International Journal of Database Theory and Application
Vol.7, No.3 (2014), pp.191-200
http://dx.doi.org/10.14257/ijdta.2014.7.3.17

An Improved Method for Robust and Efficient Clustering Using EM Algorithm with Gaussian Kernel

Aakash Soor and Vikas Mittal

Department of Electronics & Communication Engineering
National Institute of Technology Kurukshetra, Haryana, India
aakashsoor@gmail.com, vikas_mittal@nitkkr.ac.in

Abstract

Clustering is one of the main tasks used in pattern recognition and classification. Out of many methods that have been reported till date the most widely used methods are based on likelihood approach of mixture model. Among different mixture models, Expectation Maximization for Gaussian Mixture is most exploited and trusted algorithm for data clustering. However, it has some short comings such as initial parameters are to be given a-priori, convergence speed is slow and the results obtained are highly dependent upon the initial parameters. Many variations have been carried out in implementing EM algorithm but still there is ample scope for improvement. The proposed algorithm tries to overcome these shortcomings and provide more robust and efficient version of clustering algorithm. An improvement related to cluster partitioning is proposed in the existing algorithm resulting some advantages. The robustness and efficacy of the algorithm is demonstrated qualitatively as well as quantitatively with the help of some experiments.

Keywords: *Expectation Maximization, Clustering, GMM, finite mixture models, kernel methods*

1. Introduction

Pattern Recognition and classification forms an integral part of the task of image recognition. It serves the applications such as object identification, face recognition, surveillance *etc*. The first and foremost step of pattern recognition and classification is image clustering followed by classification.

Clustering is basically used to make a distinction between different objects present in an image based on some common property such as intensity, hue, saturation, pdf, cdf *etc*. The clustered image is then matched with the stored data to identify an object. Object can be anything like a human being, animals, face, finger print, trees, or any day to day objects.

The natural process after clustering is its classification. Data classification is a process of assigning a label to each and every point of the data to perceive the similarities among those points.

In the literature, various methods have been reported for clustering. Some of them are Thresholding [1], histogram based method, region growing method [2] *etc*. The most widely used method is model based clustering as it has capability to use prior knowledge to model the uncertainty in a probabilistic manner. This prior knowledge can be the spatial relationships between neighboring points/pixels in a/an data/image [3] or the whole structure of a/an data/image. So, a systematic method should be adopted which can exploit this prior knowledge to calculate the parameters efficiently. The method used here follows Expectation Maximization algorithm.

EM algorithm [4] is a finite mixture model based algorithm where missing data can be predicted using prior information. It is an iterative statistical technique for computing maximum likelihood estimates from incomplete data. It can be applied with any mixture model but here it is applied with Gaussian Mixture Model [5] because of the following advantages.

1. Gaussian Mixture model can be determined by using only two parameters i.e. mean and variance.
2. Provides honest solution of the prediction problem [6].
3. Provides an explicit and closed form of likelihood [6].

It has generally been employed for a wide variety of parameter estimation problems. EM algorithm alternates between two steps namely Expectation (E) step and Maximization (M) step. EM algorithm is guaranteed to converge to a local maximum of the data log-likelihood as function its parameters [7]. It gives accurate results provided initial parameters are known. EM algorithm has attractive features such as reliable global convergence, low cost per iteration, economy of storage, and ease of programming. However, it has linear convergence rate [8] and it is an experimentally proven fact that EM converges slowly in presence of overlapping clusters. Also, as its convergence is highly dependent on initial parameters, it may not be always possible to know it apriori [9, 10]. The need to overcome these problems is the motivation to develop the proposed algorithm.

The rest of the paper is organized as follows. Section II describes theoretical background of clustering and gives an over view of existing algorithm. The proposed algorithm is presented in Section III followed by Results and Discussion in Section IV. Paper is concluded in Section V.

2. Review of Background

This Section starts with the basic theory related to clustering algorithms. The first mentioned algorithm is based on ML estimation and is given as:

2.1. Overview of existing EM algorithm for Gaussian Mixture Model [13]

This method exploits the property of Entropy of data where the initial number of clusters are taken equal to the data points which are taken as their means. Mixing probabilities are taken by their information content and is given by $-\sum_{j=1}^{K}\alpha_j \ln \alpha_j$ Maximum likelihood is achieved by minimizing the entropy. So the joint function becomes

$$L(\alpha,\theta) = \sum_{i=1}^{n}\sum_{j=1}^{K} z_{ji}\ln\left[\alpha_j P\left(x_i,\theta_j\right)\right] \\ + \beta\sum_{i=1}^{n}\sum_{j=1}^{K}\alpha_j \ln \alpha_j \qquad (1)$$

Here 1>β > 0 is a mixing coefficient given by,

$$\beta \\ = min\left\{\frac{\sum_{j=1}^{K} exp\left(-\eta n|\alpha_j^{new}-\alpha_j^{old}|\right)}{K}, \frac{1-\alpha_1^{EM}}{-\alpha_1^{old}E}\right\} \qquad (2)$$

Where, $\eta = min\left\{1, 0.5^{\left\lfloor\frac{d}{2}-1\right\rfloor}\right\}$, $E = \sum_{j=1}^{K} \alpha_j \ln \alpha_j$

This algorithm tackles some issues such as

- Initialization problem is removed by treating all the points as starting points.
- It provides robust unsupervised clustering.
- The algorithm can tackle large number of components.

The above algorithm works well for medium sized data. But its efficiency decreases with increase in the number of data points. The reasons and issues are mentioned below.

- A large data of size $n\ x\ d$, requires high matrix size of $n\ x\ n$ for posterior PDF by (3) which in turn will require a high memory space.
- Also this algorithm fails if the data clusters are not well separated. It is deduced experimentally that for a data having 200 points per cluster, the algorithm fails if $var(a) + var(b) > d(\mu_a, \mu_b)$. Such shortcomings are reduced in the proposed algorithm.

3. Proposed Method

The proposed method overcomes the problems mentioned in Section II. The modifications proposed are

1) Data points closer to each other may possess few similar features. So partitioning method is exploited where original data points are grouped together initially in a manner to keep the number of groups high but much lower than the actual data points. This results in the reduction in computer memory usage and increase in computational speed.

2) Another modification is made in the calculation of β. It has been observed that the β value fluctuates according to the mixing coefficients (8). This provides good convergence for low overlapping data but fails for data with large overlap where it keeps on calculating well for some time but abruptly fluctuates and gives absurd result. It is demonstrated in Experiment 1. So it is modified to a simpler formula

$$\beta = exp\left(\frac{-(t-1)^{1.1}}{20}\right) + exp\left(\frac{(t-1)^{0.2}}{20}\right) - 1 \qquad (3)$$

The plot of β is shown in Fig. 1, β manages the weight of the entropy of α_j. Initially it is kept close to one so as to quickly remove the irrelevant cluster within first few iterations. As iterations increases, β decreases, which enables the clusters to grow and converge to an optimal result. Further, β is increased very slowly so as to remove the remaining unwanted clusters.

First, the joint function (1) is maximized with respect to α_j under the constraint that $\sum_{j=1}^{K} \alpha_j = 1$ by using lagrangian coefficient. The α_j^{new} is given by

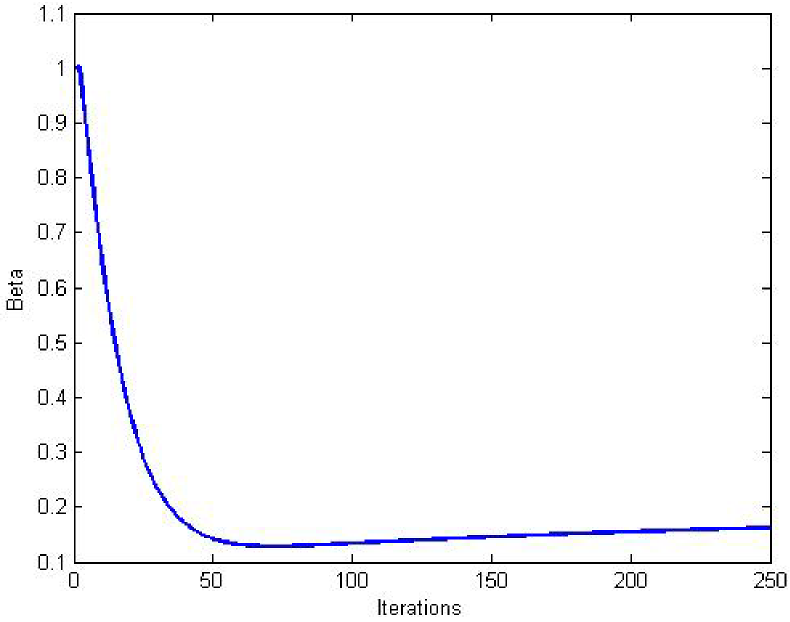

Figure 1. Plot of Beta vs. Iterations

$$\alpha_j^{new} = \alpha_j^{EM} + \beta\alpha_j^{old}\left(\ln\alpha_j^{old} - \sum_{m=1}^{K}\alpha_m^{old}\ln\alpha_m^{old}\right) \tag{4}$$

For detailed derivation, refer [13].

The rest of the procedure for the proposed algorithm is summarized as follows

1. Set initial values: $\varepsilon > 0, \beta^{(0)} = 1,, gap = 5, K^{(0)} = \frac{n}{gap}, \alpha_j^{(0)} = \frac{1}{K}, t = 1$. $\mu_j^{(0)} = \frac{group_j}{gap}$ where $group_j = x_{(1+(j-1)gap)} + x_{(2+(j-1)gap)} \cdots x_{(gap+(j-1)gap)}$.
2. Deduce $\Sigma^{(0)}$ by $\Sigma_j^{(0)} = \text{variance}(group_j)$ and calculate $z_{ji}^{(0)}$ by using $\alpha_j^{(0)}$, $\mu_j^{(0)}$ and $\Sigma_j^{(0)}$ by

$$z_{ji} = \frac{\alpha_j P(x_i|\mu_j, \Sigma_j)}{P(x_i|\theta)} = \frac{\alpha_j P(x_i|\mu_j, \Sigma_j)}{\sum_{j=1}^{K}\alpha_j P(x_i|\mu_j, \Sigma_j)} \tag{5}$$

3. Compute $\alpha_j^{EM(t)}$ and $\mu_j^{(t)}$ with $z_{ji}^{(0)}$ using maximization steps of EM algorithm [4,11] given by

$$\alpha_j = \frac{\sum_{i=1}^{n} z_{ji}}{n} \tag{6}$$

$$\mu_j = \frac{\sum_{i=1}^{n} z_{ji}\, x_i}{\sum_{i=1}^{n} z_{ji}} \tag{7}$$

4. Update $\alpha_j^{(t)}$ with $\alpha_j^{EM(t)}$ and $\alpha_j^{(t-1)}$ using (4).

5. Update $\beta^{(t)}$ using t by (3).

6. Update $K^{(t-1)}$ to $K^{(t)}$ by keeping those clusters which follow $\alpha_j^{(t)} > 1/n$ and adjust $\alpha_j^{(t)}$ and $z_{ji}^{(t-1)}$ by normalizing the them given by

$$\alpha'_j = \frac{\alpha'_j}{\sum_{j=1}^{K(updated)} \alpha'_j} \,,$$

$$z_{ji} = \frac{z'_{ji}}{\sum_{j=1}^{K(updated)} z'_{ji}} \tag{8}$$

and adjust $\mu_j^{(t)}$ by selecting accepted clusters.Check IF $t \geq 60$ and $K^{(t-1)} - K^{(t)} = 0$, THEN assign $\beta^{(t)} = 0$.

7. Update $\Sigma_j^{(t)}$ with $\mu_j^{(t)}$ and $z_{ji}^{(t-1)}$ by

$$\Sigma_j = \frac{\sum_{i=1}^{n} z_{ji}(x - \mu_j)(x - \mu_j)^T}{\sum_{i=1}^{n} z_{ji}} \tag{9}$$

To avoid the singularity problem in covariance matrices, they are updated by equation,

$$\Sigma_j = (1 - \gamma)\Sigma_j + \gamma Q \tag{10}$$

8. Update $z_{ji}^{(t)}$ with $\alpha_j^{(t)}, \mu_j^{(t)}$ and $\Sigma_j^{(t)}$ by (5).

9. Calculate $\alpha_j^{EM(t+1)}$ and $\mu_j^{(t+1)}$ from $z_{ji}^{(t)}$ by (6) and (7).

10. Find $tol = \max_{1 \leq j \leq K^{(t)}} || \mu_j^{(t+1)} - \mu_j^{(t)} ||$ and update $K^{(t-1)}$ to $K^{(t)}$ by keeping those clusters which follow $\alpha_j^{EM(t+1)} > 1/n$ and adjust $\alpha_j^{EM(t+1)}$, $z_{ji}^{(t)}$, $\alpha_j^{(t)}$ and $\mu_j^{(t+1)}$ by selecting the accepted clusters and adjust $z_{ji}^{(t)}$ by (8).

11. Check IF $tol < \varepsilon$, THEN STOP, ELSE $t = t + 1$ and return to step 4.

As stated above, algorithm starts with fairly low number of points which not only reduces clusters but reduces burden on computer. By keeping initial high value of β, it is ensured to have the maximum participation of information contained by the entropy term. This enables to quickly remove unwanted clusters and move towards convergence. Mixing coefficients actually tells the contribution of each clusters in the form of prior PDF. The minimum value it can achieve is if all the points are treated as clusters, *i.e.*, only one point per cluster. If for any clusters, the value of $\alpha_j^{(t)}$or α^{EM} is lower than the minimum value, it means it do not contribute any information and is thus neglected (as shown in step 6 and 10).

In the above algorithm, 'gap' can be taken as any integer for different results. It is empirically deduced that the above used value gives good quality results. The initial means and variances are calculated by conventional method.

4. Results and Discussions

All the experiments are performed on a PC with Core i3-processor, 2.40 GHz and 4 GB RAM with 1 GB NVidia graphic card on Windows 7 platform. For all the experiments, '*gap*' is taken as 5. Quantitative analysis of all the experiments and a complete flow chart of the algorithm is also presented.

4.1. Experiment 1

In this experiment, a two component bivariate data of 300 points is generated with $\alpha_1 = \alpha_2 = 0.5$,$\mu_1 = [0 \quad 0]$,$\mu_2 = [2 \quad 2]$ and $\Sigma_1 = \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Both algorithms are applied on same data and the output is shown.

It can be seen that the parameters estimated by the proposed algorithm is much closer to the original parameter values.

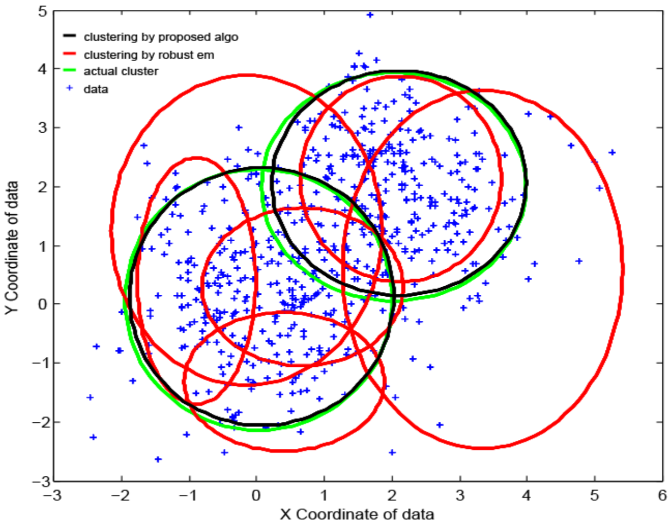

Figure 2. Scatter Plot with Clusters of Two Component Data for Experiment 1

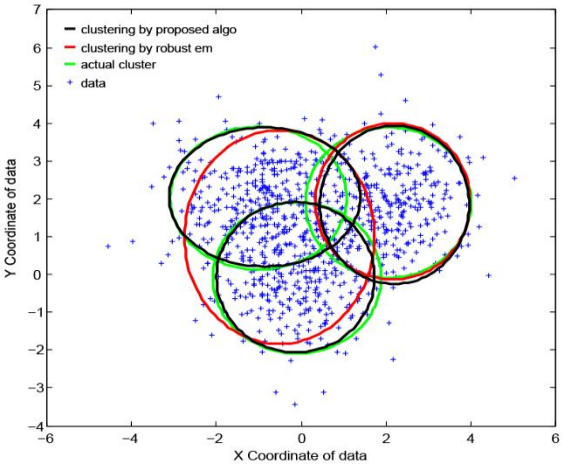

Figure 3. Scatter Plot with Clusters of Three Component Data for Experiment 2

4.2. Experiment 2

In this experiment, a 3 component bivariate data of 300 points is generated with $\mu_1 = [0 \quad 0]$, $\mu_2 = [2 \quad 2]$, $\mu_3 = [-1 \quad 2.5]$, $\alpha_1 = \alpha_2 = \alpha_3 = 0.3333$, and $\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Both algorithms are applied on same data and the output is shown above.

The robust EM algorithm misestimated the total number of clusters and gave absurd result. On the other hand proposed algorithm estimated the parameters accurately.

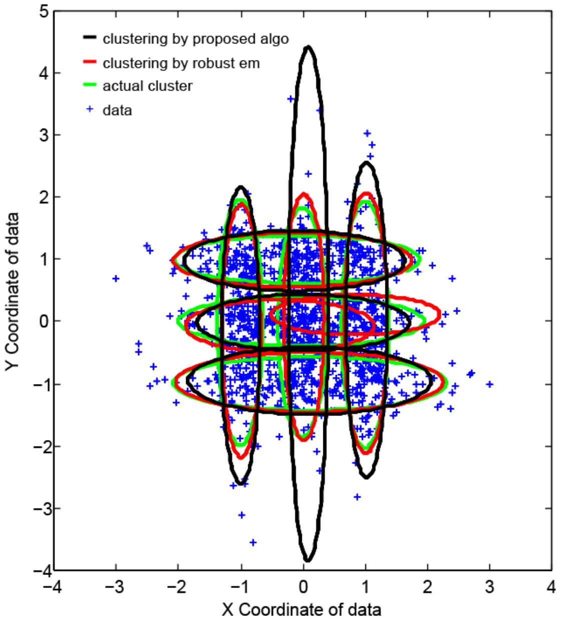

Figure 4. Scatter Plot with clusters of six component data for Experiment 3

4.3. Experiment 3

In this experiment, means are changed to $\mu_1 = \mu_3 = [0 \quad 0]'$, $\mu_3 = [-1 \quad 0]'$, $\mu_4 = [1 \quad 0]'$, $\mu_5 = [0 \quad 1]'$, $\mu_6 = [0 \quad -1]'$, keeping all other parameters same as Experiment 2.

When the clusters are well separated, both the algorithms yield accurate results. But when the data set generated has high overlap in between clusters, robust EM algorithm fails and estimated an extra cluster. Although the proposed algorithm estimates all the clusters, it deviates little bit from the actual result

Table 1. Quantitative Analysis of the Above Experiments

Exp #	Actual Parameter values		Robust EM Algorithm	Proposed Algorithm
1	α	$[0.5 \quad 0.5]$	$[0.09 \quad 0.08 \quad 0.170.04 \quad 0.37 \quad 0.23]$	$[\,0.5509 \quad 0.4491]$
	μ	$\begin{bmatrix} 0.0265 & 2.0221 \\ 0.0742 & 1.9947 \end{bmatrix}$	$\begin{bmatrix} -0.89 & 0.40 & -0.14 & 3.33 & 2.12 & 0.67 \\ 0.39 & -1.31 & 1.25 & 0.58 & 2.12 & 0.29 \end{bmatrix}$	$\begin{bmatrix} 0.0824 & 2.1116 \\ 0.1413 & 2.0654 \end{bmatrix}$
	Iterations	-	432	44
	Time(sec)	-	18.1326	3.4265
2	α	$\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$	0.3095 0.6905	0.3331 0.3020 0.3649
	μ	0.0, 2.0, -1.0 0.0, 2.0, 2.0	2.1635, -0.5094, - 1.9276, 0.9890, -	-0.1301, 2.2033, -0.8613 -0.0704, 1.8449, 2.0477
	Iterations	-	243	95
	Time(sec)	-	25.8143	10.1552
3	α	$\begin{bmatrix} \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{bmatrix}$	0.137,0.133,0.15,0.065,0.198,0.12, 0.193	0.012, 0.303, 0.082, 0.078, 0.261, 0.260
	μ	0, 0, -1,1, 0, 0 0, 0, 0, 0, 1, -1	1.014,-0.99,-0.37,0.85,-0.16,0.017,0.108 -0.019,-0.147,-0.068,0.11,0.99,0.07,-0.97	0.088,0.003,-1.000, 1.030, -0.134, 0.115 0.279,-0.012,-0.212, 0.022,0.983, -0.951
	Iterations	-	524	54
	Time(sec)	-	79.7010	19.5016

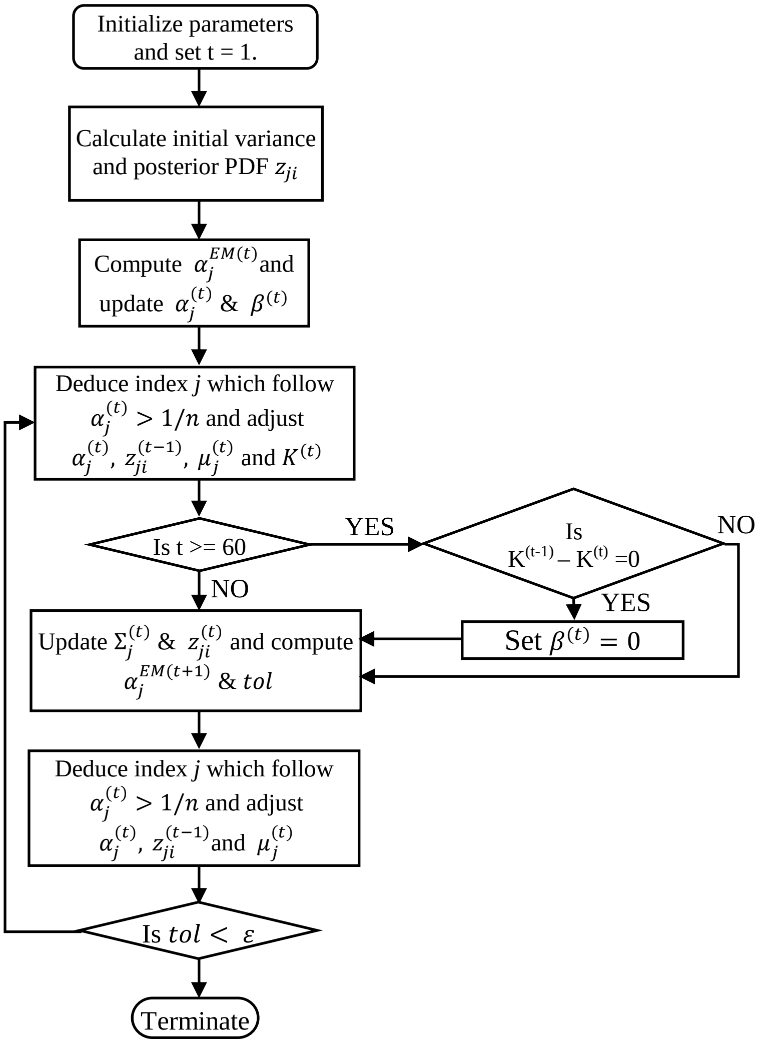
Initialize parameters and set t = 1.
Calculate initial variance and posterior PDF z_{ji}
Compute $\alpha_j^{EM(t)}$ and update $\alpha_j^{(t)}$ & $\beta^{(t)}$
Deduce index j which follow $\alpha_j^{(t)} > 1/n$ and adjust $\alpha_j^{(t)}$, $z_{ji}^{(t-1)}$, $\mu_j^{(t)}$ and $K^{(t)}$
Is t >= 60
YES
Is $K^{(t-1)} - K^{(t)} = 0$
NO
NO
YES
Update $\Sigma_j^{(t)}$ & $z_{ji}^{(t)}$ and compute $\alpha_j^{EM(t+1)}$ & tol
Set $\beta^{(t)} = 0$
Deduce index j which follow $\alpha_j^{(t)} > 1/n$ and adjust $\alpha_j^{(t)}$, $z_{ji}^{(t-1)}$and $\mu_j^{(t)}$
Is $tol < \varepsilon$
Terminate

Figure 5. Flow Chart for the Improve Algorithm

5. Conclusion

From the above experiments, it can be deduced that the proposed algorithm works well for variety of data sets. From experiment 1, it is concluded that convergence quality gets reduced with the decreasing number of data points. From experiment 2 and 3, it can be interpreted that convergence quality of the robust EM algorithm reduces as the overlap among clusters increases. Experiment 3 shows the interpretation of parameters on a variety of data sets. By repeating the above experiments, it is deduced that for obtaining accurate and precise results, two data clusters must be separated from each other with a distance greater than or equal to the sum of their variance in the relevant axes.

References

[1] K J. Batenburg and J. Sijbers, "Optimal Threshold Selection for Tomogram Segmentation by Projection Distance Minimization", IEEE Transactions on Medical Imaging, vol. 28, no. 5, pp. 676-686, **(2009).**

[2] R. Nock, F. Nielsen, “Statistical Region Merging”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 26 Issue 11,pp.1452-1458 , **(2004).**

[3] T. M. Nguyen and Q. M. J. Wu, “Dirichlet Gaussian mixture model: Application to image segmentation,” Image Vis. Comput., vol. 29, no. 12, pp. 818–828, **(2011).**

[4] A.P. Dempster, N.M. Laird, and D. Rubin. Maximum likelihood estimation from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, 39(1), pp.1–38, **(1977).**

[5] N. Pal and S. Pal, “A review of image segmentation techniques,” Pattern Recognition, vol. 26, no. 9, pp. 1277–1294, **(1993).**

[6] R.A. Davis, “Gaussian Processes, Encyclopedia of Environmetrics, Section on Stochastic Modeling and Environmental Change”, (D. Brillinger, Editor), Wiley, New York, **(2001).**

[7] C.F.J. Wu, “On the convergence properties of the EM algorithm”, Annals of Statistics 11, pp. 95–103, **(1983).**

[8] R.A. Redner, H.F. Walker, “Mixture densities, maximum likelihood and the EM algorithm”. SIAM Review 26 (2), pp. 195–239, **(1984).**

[9] J. Ma, L. Xu, M.I. Jordan, “Asymptotic convergence rate of the EM algorithm for Gaussian mixtures”, Neural Computation 12 2881–2907, **(2000).**

[10] C. Ordonez and E. Omiecinski, “FREM: Fast and Robust EM Clustering for Large Data Sets,” Proc. ACM Conf. Information and Knowledge Management, **(2002),** Nov 04-09, McLean, Virginia, USA.

[11] W. Yao, “A note on EM algorithm for mixture models”, Statistics & Probability Letters, vol. 83 Iss. 2, pp. 519–526, **(2013)**.

[12] Q. Zhao, V. Hautamäki, I. Kärkkäinen, P. Fränti, “Random swap EM algorithm for Gaussian mixture models”, Pattern Recognition Letters, Vol. 33, Iss. 16, pp. 2120-2126, (**2012).**

[13] M.S. Yang, C.Y. Lai, C.Y. Lin, “A robust EM clustering algorithm for Gaussian mixture models”, ELSEVIER Pat Rec. 45, 3950–3961, (**2012).**

Authors

Aakash Soor, the author is currently a Post Graduate student in Electronics and Communication Engineering Department at National Institute of Technology Kurukshetra, Haryana, India. He graduated in 2012 with specialization in Electronics and Communication Engineering from Dr. Kedar Nath Modi Institute of Engineering & Technology, Modinagar, India. His area of research is data clustering for Pattern Recognition.

Vikas Mittal, the author is currently Associate Professor in Electronics and Communication Engineering Department at National Institute of Technology Kurukshetra, India. He Graduated in 1992 and Post Graduated in 2004 with specialization in Electronics and Communication Engineering from NIT Kurukshetra, India. Presently, pursuing Doctorate from NIT Kurukshetra, India, he is author of many papers, articles and monographs with interests mainly in Signal and Image Processing with a focus on Satellite Image Processing.