International Journal of Computer Sciences and Engineering
Vol.**11**, Special Issue.**1**, pp.**184-189**, November **2023**
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

ISSN: 2347-2693 (E)

Research Paper

Prevention of Empty Clusters and Incomplete Data Problems using Modified K-Means and Gaussian Mixture Model

Sanjib Saha[1]

[1]Dept. of Computer Science and Engineering, National Institute of Technology, Durgapur and Dr. B. C. Roy Engineering College, Durgapur, India

Author's Mail Id: sanjib.saha@bcrec.ac.in

Abstract: Cluster analysis, in unsupervised learning, divides similar data into groups or clusters that are meaningful and useful. Due to good performance in clustering on massive data sets K-Means clustering is feasible in multiple areas of science and technology. The clustering algorithms may face problems of empty clusters and incomplete data. This empty cluster problem is caused by bad initialization of the center point and this may route to signifying performance degradation. In this theme, the K-Means clustering algorithm is revisited from the probabilistic viewpoint and reformed by the similarity among the K-Means and finite Gaussian Mixture Model (GMM). The initial centroids or current best estimate for the parameters are calculated from the list of all data, known and unknown. Therefore, any two or more primal centroids may not be equal or not very close to each other and data will be assigned to the appropriate clusters with closely fair centroids. The newly proposed modified K-Means using GMM of the Expectation Maximization approach efficiently eliminate the empty cluster and incomplete data problems.

Keywords: Unsupervised Learning, Clustering Analysis, K-Means, Expectation Maximization, Gaussian Mixture Model

1. Introduction

Clustering based on K-Means is allied to some other clustering problems. Though K-Means algorithm is one of the elite clustering algorithms it has some drawbacks such as (i) Non-globular clusters (overlapping in data between clusters); (ii) Assume wrong number of clusters; (iii) Finding empty clusters; (iv) Bad initialization to centroid point; (v) Inability to choose the number of clusters.

The empty cluster problem is caused by bad initialization and this may lead to significant performance degradation. The trouble of empty clusters arises when the primary center vectors are such that any two or more of them are either the same or very narrow altogether. In such a state, next to the allocation of data to clusters, data will be assigned to one of the clusters with nearly the same centers, and the others stay empty. This paper presents a new route that efficiently eliminates this empty cluster problem. In this theme, K-Means clustering algorithm is revisited from the probabilistic viewpoint and reformed by the relation among K-Means [1] and finite Gaussian Mixture Model (GMM) [2]. Also, the maximum-likelihood of Expectation Maximization (EM) [3] algorithm is applied to find parameters for mixture density problems and fill in the missing values for incomplete data problems. Here, the proposed algorithm is the merging of two popular algorithms which can be used for large as well as probabilistic datasets.

The review of literature on related works has been discussed in section 2. The K-Means, Gaussian Mixture Model, and Expectation Maximization are described in section 3. Section 4 is devoted to defining the proposed algorithm and section 5 shows results and then analysis to proof the performance of the new approach. The conclusion and future scope of work is discussed in section 6.

2. Related Work

Here begin with the discussion of related works in this literature. K-Means [1] is a process where an N-dimensional population is separated into K sets on the basis of a sample which appears to give groups which are logically effective within-class variance. K-Means method is feasible to process very large samples. So, the K-Means is computationally fast, easy to implement and scientifically efficient.

An iterative computational approach estimates observations having partial data based on maximum likelihood. There is an 'estimation step' followed by a 'maximization step' which is known by EM [3] algorithm, in each iteration of the algorithm. The estimation step in the EM algorithm is quit equal to a process that first approximates or "fills in" the particular data points and then the maximization step calculates the adequate statistics by filled-in values. It is proper to cover the missing values with their expectations given parameter values (E-step), then re-evaluate parameters using a least-squares estimation (variance σ2) algorithm (M-step), and iterate until the estimates show considerable alteration.

The iterative clustering [4] approach computes from a given initial value a refined starting condition. This efficient approach estimates the modes of distribution. The application of this methodology is applied to the well-known K-Means clustering algorithm and shows that a substantial refinement over randomly chosen starting points indeed leads to improve solutions and avoids empty clusters problem.

A reformed version of the K-Means [5] algorithm that effectively removes the empty cluster challenge and in which the solution is simply to add the current cluster centers to the data points when computing new cluster centers at the next iteration. There is no execution degradation due to incorporated modification.

Yang, M. S. et. al. [6] proposed Expectation Maximization clustering algorithm for GMM. The component of GMM was proposed by McLachlan, G. J. et. al. [7]. Huang, T. et. al. [8] proposed model selection for GMM. Patel, E. et. al. [9] proposed GMM in cloud-based clustering. The GMM in e-government clustering was proposed by Androniceanu, A. et. al. [10]. Löffler, M. et. al. [11] proposed GMM in spectral clustering. The GMM in transport was proposed by Chen, Y. et. al. [12]. Viroli, C. et. al. [13] proposed deep learning-based GMM. The deep learning-based GMM in image registration was proposed by Yuan, W. et. al. [14]. Shahin, I. et. al. [15] proposed hybrid GMM and deep neural network for emotion recognition. The unsupervised anomaly detection using autoencoder-based GMM was proposed by Zong, B. et. al. [16]. An, P. et. al. [17] proposed autoencoder-based GMM for cyberattack detection. The anomaly detection using GMM and long-short term memory was proposed by Ding, N. et. al. [18]. Wan, H. et. al. [19] proposed GMM for classification. The feature selection using GMM was proposed by Fu, Y. et. al. [20]. Singhal, A. et. al. [21] proposed prediction of COVID-19 using GMM. The Earthquake phase relationship using GMM was proposed by Zhu, W. et. al. [22].

Many conventional [23-26] and deep learning [27] based research works on applications and variants of GMM have been found in the literature. It motivates researchers to work on variant GMM and compare their merits and demerits.

3. Background Method

3.1 Mathematical Terms with Definition

$object, datapoint - the\ atomic\ element\ of\ clustering,$

$multiples\ of\ which\ grouped\ or\ clustered\ together$

$n - number\ of\ objects\ in\ a\ dataset$

$k - number\ of\ clusters$

$d - dimension$

$R - set\ of\ all\ real\ numbers$

$X - dataset\ to\ be\ clustered$

$x_i - datapoint\ belonging\ to\ X$

$\in - set\ membership$

$C - set\ of\ kmeans\ centers$

$c_j - kmeans\ center\ belonging\ to\ C$

$I(\alpha) - the\ indicator\ function\ on\ predicate\ \alpha$

$\sum - covariance\ matrix$

$\mu_j - Gaussian\ Expectation\ Maximization\ center$

$p(j) - the\ probability\ function\ on\ predicate\ j$

Here we start with a minute talk of relevant algorithms and models.

3.2 K-Means

The K-Means [1] clustering is a well-known partitioning technique. A clustering method constructs k partitions or a set of k clusters and each object of the dataset refers to one cluster for given a dataset of n objects and k ≤ n. In every cluster, there may be a centroid or a cluster delegate. There are different kinds of condition for deciding the significance of partitions. Based on the theories, various methods are given: K-Means, K-Medoids, and Probabilistic clustering.

The K-Means algorithm executes the following three steps and repeat until stable (= no object move group):

Step1. Find out the centroid coordinate.

Step2. Determine the distance of each object to the centroids.

Step3. Find the nearest centroid and group the object based on the least distance.

3.2.1 Pseudo code for the K-Means algorithm

$Inputs\ to\ the\ algorithm\ are$
$- k\ (the\ number\ of\ centers),$

$X(the\ n\ datapoints\ in\ d\ dimensions), and$

$the\ initial\ locations\ of\ the\ centers\ C = \{c_j\}.$

```
$\boldsymbol{KMeans}(X \in R^{n \times d}, k, C)$
while the any $c_j$ change location do
    for $i \in \{1, \ldots, n\}$ do
        $class(x_i) \leftarrow \arg min_j \parallel x_i - c_j \parallel$
    end for
    for $j \in \{i, \ldots, k\}$ do
        $c_j \leftarrow \sum_i I(class(x_i) = j)\, x_i \,/\, \sum_i I(class(x_i) = j)$
    end for
end while
return $C$
```

The reassignment step of the algorithm computes the Euclidean distance. This distance is measured from every point to every cluster mean and the minimum is found, by calculating, $class(x_i) \leftarrow \arg min_j \parallel x_i - c_j \parallel$. Each point is then reassigned to a cluster. The centroid update step then recalculates the mean of each cluster, and revises c_j for all *j*.

3.3 Gaussian Mixture Model

A Gaussian Mixture Model (GMM) [2] is a parametric model of a probability distribution of continual measures. GMM parameters are evaluated from a trained prior model by either the EM algorithm or Maximum APosteriori (MAP).

A Gaussian mixture model which is a parametric possibility density function is represented as a laded addition of n Gaussian component densities as shown by the equation 1:

$$p(x|\ \lambda) = \sum_{j=1}^{n} w_j g(x|\ \mu_j, \sum_j) \qquad (1)$$

Here, x is a d-dimension numeric data, $w_j, j = 1 \ldots n$,-are the blends of loads, and $g(x|\ \mu_j, \sum_j), j = 1 \ldots n$,-are the element Gaussian densities.

Each component density is a d-variant Gaussian function as given by the equation 2:

$$g(x|\ \mu_j, \sum_j) = \exp\{-1/2(x - \mu_j)' \sum_j^{-1} (x - \mu_j)\}/(2\pi)^{d/2}\ |\sum_j|^{1/2} \qquad (2)$$

Here, mean vector μ_j and covariance matrix $\sum_j$. The mixture weights satisfy the restraint that $\sum_{j=1}^{n} w_j = 1$.

3.4 Expectation Maximization

The Expectation-Maximization (EM) [3] algorithm is a very common and iterative method for the estimation of parameters in statistical models where certain observation is incomplete through either maximum likelihood or MAP.

EM underlies a class of algorithms in which there are two steps:

Step1. The Expectation Step: Using the latest best estimate for the parameters of the data model, we make an expression for the log-likelihood for all data, seen and unseen, and, subsequently, borderline the expression to the unseen data. This expression will depend on the latest best estimate for the model parameters and the model parameters dealt as variables.

Step2. The Maximization Step: Given the expression occurring from the former step, for the next estimate we can choose those values as model parameters that increase and maximize the expectation expression. These give the best new estimate for the Bayesian K-Means algorithm.

3.4.1 Pseudo code for the Expectation Maximization algorithm

Input to the algorithm are $k, X,$ *and*

the initial values of $p(j), \{\mu_j\},$ *and* $\{\sum_j\}$.

The function $p(x_i|\mu_j, \sum_j)$ –

is the Gaussian probability density function, and

the term $p(x_i)$*obtained from summing over*

j *the values* $p(x_i|\mu_j, \sum_j)\ p(j)$.

```
GaussianEM $(X \in R^{n \times d}, k, p(j), \{\mu_j\}, \{\sum_j\})$
while the likelihood $L(\{\mu_j\}, \{\sum_j\} \mid X)$ changes do
    //Expectation step
    for $i \in \{1, \ldots, n\}$ do
        for $j \in \{i, \ldots, k\}$ do
        $p(j \mid x_i) \leftarrow p(x_i|\mu_j, \sum_j)\ p(j)\ /\ p(x_i)$
        end for
    end for
    //Maximization step
    for $j \in \{i, \ldots, k\}$ do
        $\mu_j \leftarrow \sum_i^n p(j\ |x_i)\ x_i\ /\ \sum_i^n p(j\ |x_i)$
        $p(j) = \sum_i p(j\ |x_i)\ /\ n$
        for $l, m \in \{1, \ldots, d\}$ do
        $\sum_{jlm} \leftarrow 1\ /\ n \sum_i^n p(j\ |x_i)\ (x_{il} - \mu_{jl})^T (x_{im} - \mu_{jm})$
        end for
    end for
end while
return $(\{\mu_j\}, \{\sum_j\})$
```

E-step of the algorithm is the probability of putting point *i* to cluster *j* for analogous to the smallest distance and E-step is quite equal to the reassignment step of K-Means. M-step then perfectly recalculates the means of the new clusters and establishing the uniformity of updates. In this case, the EM algorithm for mixtures of Gaussians is likely to the K-Means.

4. Proposed Method

In the proposed algorithm P_Means, the computation of centroids of new means varies from that in the K-Means algorithm. The initial centroids or current best estimate for the parameters are calculated from the log likewise of all data. Therefore, any two or more initial centroids may not be equal or not very close to each other and data will be allocated to the appropriate clusters with closely equal centroids. Here, we negate the formation of an empty cluster. Also, the expectation part (E Step) is used to estimate missing labels to fill in if there is missing data in the datasets. After that step, all the data items are present and can be clustered easily. After getting all the data items completely these are divided into k clusters. Distance between the two data points and the centroid is measured using the Euclidian distance function. The implementation steps of the proposed algorithm to make clusters are similar to those of the original K-Means algorithm. The proposed method is shown in Figure 1.

4.1 Flow chart for the proposed method

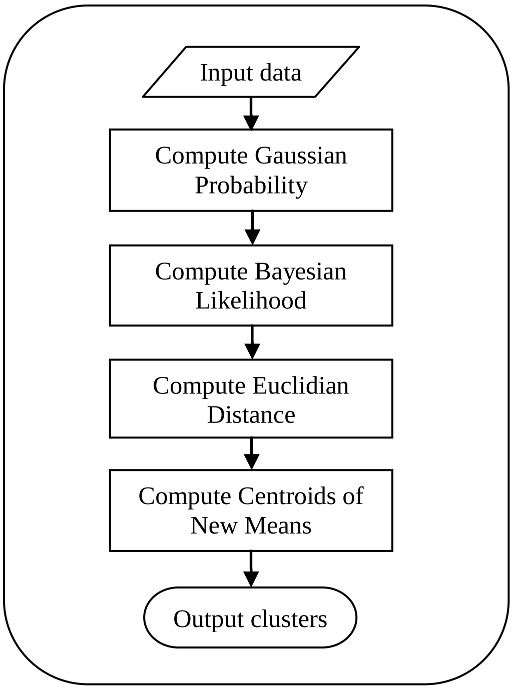
Input data
Compute Gaussian Probability
Compute Bayesian Likelihood
Compute Euclidian Distance
Compute Centroids of New Means
Output clusters

Figure 1. Flow chart of proposed method

4.2 Proposed algorithm

The proposed algorithm will perform the two steps until convergence.

Step 1: In our algorithm Likelihood function, Gaussian probability density function and Bayesian theorem are used to set the values of empty labels and calculate initial center vectors for all data. Also the algorithm computes the probabilities of assigning point *i* to cluster *j* for the one which is the smallest distance.

Step 2: This algorithm calculates the Euclidean distance from each point to each cluster mean and finds the least. Each point is reassigned to the cluster. Then recomputes the mean of each cluster, and updates c_j for every *j*.

4.3 Pseudo code for the proposed algorithm

$Inputs\ to\ the\ algorithm\ are$

$- k\ (the\ number\ of\ centers),$

$X(the\ n\ datapoints\ in\ d\ dimensions),$

$the\ initial\ locations\ of\ the\ centers\ C = \{c_j\}, and$

$the\ initial\ values\ of\ p(j), \{\mu_j\}, \{\sum_j\}.$

$The\ Gaussian\ probability\ density\ function\ is\ p(x_i|\mu_j, \sum_j).$

$The\ term\ p(x_i) obtained\ from\ summing\ over\ j\ the\ values\ p(x_i|\mu_j, \sum_j)\ p(j).$

```
P_Means $(X \in R^{n\times d}, k, C, p(j), \{\mu_j\}, \{\sum_j\})$
while likelihood $L(\{\mu_j\}, \{\sum_j\} \mid X)$ and $c_j$ changes do
    for $i \in \{1, \dots, n\}$ do
        for $j \in \{i, \dots, k\}$ do
            $p(j \mid x_i) \leftarrow p(x_i|\mu_j, \sum_j)\ p(j)\ /\ p(x_i)$ //Bayes Rule
        end for
    end for
    for $i \in \{1, \dots, n\}$ do
        $class(x_i) \leftarrow \arg min_j \parallel x_i - c_j \parallel$
    end for
    for $j \in \{i, \dots, k\}$ do
        $c_j \leftarrow \sum_i I(class(x_i) = j)\ x_i\ /\ \sum_i I(class(x_i) = j)$
    end for
end while
return $(C, \{\mu_j\}, \{\sum_j\})$
```

5. Results and Discussion

Let us consider a 1-dimensionaldata set (17 data objects): 1, 3, 2, 5, 6, 2, 3, 1, 36, 45, 3, -15, 17, 95, 31, -30, and -67. We tested these data objects through P_Means algorithm. It does not form empty cluster where basic K-Means leaves empty clusters as given in Table 1.

In the basic K-Means, complexity may be less sometimes than P_Means. P_Means is a much more efficient, realistic algorithm. The result of the experiment shows that the presented clustering algorithm P_Means can solve the empty cluster problem as shown in Figure 2 and 3.

It has been found that when the number of clusters increases P_Means algorithm can group similar objects into respective clusters. In the case of the P_Means algorithm, when the value of K is 4 or 7 then similar objects are grouped into 4 or 7 clusters respectively and no cluster is empty as shown in Table 1. For this example, K-Means algorithm creates an empty cluster when the value of K is 4 and objects are grouped into 3 clusters. For the P_Means algorithm, the number of similar objects in each cluster is also shown in the graph that when the value of K is 4 then cluster1, cluster2, cluster3 and cluster4 have 11, 3, 2 and 1 number of objects respectively as shown in Figure 3.

Table 1: Comparison of K-Means and P_Means

Cluster	K-Means			P_Means					
	K = 2	K =3	K =4	K =2	K = 3	K = 4	K= 5	K = 6	K = 7
C1	1,3,2,5,6,2,3,1,36,45,3,-15,17,95,31	1,3,2,5,6,2,3,1,-30,3,-15,17	1,3,2,5,6,2,3,1,3,-15,17	1,3,2,5,6,2,3,1,-30,3,-15,17,-67	1,3,2,5,6,2,3,1,3,-15,17	1,3,2,5,6,2,3,1,3,17,31	1,3,2,5,6,2,3,1,3,17	1,3,2,5,6,2,3,1,3,17	1,2,2,1
C2	-30,-67	36,45,95,31	36,45,95,31	36,45,95,31	36,45,95,31	36,45,95	36,45,31	36,45,31	3,5,6,3,3
C3		-67	-30,-67		-30,-67	-30,-45	-30,-15	-30	-30,-15
C4			**empty**			-67	95	-15	36,17,31
C5							-67	95	45
C6								-67	95
C7									-67

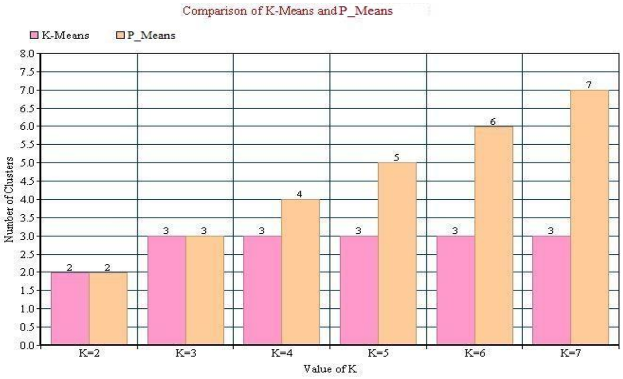

Figure 2. Comparison graph of K-Means and P_Means
Value of k (clusters) Vs No. of output clusters

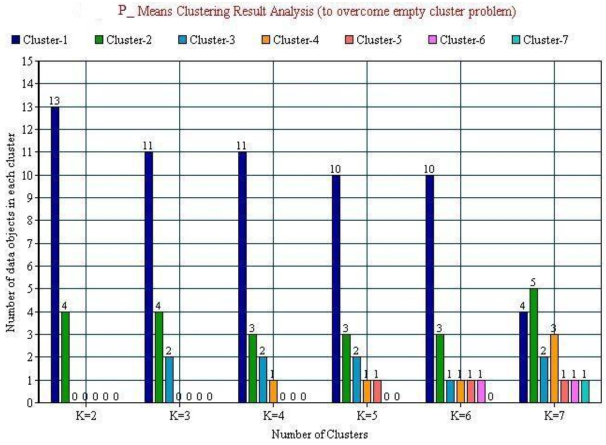

Figure 3. Performance graph of P_Means
Value of k (clusters) Vs No. of objects in each cluster

6. Conclusion and Future Scope

This paper highlights connections among Gaussian mixture models and K-Means clustering algorithms and implements the P_Means algorithm for clustering that retain some benefits of Bayesian parametric, Gaussian mixture model and K-Means algorithm. Although K-Means algorithm is widely used it has been found that the clusters generated are not proper. Hence to overcome these problems of K-Means algorithm, Expectation step and GMM of EM algorithm are added to K-Means algorithm. The proposed approach P_Means algorithm keeps up all important features of the basic K-Means. At the same moment, P_Means removes the possibility of making empty clusters and prevents incomplete data problems by filling in the missing values and giving the best cluster groups, to a great extent, without any significant performance degradation. The proposed P_Means algorithm is applied to 1-dimensional data, its application to higher dimensional data and the quantitative performance measure of the non-empty clusters will be our future work.

Conflict of Interest

There is no conflict of interest for this article.

Funding Source
There is no funding source for this article.

Authors' Contributions

The single author of this article is the only contributor.

Acknowledgements
The author is thankful to his parents, wife, and daughter for their support.

References

[1] MacQueen, J. "Classification and analysis of multivariate observations." 5th Berkeley Symp. Math. Statist. Probability. Los Angeles LA USA: University of California, **1967.**

[2] Reynolds, Douglas A. "Gaussian mixture models." Encyclopedia of biometrics 741, pp.**659-663, 2009.**

[3] Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." Journal of the royal statistical society: series B (methodological) 39.1: pp.**1-22, 1977.**

[4] Bradley, Paul S., and Usama M. Fayyad. "Refining initial points for k-means clustering." ICML. Vol.**98**, **1998.**

[5] Pakhira, Malay K. "A modified k-means algorithm to avoid empty clusters." International Journal of Recent Trends in Engineering 1.1: 220, **2009.**

[6] Yang, Miin-Shen, Chien-Yo Lai, and Chih-Ying Lin. "A robust EM clustering algorithm for Gaussian mixture models." Pattern Recognition 45.11: pp.**3950-3961, 2012.**

[7] McLachlan, Geoffrey J., and Suren Rathnayake. "On the number of components in a Gaussian mixture model." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 4.5: pp.**341-355, 2014.**

[8] Huang, Tao, Heng Peng, and Kun Zhang. "Model selection for Gaussian mixture models." Statistica Sinica: pp.**147-169, 2017.**

[9] Patel, Eva, and Dharmender Singh Kushwaha. "Clustering cloud workloads: K-means vs gaussian mixture model." Procedia Computer Science 171: pp.**158-167, 2020.**

[10] Androniceanu, Armenia, Jani Kinnunen, and Irina Georgescu. "E-Government clusters in the EU based on the Gaussian Mixture

Models." Administratie si Management Public 35: pp.**6-20, 2020.**

[11] Löffler, Matthias, Anderson Y. Zhang, and Harrison H. Zhou. "Optimality of spectral clustering in the Gaussian mixture model." The Annals of Statistics 49.5: pp.**2506-2530, 2021.**

[12] Chen, Yongxin, Tryphon T. Georgiou, and Allen Tannenbaum. "Optimal transport for Gaussian mixture models." IEEE Access 7: pp.**6269-6278, 2018.**

[13] Viroli, Cinzia, and Geoffrey J. McLachlan. "Deep Gaussian mixture models." Statistics and Computing 29: pp.**43-51, 2019.**

[14] Yuan, Wentao, et al. "Deepgmr: Learning latent gaussian mixture models for registration." Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16. Springer International Publishing, 2020.

[15] Shahin, Ismail, Ali Bou Nassif, and Shibani Hamsa. "Emotion recognition using hybrid Gaussian mixture model and deep neural network." IEEE access 7: pp.**26777-26787, 2019.**

[16] Zong, Bo, et al. "Deep autoencoding gaussian mixture model for unsupervised anomaly detection." International conference on learning representations. **2018.**

[17] An, Peng, Zhiyuan Wang, and Chunjiong Zhang. "Ensemble unsupervised autoencoders and Gaussian mixture model for cyberattack detection." Information Processing & Management 59.2 (2022): 102844.

[18] Ding, Nan, et al. "Real-time anomaly detection based on long short-Term memory and Gaussian Mixture Model." Computers & Electrical Engineering 79 (2019): 106458.

[19] Wan, Huan, et al. "A novel Gaussian mixture model for classification." 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). IEEE, 2019.

[20] Fu, Yinlin, et al. "Gaussian mixture model with feature selection: An embedded approach." Computers & Industrial Engineering 152 (2021): 107000.

[21] Singhal, Amit, et al. "Modeling and prediction of COVID-19 pandemic using Gaussian mixture model." Chaos, Solitons & Fractals 138 (2020): 110023.

[22] Zhu, Weiqiang, et al. "Earthquake phase association using a Bayesian Gaussian mixture model." Journal of Geophysical Research: Solid Earth 127.5 (2022): e2021JB023249.

[23] Datta, R. P., and Sanjib Saha. "Applying rule-based classification techniques to medical databases: an empirical study." International Journal of Business Intelligence and Systems Engineering 1.1: pp.**32-48, 2016.**

[24] Das, Subhankar, and Sanjib Saha. "Data mining and soft computing using support vector machine: A survey." International Journal of Computer Applications 77.14, **2013.**

[25] Saha, Sanjib, and Debashis Nandi. "Data Classification based on Decision Tree, Rule Generation, Bayes and Statistical Methods: An Empirical Comparison." Int. J. Comput. Appl 129.7: pp.**36-41, 2015.**

[26] Saha, Sanjib. "Non-rigid Registration of De-noised Ultrasound Breast Tumors in Image Guided Breast-Conserving Surgery." Intelligent Systems and Human Machine Collaboration. Springer, Singapore, pp.**191-206, 2023.**

[27] Saha, Sanjib, et al. "ADU-Net: An Attention Dense U-Net based deep supervised DNN for automated lesion segmentation of COVID-19 from chest CT images." Biomedical Signal Processing and Control 85: 104974, **2023.**

AUTHORS PROFILE

Sanjib Saha earned his Bachelor of Engineering and Master of Technology from Burdwan University and Jadavpur University respectively. He is pursuing PhD in Computer Science and Engineering at National Institute of Technology, Durgapur, India. He is working as an Assistant Professor in Department of Computer Science and Engineering at Dr. B. C. Roy Engineering College, Durgapur. He has published research papers in SCI and Scopus journals including conferences. His main research work focuses on Machine Learning, Deep Learning, and Medical Image Classification, Segmentation & Registration.