now calculated using the membership values of current data points. Thus, the clustering of each batch of data points is initialized with the centers obtained from the previous clustering (cluster history). The equation for calculation of cluster centre, membership function, and the objective function in FCM are modified to incorporate the weight factor. The OFCM is better compared to SFCM in that the result of SFCM is dependent on the clustering history. The aim of sWFCM (Weighted Fuzzy C-Means for data stream) algorithm [52] is to reduce the memory usage as compared to FCM. The basic approach of sWFCM algorithm is similar to the algorithms described above. However, it uses a different measure of weight. Each data is associated with a time weight factor that represents the data's influence extent on the clustering process. The algorithm iteratively updates the weighted cluster centers till objective function value reaches a required minimum or the number of iterations reaches a threshold value. Although these algorithms take into account the efficient memory usage, they are more suitable for large data sets rather then high speed real time online data due to involvement of iterative computations.

The adjustable fuzzy c-means algorithm [53] considers that the incoming data is available as snapshots (chunks of data) (i.e. spatial, temporal or spatio-temporal segments) and intends to adjust the number of clusters dynamically for each snapshot. The cluster prototypes generated in one data snapshot are successively migrated to the next data snapshot (chunk). These prototypes from previous snapshot are used as starting point for clustering process in the current data snapshot. The adjustment of number of clusters is performed by cluster splitting or merging based on a predefined threshold value. The property of dynamically adjusting the number of clusters depending on the underlying data makes this approach more appealing compared to other modified FCM methods.

A variant of Gustafson-Kessel algorithm for evolving data stream is proposed in [54]. It assumes an initial set of clusters that are obtained by applying GK algorithm offline. For each incoming data point, its distance to all the existing clusters is calculated. If the distance is less than or equal to the radius of the nearest cluster then the data point is assigned to the cluster, where the radius is the distance between the cluster centre and farthest point belonging to the cluster such that its membership degree is greater than or equal to a given membership degree threshold. In this case the cluster centre is updated using Kohonen rule and the inverse covariance matrix and its determinant are updated using Woodbury' matrix inversion lemma and a learning approach [54, 55]. If the distance of the data point is greater than the nearest cluster radius then a new cluster is created considering the data point as its centre. Its covariance matrix is initialized to the covariance matrix of the closest cluster. New clusters are accepted depending on a threshold value. The lower bound of the threshold value depends on the dimension of the data and is estimated from the minimum number of points required to learn the covariance matrix parameters. High value of this threshold parameter, in turn, discards the outliers. Although, some offline processing is required to initiate the algorithm, it does not require data snapshots/chunks later. Further, it is not iterative and automatically detects outliers.

Mean shift algorithm [56], a kernel density estimation based nonparametric technique, is capable of determining clusters with no restrictions on their shape. It uses the *mean shift* procedure [57, 58] to find the point of maximum density. The data points in the *d*-dimensional feature space are handled through their empirical probability density function (pdf) where dense region in the feature space corresponds to local maxima or mode of the distribution. This approach provides the number of clusters (modes of the pdf) automatically, but is iterative. However, if appropriately modified for online clustering, it may be used for FRB system design.

V. Evolving Clustering Methods - a basis for on-line Autonomous Self-Development of FRB Systems

As defined in [30,59] the term *evolving* is not similar to the term *evolutionary* as the former is related to the life-long self-development of an individual entity while the later is concerned towards generation of population of individuals by reproduction, mutation, and natural selection etc. The development of evolving techniques that are adaptive was motivated by the need to design dynamic systems that can continuously change over time by learning from interactions with the environment and self-monitoring.

An online clustering technique for adapting TSK fuzzy rule base is presented in [33, 34]. The method builds upon subtractive clustering and uses proximity-based *potential* value to determine a cluster centre. A Cauchy function (given in equation (8)), which is monotonic and inversely proportional to the distance between all of the data points is used to determine the potential.

$$P_i = \frac{1}{1+\frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{n+1}\left(d_{ij}^{k}\right)^2} \quad , i=1,2,\ldots,N \tag{8}$$

where $d_{ij}^{k} = z_i^{k} - z_j^{k}$, denotes the projection of the distance between the two data points z_i^k and z_j^k on the axis $z^k; k=1,2,\ldots,n+1; Z \in R^{n+1}$ and N is the number of training data samples.

For a given data point, the higher the number of surrounding neighbours the higher its potential value is. At first step, the first data sample of the input stream is established as the first cluster centre with potential set to one. As the next sample arrives its potential is calculated using a recursive form of equation (8). Since potential depends on the distance to all the data points, arrival of a new sample causes the potential of all the cluster centres to change. The potential of the new data point is compared with the potentials of all the existing cluster centres and one of the following actions is performed: (i) the new data is added as a new cluster centre if it has the highest potential compared to all the existing cluster centres; (ii) if the new data point has the highest potential and it is near to a cluster centre then it replaces the later. If both conditions are not satisfied then the data is added to the cluster with closest cluster centre and then next data sample in the stream is considered. The process continues till all the samples in the data stream have been considered. Each of the cluster centres represents a rule antecedent. Thus, with each incoming data, as the cluster centre is updated the rule antecedent automatically gets updated. One of the favourable characteristics of this algorithm is that it automatically handles the outlying data because the potential of such data would be low due to their distance from the normal data. Further, it does not require any user-defined threshold values or parameters like number of clusters etc. that are required in other clustering techniques like subtractive or mountain clustering. However, Cauchy type potential recursive calculation is crucial. The proposed algorithm has been applied for identification of evolving FRB models [60] by incorporating a threshold value. The decision of introducing a new cluster centre (rule) or replacing an existing cluster centre is considered if the potential of a new data sample is higher then the threshold value. The consequent parameters of the rules are determined using recursive least square (RLS) technique.

An evolving neural network (eNN) model, linked to the evolving rule base (eR) model, is introduced in [35]. The structure of this neural network evolves (new neurons are added or old ones are replaced) as the new data arrives and depending on the data sample's *potential*. The proposed eNN has six layers where layer 1 is input layer, layer 2 corresponds to fuzzy set in a TSK rule, layer 3 represents set of rules where each neuron represents a fuzzy rule, layer 4 represents the consequent part of a TSK rule, layer 5 aggregates the antecedent and consequent of each rule, and layer 6 generates the final output of the system. Online clustering is used to determine the cluster centres that represent a neuron in layer 3 of eNN (focal point of a rule). The condition that decides addition of a new neuron or replacement of an old neuron is similar to the one described above [33] except that the potential of a new data point is calculated recursively using a recursive form of equation (9).

$$P_k = e^{-\alpha \sum_{i=1}^{k} \sum_{j=1}^{n+1} \left(z_{kj} - z_{ij} \right)^2} \quad \text{and} \quad \alpha = \frac{4}{r_a^2} \tag{9}$$

where $z^T = [x^T; y]$ denotes augmented data vector and r_a is the radius defining the neighbourhood in a cluster [27].

A simplified approach for learning evolving TS fuzzy models is (Simple_eTS) presented in [61]. The basic approach of determining cluster centres is similar to [33, 34] (as described in paragraph 2, section V) with a simplified measure called *scatter* instead of *potential*. The recursive calculation of scatter is more efficient compared to potential. A method based on *population* (number of data samples assigned to a cluster) is also proposed that can be used to reduce the rule base. A rule/cluster is ignored by setting its firing level to *0* if the value of population is less than *1%* of the total data samples. Thus, the rule base also evolves incrementally as the data arrives.

In all the methods described above the radius of the cluster is not adaptive, rather it is predefined. The eClustering (evolving clustering) technique proposed in [62] alleviates this drawback. It is applied to the generation of evolving extended TSK type system (exTS) from data streams. The extended TSK systems are multi-input multi-output (MIMO) and a combination of both zero and first order TSK type systems. The clustering is applied to partition the input-output joint data space to retrieve the antecedent part of the fuzzy rules. A Gaussian membership function defined in equation (10) is used.

$$u_j^i = e^{-\frac{\left\| x_j - x^{i*} \right\|^2}{2\left(r_j^i \right)^2}} \tag{10}$$

where u_j^i is the membership degree of the jth input xj, j=1,…,N to the ith fuzzy set i=1,…,n; xi* is the focal point of the ith rule antecedent; r_j^i is the spread of the membership function.

From equation (10) it can be observed that two parameters are required to be determined by clustering, the cluster centre (x^{i*}) (focal point of a fuzzy rule) and the radius of cluster (spread of the membership function). The algorithm is an extension of the on-line version of the subtractive clustering used in [33,34] and the basic steps are similar to the methods described above with the following developments. Most importantly, it adapts the cluster radius based on the local spatial density and this influences the fuzzy sets of the antecedent part of a rule when the clusters are projected on to the input variables axes. Two measures, *support* and *age,* are described that can be used to replace a cluster centre (respectively, rule) along with the value of the *potential.* Support of a cluster/rule is the number of data points within the radius of a cluster centre (same as *population*). The rules with very low support can be ignored. Age is the difference between the number of data samples and the average sum of the time indices of the data sample for a given cluster. Thus, the value of age of a cluster is in the range (*0; k*] and it determines whether a cluster is *young* (values close to *0*) or *old* (value close to *k*). If a cluster is *young* it means recent data is included in the cluster. So, new data with high *potential* value can replace *old* clusters.

Another algorithm that can be applied to dynamic clustering of stream of data where the number of clusters is not required to be defined *a priori* is the Evolving Clustering Method (ECM) [36]. At the initial step of ECM, the first input data sample is considered as first cluster with the data itself as the cluster centre and the cluster radius set to zero. A threshold value (*Dth*) is also defined that limits the cluster size i.e. the cluster radius can grow only up to this threshold value. As the next sample arrives, the Euclidean distance (*dist*) between this sample and all other existing cluster centres is determined. Based on the following three conditions either the data sample is included in an existing cluster with or without any update, or a new cluster is created: (i) the cluster with minimum distance (*mindist*) to the sample is selected, if *mindist* is less than the radius of this cluster then the sample is included in the cluster and no updates are required; (ii) for every existing cluster the respective *dist* is added to the radius (let this value be *range*), the cluster with minimum *range* is selected, if range is less than twice *Dth* then the sample belongs to this cluster, the radius of this cluster is updated to *range/2* and the cluster centre is updated by positioning it in the line joining data sample and cluster centre so that now the distance between the new centre and the sample is equal to the new radius value;

(iii) if *range* is greater than twice *Dth* then a new cluster is created with the input data sample as the cluster centre. This process continues till all the data samples in the input stream are processed. ECM has been applied to Dynamic Evolving Neural–Fuzzy Inference System (DENFIS) [36], a TSK neuro-fuzzy inference system that evolves by taking into account the variations in the input data. In DENFIS, as in a typical evolving system fuzzy rules are generated and updated while the system operates. The ECM method is applied to the input data space to determine the fuzzy sets in the antecedent part of the rules. The consequence of the rules is determined by applying RLS estimator. Several variants of ECM are also present like fuzzy ECM [63] and ECM for classification [64]. Although the ECM is a simple method, outlier detection is not an integral part of the clustering process, it requires threshold to be specified and usually generates a large number of clusters (because it is distance-based, not density based) that later needs to be pruned.

In FLEXFIS (Flexible Fuzzy Inference System) [65] an incremental clustering approach is used for partitioning the input-output data space. The data are assumed to be normalized to the unit interval in each dimension of data space forming a hypercube. The clustering technique is a modified version of vector quantization (VQ) that incorporates *vigilance parameter* (from Adaptive Resonance Theory, ART [66]) for update of cluster centres. The vigilance parameter is chosen to be proportional to the diagonal of the *n*-dimensional data space. The number of clusters and the zone of influence (spread of the fuzzy sets) are not required to be predefined and are generated incrementally. One of the distinct features of this algorithm is that it selects the nearest cluster to a new data point by comparing the distance of this new data point to the surface (instead of centre) of all the existing clusters. This feature and the vigilance parameter together avoid *over-clustering*. Initially the number of clusters is set to *0*. As the first data sample arrives, it is set as the new cluster centre. For subsequent data samples one of the following steps is performed: (i) if the current data sample (*x*) is inside the range of influence of any cluster then its distance to the cluster centre of all such clusters is calculated. The cluster centre with minimum distance to *x* is selected and all its components along with the variance are updated by moving it towards the data sample (*x*). (ii) if *x* lies outside the range of influence of all the clusters then its distance to the cluster centre of all the clusters is calculated. The cluster centre with the minimum distance to *x* is selected and if this distance is greater or equal to the vigilance parameter, a new cluster is created with centre as *x*. The process continues till data is available. Although the clustering method used in FLEXFIS do not require certain parameters to be predefined (initial value of cluster numbers, cluster radius), it needs other parameters for old cluster centre shifting and new cluster generation. Moreover, the outliers are not directly addressed by this clustering algorithm.

VI. CONCLUSIONS

One of the phases of intelligent system design based on fuzzy rule base or neuro-fuzzy model is rule generation. At present the most preferred way is to generate the rules automatically from the input-output data using data clustering. The paper has provided an overview of around fifteen algorithms in the context of fuzzy rule generation and under the category of offline, online, and evolving clustering techniques. Most of the offline techniques are iterative and applies the clustering process over all the training data. Therefore, such techniques are not suitable for design of intelligent systems that continuously accept data from various sources and need to improve in terms of performance over a period of time. Using online techniques in such scenarios is intuitive as they are efficient compared to offline techniques in terms of computation and memory usage. However, there remains a huge scope for improvement of such techniques to conform them for adaptive intelligent system design. An attempt to meet such requirements is evolving techniques that have been applied to design adaptive FRB or Neuro-fuzzy systems. Such clustering methods can still be improved, such as by reducing the number of user-defined parameters, incorporating mechanisms to detect outlying data so that irrelevant clusters are not formed. Further, in most cases such techniques have been applied in scenarios where output data corresponding to an input data is available. The design of adaptive intelligent systems using evolving techniques in fully unsupervised scenarios is still an open issue.

REFERENCES

[1] E.H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic systems", *Fuzzy Sets and Systems*, vol. 26, pp. 1182-1191, 1997.

[2] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modelling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, No. 1, pp. 116-132, 1985.

[3] Li-Xin Wang and J.M. Memdel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 22, No.6, pp. 1414-1427, 1992.

[4] J.-S.R. Jang, "ANFIS: Adptive-network-based fuzzy inference systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, No. 3, pp. 665-685, 1993.

[5] J.-S.R. Jang and C.-T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Transactions on Neural Networks*, vol. 4, No. 1, pp. 156-159, 1993.

[6] K.L. McGraw and K.H. Harbinson-Briggs, "Knowledge Acquisition: Principles and Guidelines," New Jersey: Prentice Hall, Englewood Cliffs, 1989.

[7] T.-P. Hong and J.-B Chen, "Finding relevant attributes and membership functions," *Fuzzy Sets and systems*, vol. 103, pp. 389-404, 1999.

[8] N.R. Pal, "Soft Computing for feature analysis," *Fuzzy Sets and Systems*, vol. 103, pp. 201-221, 1999.

[9] Y. Lin and G.A. Cunningham, "A fuzzy approach to input variable identification," *in Proceedings of IEEE Conference on Fuzzy Systems* 1994, pp. 2031-2036.

[10] H. Ishibuchi, K. Nozaki, H. Tanaka, Y. Hosaka, and M. Matsuda, "Empirical study on learning in fuzzy systems by rice test analysis," *Fuzzy Set and Systems*, vol. 64, pp. 129-144, 1994.

[11] S. Mitra and Y. Hayashi, "Neuro-Fuzzy generation: Survey in Soft Computing Framework," *IEEE Transactions on Neural Networks*, vol. 11, No. 3, pp. 748-768, 2000.

[12] J. Moody and C. Darken, "Fast learning in networks of locally-tuned process units," *Neural Computing*, vol. 1, pp. 281-294, 1989.

[13] K.B. Cho and B.H. Wang, "Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction," *Fuzzy Sets and Systems*, vol. 83, pp. 325-339, 1996.

[14] P. Angelov, R. Buswell, "Automatic generation of fuzzy rule-based models from data by genetic algorithms," *Information Sciences—Informatics and Computer Science*, vol. 150, Issue1-2, pp. 17-31, 2003.

[15] O. Cordon, F. Gomide et al., "Ten years of genetic fuzzy systems: Current framework and new trends," *Fuzzy Sets and Systems*, vol. 141, No. 1, pp. 5-31, 2004.

[16] C.-K. Chiang, H-Y. Chung, and J.-J Lin, "A self-learning fuzzy logic controller using genetic algorithms with reinforcements," *IEEE Transactions on Fuzzy Systems*, vol. 5, No. 3, pp. 460-467, 1997.

[17] F. Klawonn, "Fuzzy Sets and Vague Environments", *Fuzzy Sets and systems*, vol. 66, pp 207-221, 1994.

[18] R. Kruse, J. Gebhardt, F. Klawonn, "Foundations of Fuzzy Systems," Wiley, Chichester, 1994.

[19] T. Hastie, R. Tibshirani, J. Friedman, "The Elements of Statistical Learning Data mining, Inference, and Prediction," Second Edition, Springer, 2009, ISBN: 978-0-387-84857-0.

[20] J.V. de Oliveira and W. Pedrycz (eds.), "Advances in Fuzzy Clustering and its Applications," Wiley, 2007, ISBN: 978-0-470-02760-8 (HB).

[21] M. Sugeno and T. Yasukawa, "A fuzzy-logic based approach to qualitative modelling," *IEEE Transactions on Fuzzy Systems*, vol. 1, pp. 7-31, 1993.

[22] M.R. Emami, I.B. Tŭrksen, and A.A. Goldenberg, "Development of a systematic methodology of fuzzy logic modeling," *IEEE Transactions on Fuzzy Systems*, vol. 6, pp. 346-361, 1998.

[23] R. Babuška and H.B. Verbruggen, "A new identification method for linguistic fuzzy models," *in Proceedings of IEEE International Joint Conference on Fuzzy Systems and Fuzzy Engineering Symposium*, Yokohama, Japan, 20-24 March, 1995, Vol. 2, pp. 905-912.

[24] H. Salehfar, N. Bengiamin, and J. Huang, "A systematic approach to linguistic fuzzy modeling based on input-output data," *in Proceedings of the 2000 Winter Simulation Conference*, Orlando, FL, USA, 10-13 Dec, 2000, Vol. 1, pp. 480-486.

[25] R. Babuška and H.B. Verbruggen, "An overview of fuzzy modelling for control," *Control Engineering Practice*, vol. 4, No. 11, 1593-1606-1996.

[26] J. Zhao, V. Wertz and R. Gorez, "A fuzzy clustering method for the identification of fuzzy models for dynamical systems," *in Proceedings of 9th IEEE International Symposium on Intelligent Control*, Columbus, Ohio, USA, 1994.

[27] S.L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent Fuzzy Systems*, vol. 2, pp. 267-278, 1994.

[28] P. Angelov, "Evolving rule-based models- A tool for design of flexible adaptive systems," Physica-Verlag, 2002, ISBN: 3-7908-1457-1.

[29] P. Angelov and A. Kordon, "Adaptive inferential sensors based on evolving fuzzy models," *IEEE Transactions on Systems, Man, and Cybernetics –B*, pp. 1-11, 2009.

[30] P. Angelov, X. Zhou, "Evolving fuzzy-rule-based classifiers from data streams," *IEEE Transactions on Fuzzy Systems*, vol. 16, No. 6, pp. 1462-1475, 2008.

[31] G.A. Carpenter and S. Grossberg, "Adaptive Resonance Theory, Adaptive resonance theory (ART), in The Handbook of Brain Theory and Neural Networks," M.A. Arbib, Ed. Cambridge, MA: MIT Press, 1995, pp. 87-90.

[32] D. Gustafsson and W. Kessel, "Fuzzy Clustering with a Fuzzy Covariance Matrix", *in Proceedings of the IEEE CDC, San Diego, California*, IEEE Press, Piscataway, New Jersey, 1979, pp.761-766.

[33] P. Angelov, R. Buswell, "Evolving Rule-based Models: A Tool for Intelligent Adaptation," *9th IFSA World Congress*, Vancouver, BC, Canada, 25-28 July 2001, pp.1062-1067.

[34] P. Angelov, An Approach for Rule-base Adaptation using On-line Clustering, *2nd EUNITE Conference*, 19-22 September 2002, Albufeira, Portugal, pp. 47-52.

[35] P. Angelov, D. Filev, "Flexible Models with Evolving Structure," *IEEE Symposium on Intelligent Systems*, Varna, Bulgaria, 10-12 September 2002, v.2, pp.28-3.

[36] N. Kasabov and Q. Song, "DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Transactions on Fuzzy Systems,* 2002, vol. 10, pp. 144-154.

[37] J.C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact, well –separated clusters," *Journal of Cybernetics*, vol. 3, pp. 32-57, 1974.

[38] J. Bezdek, "Pattern Recognition with fuzzy objective function algorithms," Plenum Press, New York, 1981.

[39] H.S. Hwang, K.B. Woo, "Linguistic fuzzy model identification", *IEEE proceedings-control theory and applications*, Vol. 142, Issue 6, pp. 537-544, No. 1995.

[40] Y. Yingchun, Z. Laibin, L. Wei, W. Zhaohui, "Fuzzy C-means algorithm in work consdition recognition of oil pipeline", *4th IEEE conference on Industrial Electronics and Appliciations*, 25-27 May 2009, pp. 682-686.

[41] Hyong-Euk Lee, Kwang-Hyun Park, Z.Z. Bien, "Iterative fuzzy clustering algorithm with supervision to construct probabilistic fuzzy rule base from numerical data", *IEEE Transactions on Fuzzy Systems*, Vol. 16, Issue 1, pp. 263-277, Feb. 2008.

[42] F. Höppner, F. Klawonn, R. Kruse and T. Runkler, "Fuzzy Cluster Anlysis", J. Wiley & Sons, Chichester, United Kingdom, 1999.

[43] R. R. Yager and D. P. Filev, "Approximate cluster via the mountain method", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 24, No. 8, pp. 1279-1284, 1994.

[44] R.R. Yager and D.P. Filev, "Learning of fuzzy rules by mountain clustering", *in Proceedings of SPIE Conference on Applications of Fuzzy Logic Technology*, Boston, MA, pp. 246-254, 1993.

[45] S.L. Chiu, "Extracting fuzzy rules from data for function approximation and pattern classification", Chapter 9 in Fuzzy Information Engineering: A guided Tour of Applications, D. Dubois, H. Prade, and R. Yager (Ed.), John Wiley & Sons, 1997.

[46] S. Chopra, R. Mitra, V. Kumar, "Identification of rules using subtractive clustering with application to fuzzy controllers", *in Proceedings of the 3rd International Conference on Machine Learning and Cybernetics*, Shanghai, 26-29 Aug 2004, pp. 4125-4130, 2004.

[47] K. Demirti and M. Khoshnejad, "Autonomous parallel parking of a car-like mobile robot by a neuro-fuzzy sensor-based controller", *Fuzzy Sets and Systems*, Vol. 160, pp. 2876-2891, 2009.

[48] M.M. Gaber, A. Zaslavsky, S. Krishnaswamy, "Mining data streams: A Review", *ACM SIGMOD Record*, Vol. 34, Issue, pp. 18-26, 2005.

[49] Charu C. Aggarwal (ed.), Data Streams: Models and algorithms, Springer, 2007.

[50] P. Hore, L.O. Hall and D.B. Goldgof, "A fuzzy c means variant for clustering evolving data streams", *IEEE International Conference on Systems, Man and Cybernetics, Montreal*, October 2007, pp. 360-365.

[51] P. Hore, L.O. Hall, D.B. Goldgof, W. Cheng, "Online fuzzy c means", *Annual Meeting of the North American Fuzzy Information Processing Society*, 19-22 May 2008, pp. 1-5.

[52] R. Wan, X. Yan and X. Su, "A weighted fuzzy clustering algorithm for data stream", *2008 ISECS International Colloquium on Computing, Communication, Control and Management*, 3-4 August 2008, Vol. 1, pp. 360 – 364, 2008.

[53] W. Pedrycz, "A dynamic data granulation through adjustable fuzzy clustering", *Pattern Recognition Letters*, Elsevier, Vol. 29, pp. 2059-2066, 2008.

[54] O. Georgieva and D. Filev, "Gustafson-Kessel algorithm for evolving data stream clustering", *in Proceedings of International Conference on Computer Systems and Technologies 2009*, pp. IIIB.14-1-6.

[55] T. Kohonen, "Self Organization and associative memory", 3rd Edition, Springer Information Sciences Series, 1989, ISBN: 0-387-51387-6.

[56] D. Comaniciu and P. Meer, "Mean Shift: A robust approach toward feature space analysis", *IEEE Transactions on Pattern Analysis and machine Intelligence*, Vol. 24, No. 5, pp. 603-619, 2002.

[57] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition", *IEEE Transactions on Information Theory,* Vol. 21(1), pp. 32–40, 1975.

[58] Y. Cheng, "Mean shift, mode seeking, and clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17(8), pp.790–799, 1995.

[59] N. Kasabov, "Evolving Connectionist Systems Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines", Springer-Verlag London, 2003, ISBN: 85233-400-2.

[60] P. Angelov and R. Buswell, "Identification of Evolving Fuzzy Rule-Based Models", *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 5, pp. 667-677, 2002.

[61] P. Angelov and D. Filev, "Simple_eTS: A simplified method for learning evolving Takagi-Sugeno fuzzy models", *in Proceedings of*

2005 IEEE International Conference on Fuzzy Systems, Reno, NE, USA, 2005, pp. 1068-1073.

[62] P.Angelov and X. Zhou, "Evolving fuzzy systems from data streams in real-time", *2006 International Symposium on Evolving Fuzzy Systems*, September, 2006, pp. 29-35.

[63] V. Ravi, E.R. Srinivas and N.K. Kasabov, "On-line evolving fuzzy clustering", *in Proceedings of International conference in Computer Intelligence Multimedia Applications*, 13-15 Dec. 2007, Vol. 1, pp. 347-351.

[64] Q. Song and N. Kasabov, "Weighted data normalization and feature selection for evolving connectionist systems", *in Proceedings of 8th Australia New Zealand Intelligent Inf. Syst. Conf.*, Sydney, NSW, Australia, Dec. 2003, pp. 285-290.

[65] E.D. Lughofer, "FLEXFIS: A robust incremental learning approach for evolving Takagi-Sugeno fuzzy models", *IEEE Transactions on Fuzzy Systems*, Vo. 16, No. 6, pp. 1393-1410, 2008.

[66] G.A. Carpenter and S. Grossberg, "Adaptive resonance theory (ART), in The Handbook of Brain Theory and Neural Networks", M.A. Arbib, Ed. Cambridge, MA: MIT Press, 1995, pp. 79-82.