Evolving Local Means Method for Clustering of Streaming Data

Rashmi Dutta Baruah
School of Computing and Communications
Lancaster University
Lancaster, LA1 4WA, UK
r.duttabaruah@lancaster.ac.uk

Plamen Angelov
School of Computing and Communications
Lancaster University
LA1 4WA, UK
p.angelov@lancaster.ac.uk

***Abstract*—A new on-line evolving clustering approach for streaming data is proposed in this paper. The approach is based on the concept that local mean of samples within a region has the highest density and the gradient of the density points towards the local mean. The algorithm merely requires recursive calculation of local mean and variance, due to which it easily meets the memory and time constraints for data stream processing. The experimental results using synthetic and benchmark datasets show that the proposed approach attains results at par with offline approach and is comparable to popular density-based mean-shift clustering yet it is significantly more efficient being one-pass and non-iterative.**

Keywords-; online clustering, sequential clustering, evolving clustering, data streams

I. INTRODUCTION

Clustering algorithms play an important role in learning models from data. In the context of learning fuzzy rule-base or neuro-fuzzy model, clustering is one of the approaches that is applied extensively to automatically generate rules from input-output data. In the present scenario, the huge amount of data with high data rate received from various applications such as packet monitoring in the IP network, real time surveillance systems, and sensor networks, commonly referred as *data stream,* calls for adaptive or evolving model capable of learning continuously without requiring explicitly the past data Learning such models from data streams in turn require the algorithm to be fast and recursive (non- iterative or one-pass), incremental (training examples presented and learnt one at a time), memory efficient (need not store previously seen training examples). An algorithm possessing all these attributes is often referred as *on-line* algorithm. In addition to these requirements, other essential requirements include; i) handling outliers; ii) evolving (adapting the structure- create new clusters if needed and merge existing ones); iii) not to presume the number of clusters which are usually defined by the user. An algorithm that possesses these additional requirements and is on-line is called evolving [1-4].

In terms of statistical theory, the two groups of approaches to clustering are parametric and nonparametric [5]. In parametric approaches, a clustering criterion is defined assuming an underlying distribution of the data and attempt is made to find the parameter values for this distribution. A typical example is Gaussian Mixture Model (GMM). In contrast to parametric approach, the nonparametric approach, neither considers clustering criteria nor assumes any mathematical structure for the distribution of data. In nonparametric approach, clustering can be formulated as the problem of estimation of means or modes of a mixed probability density distribution. The regions with high local density in the data space can correspond to clusters and these dense regions are nothing but the modes of the underlying unknown probability density function (PDF).

There are several offline clustering algorithms that are based on mode detection. One of the widely used offline gradient-based mode detection approaches is mean-shift [6-7]. It uses the concept of nonparametric density estimation, commonly known as kernel density estimation (KDE) to estimate the gradient of a PDF using the neighbouring points within a small region around that point. So far, mean-shift has been widely used in the areas like image segmentation. Since the method automatically detects the modes that are equivalent to cluster centres and requires just one user-defined parameter (bandwidth/radius), it is quite appealing to the area of fuzzy model identification also. However, the main drawback of such algorithms is that they are iterative and require all the data samples to be present in the memory, thus they are not suitable for online applications for example fuzzy model identification from data streams. Further, it is difficult to define the neighbourhood of a data point when data stream is considered because past samples are discarded.

In this paper we propose a novel online evolving clustering approach, named as *Evolving Local Means* (ELM) clustering. Despite being simple and having the desirable features of density based approaches, it is applicable to data streams. It uses the concept of non-parametric gradient estimate of a density function using *local mean* [5]. During the clustering process the local mean is updated as the samples from the data stream arrive. It adds new clusters when the density pattern changes and thus we use the term *evolving.*

The rest of the paper is organized as follows: a brief overview of related work is presented in section II, section III describes the evolving local means clustering algorithm,

section IV discusses the experimental results, and section V concludes the paper with directions to future work.

II. Related Work

Though the literature provides a plethora of clustering approaches, we are limiting our discussion here to those approaches that are based on mode detection via density estimation. The conventional clustering approaches based on density estimation mostly uses Parzen windows [8]. Due to the demand for online algorithms in real-time applications the focus has shifted from conventional offline clustering to online clustering. One of the popular offline clustering approach is mean-shift which has been applied extensively in image processing applications. Fukunaga and Hostetler [7] presented a gradient estimation approach to clustering and its application to pattern recognition. Like KDE (Parzen Windows), the gradient of a probability density function is estimated using the data samples bounded within a small region and a general form of kernel gradient density estimate is presented. A *mean-shift* class of estimate is developed based on the fact that the mean value of samples within a small region is closely related to the density gradient. They also showed how a gradient ascent clustering can be achieved using the mean-shift class of estimates. In [9] the idea of mean-shift for mode-seeking is theoretically analysed and in [6] mean-shift clustering is applied to feature space analysis. Touzani and Postaire [10] identified that since these methods are gradient based and use differential operators they tend to generate higher number of modes than the actual PDF in noisy situations. They proposed a mode detection algorithm where modes or high density regions are detected by thresholding the PDF at a required level. Samples with estimated value of PDF higher than the threshold value are labeled as a mode. Similarly, samples with PDF value below the threshold are assigned a "valley" label. The decision of labeling a sample as mode or valley is made by considering its spatial relationships with neighbouring points. Specifically, the clustering approach or identification of mode/valley is based on relaxation scheme where the labels of samples are iteratively updated according to a compatibility measure defined among the neighbouring labels. It is shown experimentally that the clustering approach works well in noisy data and small sample size. Another approach that first discretizes the data space and then apply the iterative thresholding method is discussed in [11]. The apparent problems with these approaches are that they are computationally expensive due to the iterative nature. These methods assume all the data is available i.e. require all the samples to be present in the memory before processing, thus are costly in terms of storage. Also good results are not achieved if the sample size is small.

In [12], a sequential method to approximate a multimodal density function with a mixture of Gaussians is presented where the density is represented as sum of weighted Gaussians. The parameters, number of Gaussians, weights, means, and covariances are determined automatically. Mean-shift procedure is applied to detect the modes. Each mode corresponds to a Gaussian component and the mode itself constitutes the mean of the Gaussians. The weight of each Gaussian is determined by adding the kernel weights of the data points that converged to the mode. The covariance matrix associated with each Gaussian is determined using Hessian matrix by fitting the curvature around the mode location. The density function is updated at each time the new data arrives and again mean-shift is applied to detect the modes. Due to its online nature the method is suitable for real-time computer vision applications. Though this approach provides good models if the modes of distribution are Gaussian and well separated, it fails when the distribution is non-Gaussian for example in skewed distributions. Literature provides several other methods that are based on online estimation of GMM. Some methods assume the data to be available as block of data [13] while others need some parameters to be specified a priori [14-15].

III. Evolving Local Mean Clusterng Approach

One simple approach to density based clustering is to assign each sample to the nearest mode along the direction of the gradient at the sample, where each mode represents a cluster centre. An iterative gradient based algorithm like mean-shift shifts each sample by an amount proportional to the gradient at the sample until convergence. It uses the simple concept that *local-mean* can be used as an estimate of gradient of a density function at a point. The gradient of the density is, therefore, pointing towards the local mean. Since ELM clustering adopts the idea of local-mean and gradient from mean-shift algorithm, we first discuss briefly the mean-shift clustering in section A and then describe our approach in section B.

A. Brief Overview of Mean-shift Clustering Algorithm

In [7] the mean-shift algorithm is given, which is based on gradient clustering algorithm. After each iteration, the algorithm shifts each point closer to the nearest mean and finally converges to the nearest mode or cluster centre. The estimation of the density gradient is obtained by the gradient of the kernel density estimate using, for example, Epanechnikov kernel.

Given N independent and identically distributed random data points $\mathbf{x}_i$, $i = 1, .., N$ in an n dimensional space R^n with an unknown density p, the multivariate kernel density estimator $\hat{p}(x)$ at $\mathbf{x}$ is given as ,

$$\hat{p}(x) = \frac{1}{Nr^n} \sum_{i=1}^{N} K\left(\frac{x - x_i}{r} \right) \quad (1)$$

where $K(\cdot)$ is the kernel function that is symmetric but not necessarily positive and integrates to one, and $r > 0$ is the radius or bandwidth.

The density gradient estimate can be given as [7],

$$\hat{\nabla} p(x) \equiv \nabla \hat{p}(x) = \frac{1}{Nr^n} \sum_{i=1}^{N} \nabla K\left(\frac{x - x_i}{r} \right) \quad (2)$$

The density gradient estimate using Epanechnikov kernel (3) can be given as (4),

$$K(u)=\begin{cases}\frac{1}{2}c_n^{-1}(n+2)(1-\|u\|^2), \text{ if } \|u\|^2<1\\ 0, \text{ otherwise}\end{cases} \quad (3)$$

where c_n is the volume of the unit n-dimensional sphere. Now, using Epanechnikov kernel,

$$\nabla K\left(\frac{x-x_i}{r}\right)=\nabla\left[\frac{1}{2c_n}(n+2)\left(1-\left\|\frac{x-x_i}{r}\right\|^2\right)\right] \quad (4)$$
$$=\frac{1}{2c_n}\frac{(n+2)}{r^2}\left[-2(x-x_i)\right]$$

$$\therefore \quad \hat{\nabla}p(x)=\frac{1}{N2(r^n c_n)}\frac{(n+2)}{r^2}\sum_{i=1}^{N}\left[-2(x-x_i)\right]$$
$$=\frac{1}{N(r^n c_n)}\frac{(n+2)}{r^2}\sum_{i=1}^{N}(x_i-x)$$
$$=\frac{N_x}{N(r^n c_n)}\frac{(n+2)}{r^2}\left(\frac{1}{N_x}\sum_{x_i\in S_r(x)}(x_i-x)\right) \quad (5)$$

where the region $S_r(\mathbf{x})$ is a hyper-sphere of radius r having the volume $r^n c_n$, centred at $\mathbf{x}$, and containing N_x data points. The last term in (5) is known as *sample mean shift* ($M(\mathbf{x})$) [7].

$$M(x)=\frac{1}{N_x}\sum_{x_i\in S_r(x)}(x_i-x)=\mu-x \quad (6)$$

where μ is the local-mean i.e. the mean of samples in the region $S_r(\mathbf{x})$.

Also, the constant term in (5) $\frac{N_x}{N(r^n c_n)}$ is the probability density estimate using a uniform kernel over the region $S_r(\mathbf{x})$, thus we can write,

$$\hat{p}(x)=\frac{N_x}{N(r^n c_n)} \quad (7)$$

$$\nabla\hat{p}(x)=\hat{p}(x)\frac{(n+2)}{r^2}M(x) \quad (8)$$

Each sample is moved towards the mode using gradient ascent (9) with normalized gradient. The normalized gradient allows data points far from the mode (local maximum) to move faster with larger step size, and smaller step size near the mode. This is because the density $\hat{p}(x)$ at the points far from the mode (or near local minimum) would be small.

$$\mathbf{x}_j^{k+1}=\mathbf{x}_j^k+c\frac{\nabla\hat{p}(x)}{\hat{p}(x)} \quad (9)$$

where $\mathbf{x}_j^k = j^{th}$ data sample at k^{th} iteration.

Substituting the constant $c=\frac{r^2}{n+2}$, we get

$$x_j^{k+1}=x_j^k+M(x_j^k)=x_j^k+\mu-x_j^k=\mu \quad (10)$$

Thus, equation (10) shows that each sample is shifted with a value equal to the local mean.

B. ELM Clustering Algorithm

The mean-shift algorithm (10) transforms each data sample to the mean of the data samples within the region S_r around it. Considering that the entire data set is divided into convex subsets that are greater than r distance apart, the data samples will always remain within their respective sets or clusters and cannot diverge. This is due to the fact that equation (10) is always a convex combination of members from the same convex set. Also, as soon as all the observations in such a set lie within a distance r from one another, the next iteration will transform them all to a common point, their sample mean [7]. ELM approach is developed based on this feature of the mean-shift algorithm. The issue with the online approach is that since past samples are required to be discarded so it is not possible to identify the neighbourhood of a sample. Therefore, we use heuristics to determine the neighbourhood of a sample and to decide to which local mean (cluster centre) the sample should be associated to. In ELM, a cluster is represented with two parameters: cluster centre, denoted by $\boldsymbol{\mu}_i$, is the local mean, and a distance parameter, denoted by σ_i, is the average norm in the i^{th} cluster. The algorithm can learn model either from scratch or with already existing clusters, provided each cluster is represented with the two parameters, $\boldsymbol{\mu}_i$ and σ_i. Each sample is considered to be bounded by a region of radius r (similar to a kernel). As a sample $\mathbf{x}$ arrives, its distance to all the existing cluster centres is computed. Let us denote the distance from x to i^{th} cluster centre $\boldsymbol{\mu}_i$ by d_i. If x satisfies *condition 1*, it means the region around $\mathbf{x}$ and the region around $\boldsymbol{\mu}_i$ overlaps (Fig. 1), then sample $\mathbf{x}$ is assigned to cluster i.

$$\textit{Condition 1: } d_i < (max(\sigma_i, r)+r) \quad (11)$$

The parameters $\boldsymbol{\mu}_i$ and σ_i are updated recursively after the assignment. If the region around x overlaps with more than one cluster then the nearest one is considered. Once the parameters $\boldsymbol{\mu}_i$ and σ_i are updated it is checked if there is any further overlapping with existing clusters in such a case clusters are merged. On the other hand, if the region around x does not overlap with any existing clusters i.e. if condition 1 is not satisfied then x is declared as a new cluster centre. Algorithm 1 in Appendix summarizes the ELM clustering approach.

The idea is that in a convex region with unimodal data distribution, the mean of all the samples in that region can be considered as the point with highest density and thus the mode. If we apply mean shift approach in such a convex region, all the samples would converge to the sample mean. In ELM clustering approach we avoid the intermediate steps of moving a sample towards the mode in steps proportional to the gradient. Consider a convex region (or cluster) i represented by $\boldsymbol{\mu}_i$ and σ_i. Since in online approach past samples are

discarded, the local mean represents the samples seen so far. When a sample **x** arrives, and if it satisfies the *condition 1* then it means the neighbourhood of **x** contains samples that are in region around $\boldsymbol{\mu}_i$ (Fig. 1). In mean-shift algorithm, **x** will be shifted to the mean of the samples in region, and with successive iteration **x** will finally converge to $\boldsymbol{\mu}_i$. (Fig. 2) because region around **x** is part of the convex region and $\boldsymbol{\mu}_i$ is the mode of that region. In ELM clustering we directly assign **x** to i^{th} cluster since the distance between region around **x** and around $\boldsymbol{\mu}_i$ is less than r. Whenever a sample is newly assigned to a cluster its density changes thus $\boldsymbol{\mu}_i$ and σ_i are updated. Thus, we use the term evolving local mean for the local mean. When region around x overlaps with more than one cluster, we consider that **x** will move towards the cluster with largest overlap using a simple heuristic that larger region will contain more number of samples and thus density will be high.

IV. Experimental Results

In order to evaluate our approach and compare it with classical mean-shift approach [7] we conducted various experiments with both synthetic and benchmark datasets [16]. The algorithms, both ELM and mean-shift, were developed using MATLAB 7.1 and performance was evaluated on a PC with processor speed 2.66 GHz and 2.0 GB RAM. The data is considered as pseudo data streams and processing is done on a per-sample basis in case of ELM. When considering data streams, the sequence in which samples are received often affects the model. So, ELM is tested on 10 different random variations of the same data set and the average of the results is presented here. This is equivalent to 10 different data sequences when processing is done on a per-sample basis. A total of 6 data sets are considered for testing, three synthetic data sets, two benchmark datasets (Table I), and an image data. The aim of using the simple synthetic data sets is mainly to analyze the cluster centre positions of the two clustering algorithms. For evaluation and comparison we have considered the following parameters: average cluster purity, average distance between the cluster centers (modes), and execution time. The cluster purity parameter measures the quality of the clusters using the class information and is given as:

$$purity = \frac{\sum_{i=1}^{C} \frac{N_i^d}{N_i}}{C} \times 100\%$$

where C is the number of clusters, N_i^d is the number of samples in cluster i with the dominant class label, and N_i denotes the total number of samples in cluster i.

To measure the closeness of the cluster centers identified by ELM and mean-shift approach, we use the average distance between the centers. With a fixed radius, it is not certain that both mean-shift and ELM would generate the same number of clusters. Suppose, ELM has generated M number of clusters and mean-shift has N number of clusters, and $M>N$ then while measuring the average distance, we take into account only N number of clusters. For a typical data sequence of the synthetic dataset, Fig. 2 (a), (c), (d) show the clusters obtained

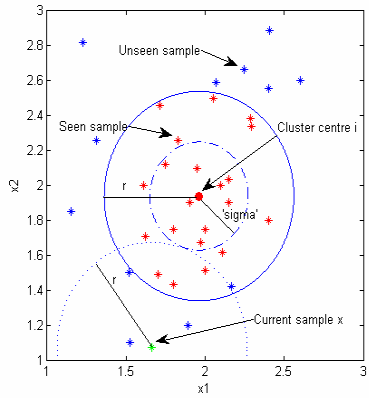


Figure 1. ELM over hypothetical data

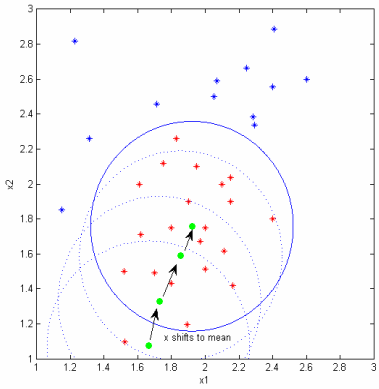


Figure 2. Mean-shift over hypothetical data

by ELM and Fig. 2 (b), (e), (f) show the clusters obtained by mean-shift. Fig. 3 compares the cluster centers identified by ELM and mean-shift. Fig. 4 gives the KDE using Gaussian kernels and shows the cluster centres obtained by ELM over the same plot. Table II gives a comparison of various evaluation parameters for the synthetic dataset. While applying the algorithms on the various data sequences a fixed radius has been used. For DS1, Ds2, a radius of 0.25 was chosen and for DS 3, a radius of 0.15 was used. As different data sequences give different number of clusters, the average and the mode of the number of clusters generated are given in column II of Table II. It is apparent from the Fig. 2, Fig.3, Fig. 4, and the results shown in the Table II that for DS1 and DS2

TABLE I. Data Set Description

Data Set	#Features	#Samples	#Classes #Cluster
DS1 (normal distribution)	2	350	3
DS2 (uniform distribution)	2	400	3
DS3 (non-convex data set)	2	3000	2
Iris	4+1 label	150	3
Wine	13+1 label	178	3

both the algorithms give similar clusters with negligible difference in the coordinates of the cluster centres. For the non-convex data, both the algorithms behave in a similar manner, though they do not generate the actual number of clusters. However, such clusters or partitions are useful for some problems like online identification of evolving fuzzy models [1, 3]. In such applications, identification of cluster centres is more important than identifying patterns. With a higher value of radius, both the crescent shape distributions would be merged into a single cluster with the cluster centre in between the two crescents. Thus, the centre would not lie on the actual distribution. This is because both the algorithms are based on computation of means and use Euclidean distance as a distance measure. Also, the data distribution in the two crescents is somewhat uniform. In Fig. 4, the black dots indicate the centres identified by ELM. It clearly shows that the centres identified by ELM can be considered as modes of the distribution. Table III compares the two algorithms in terms of cluster purity along with other parameters for the benchmark datasets. In case of the Iris dataset, ELM gives high cluster purity, and in case of Wine dataset it is marginally lower than mean-shift. As far as the average distance between the cluster centres are concerned they are not too far from each other. If we compare the execution time, mean-shift incurs more time in all the datasets due to its iterative nature. The significant difference in execution time of ELM over mean-shift is noticeable in column III of Table II where the execution time is calculated after addition of just one sample. After executing both the algorithms once for each of the three synthetic data sets, a sample, x = [0.6 0.2] was added to all the datasets. Due to the incremental nature, ELM just needed an additional 23.35 ms. ELM reused all the information, like local mean and variance calculated in the past, and just updated them when this new sample arrived. On the other hand, addition of just one sample to the data sets induced mean-shift to re-compute the clusters over the entire data sets. It is worth to mention here that we have programmed both the algorithms in a general manner without giving any special emphasis to software principles for optimization of execution time. Also, mean-shift is mostly applied to data with not very high dimensions and has been predominantly applied to image processing area. So, for preliminary experiments, we selected two bench mark datasets with reasonable number of features. Finally, both ELM and mean-shift were applied to image colour segmentation and the results are shown in Fig. 5. For the given image, ELM generated 22 clusters and mean-shift generated 25 clusters. The result achieved by ELM is comparable to mean-shift.

TABLE II. SYNTHETIC DATA SETS TEST RESULTS

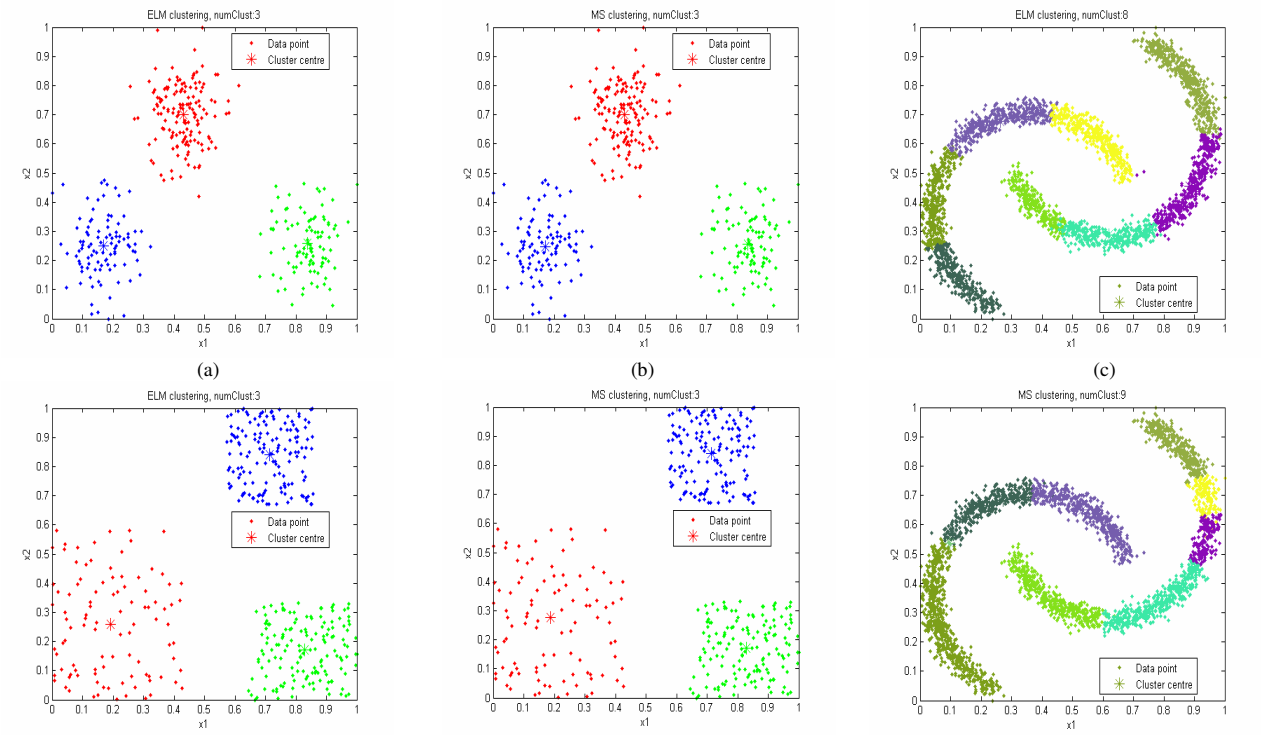
Data set	#Clusters		Execution Time (overall) (ms)		Execution Time (ms) (addition of one sample)		Average distance between centers
	ELM	*Mean-shift*	*ELM*	*Mean-shift*	*ELM*	*Mean-shift*	
	Avg., Mode	*Avg., Mode*					
DS1	3,3	3,3	80.30	129.53	83.46+23.35 = 106.81	131.18+ 131.59 = 262.77	0.00001
DS2	3.2,3	3,3	160.96	250.44	161.43+23.35 = 184.78	254.12+ 240.87 = 494.99	0.00086
DS3	8,8	9,9	929.63	18442.18	927.31+23.38 = 950.69	18405.94+ 18414.03 = 36819.97	0.13534

TABLE III. BENCHMARK DATA SETS TEST RESULTS

Data set	#Clusters		Cluster Purity (%)		Execution Time (ms)		Average distance between centers
	ELM	*Mean-shift*	*ELM*	*Mean-shift*	*ELM*	*Mean-shift*	
	Avg., Mode	*Avg., Mode*					
Iris	4.6,5	3.6,4	91.92	86.75	104.63	148.27	0.95082
Wine	6,6	5,5	78.73	79.64	87.60	122.69	1.27527

V. CONCLUSIONS AND FUTRURE WORK

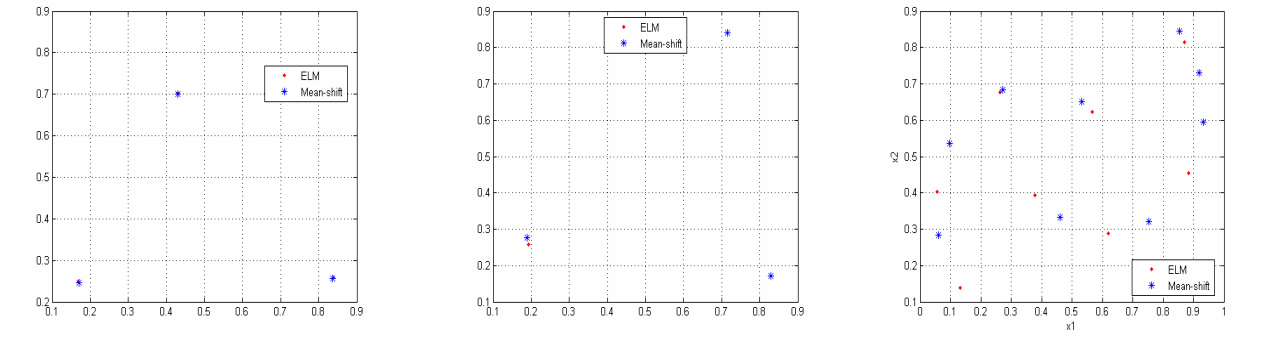
In this paper a new evolving clustering approach is proposed that inherits the basic concept from mean-shift. The algorithm mainly requires recursive calculations of two parameters viz. local mean and local variance. Like other non-parametric approaches, it requires a predefined parameter, the radius or bandwidth. The preliminary experimental results presented here show that ELM results are comparable to mean-shift and requires less execution time compared to mean-shift approach especially in case of large data sets. Also, while processing it requires only the current sample and previously stored local mean and variance, thus needs only a meager amount of memory. Though only experimental results with few datasets have been presented, the results are very promising. A thorough experimental analysis and comparison with other online clustering approaches is needed to be done. At present, the radius parameter is predefined. Though the literature provides several approaches for adaptation of the radius, there is still scope for an investigation especially for the online case. Though, the algorithm updates the cluster centres automatically, a strategy is required to remove outdated cluster centres. Further, during our experiments, we simply considered clusters with less than three samples as outliers. A formal approach for identification of outliers is required. In online approaches when to declare a cluster as an outlier is



(a) (b) (c)

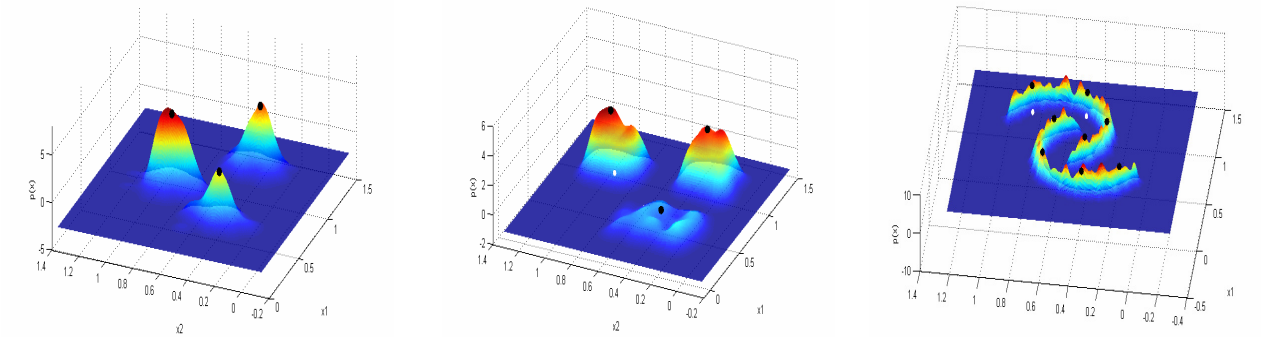
(d) (e) (f)

Figure 2. ELM (a, c, d) and Mean-shift (b, e, f) clustering over synthetic data



(a) (b) (c)

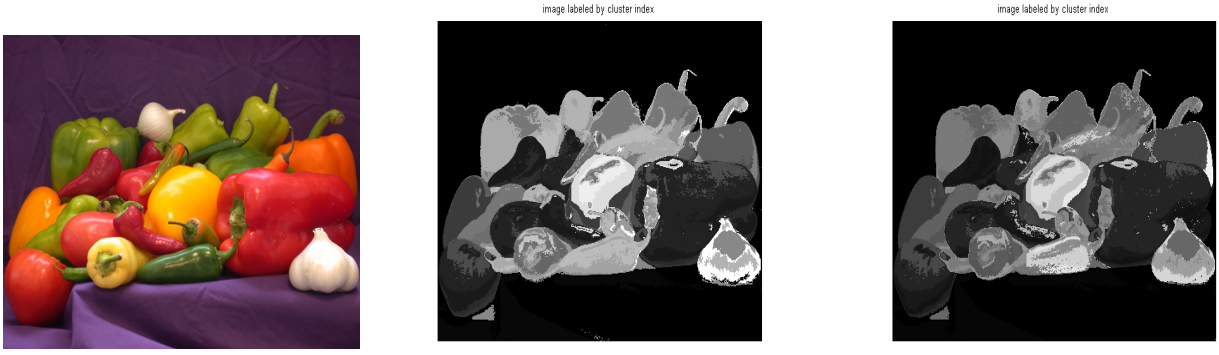
Figure 3. Cluster centres identified by ELM and mean-shift (a) DS1 (b) DS2 (c) DS3



(a) (b) (c)

Figure 4. Density Estimates using KDE. Black dots indicate the modes identified by ELM. (a) DS1 (b) DS2 (c) DS3

more difficult. For example, a cluster presently with only two



(a) (b) (c)

Figure 5.(a) original Image (b) Segmented image using ELM (c) Segmented image using mean-shift

samples may receive more samples later and thus may not qualify for an outlier in future. ELM is anticipated to be applicable to numerous areas. For example, online identification of fuzzy models that requires mainly detection of modes, online object tracking etc. In future, we intend to develop applications using this simple online clustering approach.

REFERENCES

[1] P. Angelov, "Fuzzily Connected Multimodel Systems Evolving Autonomously From Data Streams," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 41, pp. 898-910, 2011.

[2] P. Angelov, D. Filev, and N. Kasabov Eds., *Evolving Intelligent Systems: Methodology and Applications*. John Willey and Sons, IEEE Press Series on Computational Intelligence, April 2010.

[3] R. Dutta Baruah and P. Angelov, "Evolving fuzzy systems for data streams: a survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol. 1, pp. 461-476, 2011.

[4] E.Lughofer, *Evolving Fuzzy Systems- Methodologies, Advanced Concepts and Applications* vol. 266 Berlin, Heidelberg: Springer 2011.

[5] K. Fukunaga, "Clustering," in *Introduction to Statistical Pattern Recognotion*, second ed San Diego, CA, USA: Academic Press Professional, Inc., 1990, pp. 508-559.

[6] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, pp. 603-619, 2002.

[7] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *Information Theory, IEEE Transactions on,* vol. 21, pp. 32-40, 1975.

[8] E. Parzen, "On Estimation of a Probability Density Function and Mode," *The Annals of Mathematical Statistics,* vol. 33, pp. 1065-1076, 1962.

[9] C.Yizong, "Mean shift, mode seeking, and clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 17, pp. 790-799, 1995.

[10] A. Touzani and J. G. Postaire, "Mode detection by relaxation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 10, pp. 970-978, 1988.

[11] M. Herbin, N. Bonnet, and P. Vautrot, "A clustering method based on the estimation of the probability density function and on the skeleton by influence zones: application to image processing," *Pattern Recogn. Lett.,* vol. 17, pp. 1141-1150, 1996.

[12] H. Bohyung, D. Comaniciu, Z. Ying, and L. S. Davis, "Sequential Kernel Density Approximation and Its Application to Real-Time Visual Tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 30, pp. 1186-1197, 2008.

[13] M. Song and H. Wang, "Highly efficient incremental estimation of gaussian mixture models for online data stream clustering," *SPIE: Intelligent Computing: Theory and Applications,* pp. 174-183, 2005.

[14] A. Declercq and J. H. Piater, "Online learning of gaussian mixture models - a two level approach," in *Intl. Conf. Comp. Vision, Imaging and Comp. Graph. Theory and Applications* 2008, pp. 605-611.

[15] W. F. Szewczyk, "Time-evolving adaptive mixtures," Tech. Report, National Security Agency2005.

[16] A. Frank and A. Asuncion. UCI Machine Learning Repository [Online]. Available: http://archive.ics.uci.edu/ml

APPENDIX

Algorithm 1

$\mathbf{x}_i$ = current data sample, i indicates the instant at which $\mathbf{x}$ has arrived or simply the position in a data sequence. $\mathbf{x}_i$ is n-dimensional vector, $\mathbf{x} \in R^n$.

$\boldsymbol{\mu}_i = i^{th}$ cluster centre (local mean).
r = radius, which is a predefined parameter required during clustering.
$\boldsymbol{\sigma}_i$ = average from i^{th} centre to all the samples in cluster i.
c = number of clusters,
$count_i$ = number samples belonging to i^{th} cluster.
$\|\mathbf{x}- \mathbf{y}\|$ = norm of vector $\mathbf{x}$-$\mathbf{y}$.
$\boldsymbol{\alpha}_i$ = sum of all x in i^{th} cluster, $\boldsymbol{\beta}_i$ = sum of all x^2 in i^{th} cluster.

Step 1: Read the first sample $\mathbf{x}_1$.
Create the first cluster around this sample and set the following.
$\boldsymbol{\mu}_1 = \mathbf{x}_1$, $\boldsymbol{\sigma}_1 = \mathbf{0}$, $c = 1$, $count_1 = 1$, $\boldsymbol{\alpha}_i = \mathbf{x}$, $\boldsymbol{\beta}_i = \mathbf{x}^2$

Step 2: Repeat the following steps until samples are available (or until not interrupted).

Step 3: Read the next sample $\mathbf{x}_i$.
Calculate the distance between $\mathbf{x}_i$ and all the existing cluster centres $\mathbf{cc}_j$
$dist_{ij} = \| \mathbf{x}_i - \boldsymbol{\mu}_j \|$, for all $j = 1, .., c$

Step 4: Select the cluster centres that satisfies the following:
$dist_{ij} < (max\ (\boldsymbol{\sigma}_j\ , r) + r)$ for all $j = 1, .., c$

Let $s1$ be the set of indices of all such cluster centres that satisfy the above condition.

Step 5: IF s1 is not empty THEN go to Step (6)
ELSE Create a new cluster around $\mathbf{x}_i$.
$c = c+1$, $\boldsymbol{\mu}_c = \mathbf{x}_i$, $\boldsymbol{\sigma}_c = \mathbf{0}$, $count_c = 1$,
go to Step (2)

Step 6: Select the p^{th} cluster centre that is closest to $\mathbf{x}_i$ and satisfies the condition given in Step (4).
$dist_{ip} = \| \mathbf{x}_i - \boldsymbol{\mu}_p \| = \min(\|\mathbf{x}_i - \boldsymbol{\mu}_l\|)$ for all $l \in s1$

Considering that now $\mathbf{x}_i$ belongs to p^{th} cluster, update the cluster centre and average distance.
$\boldsymbol{\beta}_p = \boldsymbol{\beta}_p + \mathbf{x}_i^2$, $\boldsymbol{\alpha}_p = \boldsymbol{\alpha}_p + \mathbf{x}_i$
$\mathbf{mean} = (count_p \times \boldsymbol{\mu}_p + \mathbf{x}_i\ ) / (count_p + 1)$
$\mathbf{variance} = (\boldsymbol{\beta}_p + count_p \times \mathbf{mean}^2 - 2 \times \mathbf{mean} * \boldsymbol{\alpha}_p) / (count_p + 1)$
$\boldsymbol{\mu}_p = \mathbf{mean}$, $\boldsymbol{\sigma}_p = \mathbf{variance}$, $count_p = count_p + 1$

Step 7: Since now the position of the p^{th} cluster centre has shifted.
Determine if it is required to be merged with any existing cluster centre that is close enough.
$dist_{pj} = \| \boldsymbol{\mu}_p - \boldsymbol{\mu}_j \|$ for all $j = 1, .., c$ and $j \neq p$

Select the cluster centres that satisfy the following condition.
$dist_{pj} < max(\boldsymbol{\sigma}_p\ , r) + max(\boldsymbol{\sigma}_j, r)$ for all $j = 1, .., c$ and $j \neq p$

Let $s2$ be the set of indices of all such cluster centres that satisfy the above condition.

Step 8: If $s2$ is not empty then select the closest cluster centre q.
$dist_{pq} = \| \boldsymbol{\mu}_p - \boldsymbol{\mu}_q \| = min\ (\|\boldsymbol{\mu}_p - \boldsymbol{\mu}_l\|)$ for all $l \in$ s2

Merge cluster p and cluster q and update centre position, variance, and count.