Bayesian Learning of Finite Asymmetric Gaussian Mixtures

No Author Given

No Institute Given

Abstract. Asymmetric Gaussian mixture (AGM) model has been proven to be more flexible than the classic Gaussian mixture model from many aspects. In contrast with previous efforts that have focused on maximum likelihood estimation, this paper introduces a fully Bayesian learning approach using Metropolis-Hastings (MH) within Gibbs sampling method to learn AGM model. We show the merits of the proposed model using synthetic data and a challenging intrusion detection application.

Keywords - Asymmetric Gaussian Mixture, Metropolis-Hastings, Gibbs sampling, MCMC, Intrusion detection

1 Introduction

A large volume of data is generated everyday. A crucial task is the analysis and modeling of these data. Many statistical and data mining approaches have been proposed in the past. Among these approaches, finite mixture models have received a lot of attention because they are flexible and powerful probabilistic tools for modeling data [1]. In recent years, there has been an increasing trend of applying finite mixtures into unsupervised learning domains involving statistical modeling of data, such as astronomy, ecology, bioinformatics, pattern recognition, computer vision and machine learning [2]. As an improvement of naive Bayes methodologies, mixture modeling can be viewed as the superimposition of a finite number of component densities which respects the dependency between data groups, bringing more generality and robustness.

As an efficient approach, Gaussian mixture model (GMM) [3] is widely deployed because of its outstanding suitability in several domains such as computer vision, pattern recognition and data mining. In this paper, we choose asymmetric Gaussian mixture (AGM) model [4] for modeling because it uses two variance parameters for left and right parts of each distribution in the mixture which allows to accurately model non-Gaussian datasets including asymmetric ones.

A challenging issue when deploying mixture models is the learning of the model's parameters. The estimation of the parameters of mixture distributions can be accomplished by using maximum-likelihood-based expectation maximization (EM) [5] algorithm. However, EM has some drawbacks such as overfitting and dependency on initialization [6] [7]. Therefore, an alternative is the fully Bayesian approach, based for instance on Markov Chain Monte Carlo (MCMC) methods, which has been found to be useful in many applications by considering parameters priors which can avoid overfitting problems. As a sampling-based learning approach, the main difficulty of MCMC method is that, under some circumstances, direct sampling is not always straightfoward. As widely deployed implementations of MCMC method, Metropolis-Hastings (Hastings, 1970) [8] and Gibbs sampling (Geman and Geman, 1984) [9] methods can be introduced to solve this problem through applying proposal priors and posteriors and sampling one parameter by giving the others. By combining the advantages of both sampling techniques together, the Metopolis-Hastings within Gibbs method [6] is selected as the learning algorithm for AGM model.

The rest of this paper is organized as follows. Section 2 illustrates the AGM model and its Bayesian learning process. Section 3 is devoted to experimental results using both synthetic data and a real application (network intrusion detection). Finally, Section 4 concludes the paper.

2 Bayesian Model

2.1 Asymmetric Gaussian Mixture Model

Assuming that the AGM model has M components then the likelihood function (Elguebaly and Bouguila, 2013) [4] is defined as follows:

$$p(\mathcal{X}|\Theta) = \prod_{i=i}^{N} \sum_{j=1}^{M} p_j p(X_i|\xi_j) \tag{1}$$

where $\mathcal{X} = (X_1, ..., X_N)$ is the set of N observations, $\Theta = \{p_1, ..., p_M, \xi_1, ..., \xi_M\}$ represents the parameters set, p_j ($0 < p_j \leq 1$ and $\sum_{j=1}^{M} p_j = 1$) is the weight for each component in the mixture model and ξ_j is the AGD parameters of mixture component j. Giving $X = (x_1, ..., x_d)$, the probability density function (Elguebaly and Bouguila, 2013) [4] can be defined as follows:

$$p(X|\xi_j) \propto \prod_{k=1}^{d} \frac{1}{(\sigma_{l_{jk}} + \sigma_{r_{jk}})} \times \begin{cases} \exp\left[-\frac{(x_k-\mu_{jk})^2}{2(\sigma_{l_{jk}})^2}\right] & if\ x_k\ <\ \mu_{jk} \\ \exp\left[-\frac{(x_k-\mu_{jk})^2}{2(\sigma_{r_{jk}})^2}\right] & if\ x_k\ \geqslant\ \mu_{jk} \end{cases} \tag{2}$$

where $\xi_j = (\mu_j, \sigma_{lj}, \sigma_{rj})$ is the set of parameters of component j and $\mu_j = (\mu_{j1}, ..., \mu_{jd})$ is the mean, $\sigma_{lj} = (\sigma_{lj1}, ..., \sigma_{ljd})$ and $\sigma_{rj} = (\sigma_{rj1}, ..., \sigma_{rjd})$ are the left and right standard deviations for AGD. To be more specific, $x_k \sim N(\mu_{jk}, \sigma_{ljk})$ $(x_k < \mu_{jk})$ and $x_k \sim N(\mu_{jk}, \sigma_{rjk})$ $(x_k \geqslant \mu_{jk})$ for each dimension.

In order to simplify the Bayesian learning process, we introduce a M-dimensional membership vector Z for each observation $X_i, 1 < i < N, Z_i = (Z_{i1}, ..., Z_{iM})$ which indicates to which specific component X_i belongs (Bouguila, Ziou and Monga, 2006) [2], such that:

$$Z_{ij} = \begin{cases} 1 & \text{if } X_i \text{ belongs to component } j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

in other words, $Z_{ij} = 1$ only if observation X_i has the highest probability of belonging to component j and accordingly, for other components, $Z_{ij} = 0$.

By combining Eq. (1) and Eq. (3) together we derive the complete likelihood function:

$$p(\mathcal{X}, Z|\Theta) = \prod_{i=1}^{N} \prod_{j=1}^{M} (p_j p(X_i|\xi_j))^{Z_{ij}} \tag{4}$$

2.2 Learning Algorithm

Before describing MH-within-Gibbs learning steps, the priors and posteriors need to be specified. First, we denote the postorior probability of membership vector Z as $\pi(Z|\Theta, \mathcal{X})$ (Elguebaly and Bouguila, 2011) [10]:

$$Z^{(t)} \sim \pi(Z|\Theta^{(t-1)}, \mathcal{X}) \tag{5}$$

the number of observations belonging to a specific component j can be calculated using $Z^{(t)}$ as follows:

$$n_j^{(t)} = \sum_{i=1}^{N} Z_{ij} \ (j = 1, ..., M) \tag{6}$$

thus $n^{(t)} = (n_i^{(t)}, ..., n_M^{(t)})$ represents the number of observations belonging to each mixture component.

Since the mixture weight p_j satisfies the following conditions ($0 < p_j \leq 1$ and $\sum_{j=1}^{M} p_j = 1$), a natural choice of the prior is Dirichlet distribution as follows [11]:

$$\pi(p_j^{(t)}) \sim \mathcal{D}(\gamma_1, ..., \gamma_M) \tag{7}$$

where γ_j is known hyperparameter. Consequently, the posterior of the mixture weight p_j is:

$$p(p_j^{(t)}|Z^{(t)}) \sim \mathcal{D}(\gamma_1 + n_1^{(t)}, ..., \gamma_M + n_M^{(t)}) \tag{8}$$

Direct sampling of mixture parameters $\xi \sim p(\xi|Z, \mathcal{X})$ could be difficult so Metropolis-Hastings method should be deployed using proposal distributions for $\xi^{(t)} \sim q(\xi|\xi^{(t-1)})$. To be more specific, for parameters of AGM model which are μ , σ_l and σ_r, we choose proposal distributions as follows:

$$\mu_j^{(t)} \sim \mathcal{N}_d(\mu_j^{(t-1)}, \Sigma) \tag{9}$$

$$\sigma_{lj}^{(t)} \sim \mathcal{N}_d(\sigma_{lj}^{(t-1)}, \Sigma) \tag{10}$$

$$\sigma_{rj}^{(t)} \sim \mathcal{N}_d(\sigma_{rj}^{(t-1)}, \Sigma) \tag{11}$$

the proposal distributions are d-dimensional Gaussian distributions with Σ as d x d identity matrix which makes the sampling a random walk MCMC process.

As the most important part of Metropolis-Hastings method, at the end of each iteration, for new generated mixture parameter set $\Theta^{(t)}$, an acceptance ratio r needs to be calculated in order to make a decision whether they should be accepted or discarded for the next iteration. The acceptance ratio r is given by:

$$r = \frac{p(\mathcal{X}|\Theta^{(t)})\pi(\Theta^{(t)})q(\Theta^{(t-1)}|\Theta^{(t)})}{p(\mathcal{X}|\Theta^{(t-1)})\pi(\Theta^{(t-1)})q(\Theta^{(t)}|\Theta^{(t-1)})} \tag{12}$$

where $\pi(\Theta)$ is the proposed prior distribution which can be decomposed to d-dimensional Gaussian distributions such that $\mu \sim \mathcal{N}_d(\eta, \Sigma)$ and $\sigma_l, \sigma_r \sim \mathcal{N}_d(\tau, \Sigma)$ given known hyperparameters η and τ. Since mixture weight p has been computed previously during the Gibbs sampling part, it should not be included in Eq. (12). Further information about the calculation of acceptance ratio r is explained in Appendix A.

Once acceptance ratio r is derived by Eq. (15), we compute acceptance probability $\alpha = min[1, r]$ [12]. Then $u \sim U_{[0,1]}$ is supposed to be generated randomly. If $\alpha < u$, the proposed move should be accepted and parameters should be updated by $p^{(t)}$ and $\xi^{(t)}$ for next iteration. Otherwise, we discard $p^{(t)}$, $\xi^{(t)}$ and set $p^{(t)} = p^{(t-1)}$, $\xi^{(t)} = \xi^{(t-1)}$.

We summarize the MH-within-Gibbs learning process for AGM model in the following steps:

Input: Data observations $\mathcal{X}$ and components number M
Output: AGM mixture parameter set Θ

1. Initialization
2. Step t: For $t = 1, \dots$

   Gibbs sampling part

   (a) Generate $Z^{(t)}$ from Eq. (5)
   (b) Compute $n_j^{(t)}$ from Eq. (6)
   (c) Generate $p_j^{(t)}$ from Eq. (8)

   Metropolis-Hastings part

   (d) Sample $\xi_j^{(t)}$ $(\mu_j^{(t)}, \sigma_{lj}^{(t)}, \sigma_{rj}^{(t)})$ from Eqs. (9) (10) (11)
   (e) Compute acceptance ratio r from Eq. (15)
   (f) Generate $\alpha = min[1, r]$ and $u \sim U_{[0,1]}$
   (g) If $\alpha \geq u$ then $\xi^{(t)} = \xi^{(t-1)}$

3 Experimental Results

3.1 Design of Experiments

We apply the AGM model to both synthetic data and intrusion detection. For synthetic data validation, testing observations will be generated from AGM with known components number M and experimental results will be evaluated by comparing the estimated and actual mixture parameters. In intrusion detection application, we select NSL-KDD dataset [13] as testing database. K-means algorithm is used for initialization and the results analysis will be based on statistics derived from confusion matrix.

3.2 Synthetic Data

The main goals of this section are feasibility analysis and efficiency evaluation of the AGM learning algorithm. Observations number is set to 300 grouped into two clusters ($M = 2$). Hyperparameters are set to $\gamma_j = 1$ [14] for sampling mixture weight p_j from Eq. (8). η and τ are considered as d-dimensional zero vectors in prior distributions of mixture parameter ξ.

Different proposed component numbers ($M' = 1, \ldots, 5$) are tested during the AGM learning process and the statistics are summarized in Table 1. In order to select the best number of components, we consider marginal likelihood as described in [6]. The probability density functions are plotted for both original and estimated AGM components and the polylines show the trace of accepted moves for each component.

In terms of the best fit result, the accuracy is evaluated by calculating the Euclidean distance between original and estimated mixture parameter sets ξ and $\hat{\xi}$ (Table 2). In summary, the estimation of mean is accurate because the Euclidean distance between μ_j and $\hat{\mu}_j$ is small but the distance between standard deviation σ_{lj}, σ_{rj} and $\hat{\sigma}_{lj}, \hat{\sigma}_{rj}$ is slightly significant. However, this difference has not affected the clustering result.

Table 1. AGM Learning Statistics

Component number M'	***Moves accepted***	***Acceptance ratio***	***Marginal likelihood***
1	22	7.33%	-1596.143
2	11	3.67%	-1500.370
3	14	4.67%	-1684.518
4	63	21.00%	-1522.148
5	39	13.00%	-1517.533

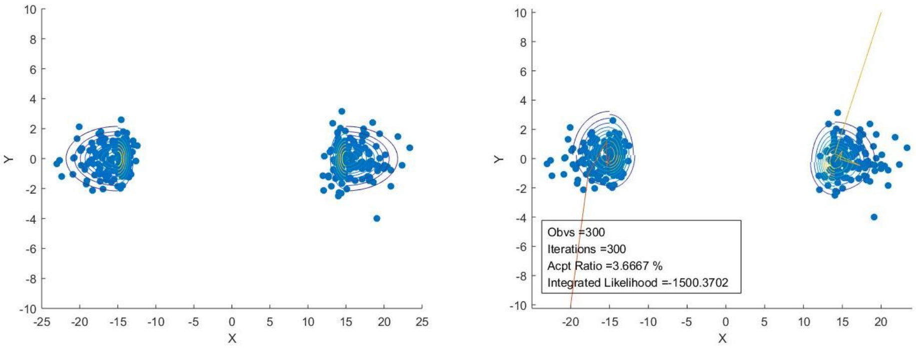

Fig. 1. Original synthetic observations and learning result ($M' = M = 2$)

Table 2. Accuracy Analysis ($M' = M = 2$)

Component number $j = 1$	**Mean** (μ_j)	**Left standard deviation** (σ_{lj})	**Right standard deviation** (σ_{rj})
ξ	[-15.00, 0.00]	[10.00, 1.00]	[1.00, 1.00]
$\hat{\xi}$	[-14.99, 0.25]	[4.77, 1.13]	[2.31, 1.88]
Euclidean Distance	0.246	5.236	1.581
Component number $j = 2$	**Mean** (μ_j)	**Left standard deviation** (σ_{lj})	**Right standard deviation** (σ_{rj})
ξ	[15.00, 0.00]	[1.00, 1.00]	[10.00, 1.00]
$\hat{\xi}$	[14.02, -0.24]	[2.04, 1.04]	[5.70, 1.59]
Euclidean Distance	1.010	1.036	4.338

3.3 Intrusion Detection

Along with the development of information-based industries, network security problems are becoming increasingly important today. In order to address this challenge, many data mining methodologies were proposed including both classification-based [15] and clustering-based [16] ones. However, classification-based solutions generally perform ineffectively for dynamic and variate attacking methods because changes of the intrusion patterns cannot be automatically adapted by supervised learning algorithms. Consequently, unsupervised approach such as AGM model is more favorable for modern intrusion-detection.

We select NSL-KDD [13], an improved KDDCUP'99 intrusion-detection dataset, as the testing target since redundant records have been removed from original dataset to avoid potential learning bias. Before applying the testing models onto the dataset, the data pre-processing is needed since discrete enumerated values must be translated to numerical ones and be normalized properly to lead an accurate result. Therefore, we substitute enumerated values with their numbers of occurrences which could reflect the density distribution of discrete values. Having all numerical data in hand, we apply feature scaling method to normalize numerical values between 0 to 1 as follows:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{13}$$

where x and x' denote original and normalized values. In this way we could use unified proposal distribution for every dimension with the same value of hyperparameter Σ during random walk MCMC sampling step (Table 3).

K-means clustering algorithm [17] is chosen for the comparison of accuracy. Testing data records with total amount of 25192 (20% of NSL-KDD dataset) are clustered into two groups with 11743 intrusions and 13449 normal behaviors indicating components number $M' = 2$. In order to better evaluate the pros and cons of models, results derived from Gaussian mixture model (GMM) will also be taken into consideration. The comparison based on confusion matrices resulted from K-means, GMM and AGM model (Table 4) reveals the fact that based on a less accurate initialization given by K-means (60.85%), GMM performs almost the same way as K-means and the difference between these two models is trivial. In contrast, AGM model makes a significant improvement with much higher accuracy rate (80.47%) and precision percentage (96.86%), while much lower false positive rate (4.26%) illustrating AGM model is capable of effectively detecting intrusions from background noises. Compared with K-means and GMM, AGM model has a higher false negative rate (28.58%) which means it tends to strictly identify normal behaviors as intrusions which could be mitigated by reducing dimensions of dataset using feature selection methodologies.

Table 3. Translation and Normalization of Internet Protocols (Enumerated Values)

Internet Protocols	***Number of Occurrences***	***Normalized Values***
ICMP	1655	0
UDP	3011	0.071867
TCP	20526	1

Table 4. Confusion Matrices and Statistics of K-means, GMM and AGM Models

K-means

	NF [a]	***F*** [b]
NF	2445	9298
F	565	12884

GMM

	NF	***F***
NF	2464	9279
F	584	12865

AGM

	NF	***F***
NF	11484	259
F	5621	7828

	K-means	***GMM***	***AGM***
Accuracy	60.85%	60.85%	76.66%
Precision	20.82%	20.98%	97.79%
False Positive Rate	41.92%	41.90%	3.20%
False Negative Rate	18.77%	19.16%	32.86%

[a] Non fault-prone, [b] Fault-prone.

4 Conclusion and Future Work

This paper illustrated a new intrusion detection approach by applying asymmetric Gaussian mixtures with a fully Bayesian learning process which is achieved by applying a hybrid sampling-based MH-within-Gibbs learning algorithm. According to the experiment results, the AGM model is proved as an effective approach for clustering. In spite of the advantages of AGM we mentioned above, some improvements are still needed to promote the accuracy and flexibility and mitigate the drawbacks. Therefore, we plan to extend the Bayesian learning process and introduce model selection and feature selection methodologies to improve the performance in the case of high-dimensional datasets.

Appendix A

4.1 Derivation of Acceptance Ratio r by Eq. (12)

The derivation of acceptance ratio r is based on the assumption that mixture parameters are independent from each other which means that:

$$\begin{aligned}\pi(\Theta) &= \pi(p,\xi) = \pi(\xi) \\ &= \prod_{j=1}^{M} \pi(\mu_j)\pi(\sigma_{lj})\pi(\sigma_{rj}) \\ &= \prod_{j=1}^{M} \mathcal{N}_d(\mu_j|\eta,\Sigma)\mathcal{N}_d(\sigma_{lj}|\tau,\Sigma)\mathcal{N}_d(\sigma_{rj}|\tau,\Sigma) \end{aligned} \tag{14}$$

in Eq. (14), since the mixture weigh p is generated following Gibbs sampling method whose acceptance ratio is always 1, it should be excluded from Metropolis-Hastings estimation step. Accordingly, apply the same rule to the proposal distribution as well:

$$\begin{aligned} q(\Theta^{(t)}|\Theta^{(t-1)}) &= q(\xi^{(t)}|\xi^{(t-1)}) \\ &= \prod_{j=1}^{M} \mathcal{N}_d(\mu_j^{(t)}|\mu_j^{(t-1)},\Sigma)\mathcal{N}_d(\sigma_{lj}^{(t)}|\sigma_{lj}^{(t-1)},\Sigma)\mathcal{N}_d(\sigma_{rj}^{(t)}|\sigma_{rj}^{(t-1)},\Sigma) \end{aligned} \tag{15}$$

by combining Eqs. (2) (4) (9) (10) (11) (14) and (15), equation (12) can be written as follows:

$$\begin{aligned} r &= \frac{p(\mathcal{X}|\Theta^{(t)})\pi(\Theta^{(t)})q(\Theta^{(t-1)}|\Theta^{(t)})}{p(\mathcal{X}|\Theta^{(t-1)})\pi(\Theta^{(t-1)})q(\Theta^{(t)}|\Theta^{(t-1)})} \\ &= \prod_{i=i}^{N}\prod_{j=1}^{M}\Big(\frac{p(X_i|\mu_j^{(t)},\sigma_{lj}^{(t)},\sigma_{rj}^{(t)})}{p(X_i|\mu_j^{(t-1)},\sigma_{lj}^{(t-1)},\sigma_{rj}^{(t-1)})} \\ &\times \frac{\mathcal{N}_d(\mu_j^{(t)}|\eta,\Sigma)\mathcal{N}_d(\sigma_{lj}^{(t)}|\tau,\Sigma)\mathcal{N}_d(\sigma_{rj}^{(t)}|\tau,\Sigma)}{\mathcal{N}_d(\mu_j^{(t-1)}|\eta,\Sigma)\mathcal{N}_d(\sigma_{lj}^{(t-1)}|\tau,\Sigma)\mathcal{N}_d(\sigma_{rj}^{(t-1)}|\tau,\Sigma)} \\ &\times \frac{\mathcal{N}_d(\mu_j^{(t-1)}|\mu_j^{(t)},\Sigma)\mathcal{N}_d(\sigma_{lj}^{(t-1)}|\sigma_{lj}^{(t)},\Sigma)\mathcal{N}_d(\sigma_{rj}^{(t-1)}|\sigma_{rj}^{(t)},\Sigma)}{\mathcal{N}_d(\mu_j^{(t)}|\mu_j^{(t-1)},\Sigma)\mathcal{N}_d(\sigma_{lj}^{(t)}|\sigma_{lj}^{(t-1)},\Sigma)\mathcal{N}_d(\sigma_{rj}^{(t)}|\sigma_{rj}^{(t-1)},\Sigma)} \end{aligned} \tag{16}$$

References

1. McLachlan, G. and Peel, D. (2000). Finite mixture models. New York: Wiley.
2. Bouguila, N., Ziou, D. and Monga, E. (2006). Practical Bayesian estimation of a finite beta mixture through gibbs sampling and its applications. Statistics and Computing, 16(2), pp.215-225.
3. Richardson, S. and Green, P. (1997). On Bayesian Analysis of Mixtures with an Unknown Number of Components (with discussion). Journal of the Royal Statistical Society: Series B (Statistical Methodology), 59(4), pp.731-792.
4. Elguebaly, T. and Bouguila, N. (2013). Background subtraction using finite mixtures of asymmetric Gaussian distributions and shadow detection. Machine Vision and Applications, 25(5), pp.1145-1162.

5. Dempster, A.P., Laird, N.M. and Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society. Series B (methodological), pp.1-38.
6. Bouguila, N., Ziou, D. and Hammoud, R. (2008). On Bayesian analysis of a finite generalized Dirichlet mixture via a Metropolis-within-Gibbs sampling. Pattern Analysis and Applications, 12(2), pp.151-166.
7. Bouguila, N. and Elguebaly, T. (2012). A fully Bayesian model based on reversible jump MCMC and finite Beta mixtures for clustering. Expert Systems with Applications, 39(5), pp.5946-5959.
8. Hastings, W. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika, 57(1), p.97.
9. Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6(6), pp.721-741.
10. Elguebaly, T. and Bouguila, N. (2011). Bayesian learning of finite generalized Gaussian mixture models on images. Signal Processing, 91(4), pp.801-820.
11. Marin, J.M. and Mengersen, K.R.C., 2005. Handbook of Statistics: Bayesian modelling and inference on mixtures of distributions, Vol. 25.
12. Luengo, D. and Martino, L., 2013, May. Fully adaptive gaussian mixture metropolis-hastings algorithm. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on (pp. 6148-6152). IEEE.
13. Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A.A., 2009, July. A detailed analysis of the KDD CUP 99 data set. In Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on (pp. 1-6). IEEE. Vancouver
14. Stephens, M., 2000. Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods. Annals of statistics, pp.40-74.
15. Puttini, R.S., Marrakchi, Z. and M, L., 2003, March. A Bayesian Classification Model for RealTime Intrusion Detection. In AIP Conference Proceedings (Vol. 659, No. 1, pp. 150-162). AIP.
16. Zhong, S., Khoshgoftaar, T.M. and Seliya, N., 2007. Clustering-based network intrusion detection. International Journal of reliability, Quality and safety Engineering, 14(02), pp.169-187.
17. Hartigan, J.A. and Wong, M.A., 1979. Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), pp.100-108.