Noname manuscript No.
(will be inserted by the editor)

Insert your title here

If you have a subtitle, write it here

Shuai Fu · Nizar Bouguila

Received: date / Accepted: date

Abstract A novel unsupervised Bayesian learning framework based on asymmetric Gaussian mixture (AGM) statistical model is proposed since AGM is shown to be more effective compared to the classic Gaussian mixture. The Bayesian learning framework is developed by adopting sampling-based Markov chain Monte Carlo (MCMC) methodology. More precisely, the fundamental learning algorithm is a hybrid Metropolis-Hastings within Gibbs sampling solution which is integrated within a reversible jump MCMC (RJMCMC) learning framework, a self-adapted sampling-based MCMC implementation, that enables model transfer throughout the mixture parameters learning process, therefore, automatically converges to the optimal number of data groups. Furthermore, considering the involvement of high-dimensional visual datasets, a dimensionality reduction algorithm based on mixtures of distributions is included to tackle the irrelevant and extraneous features. The performance comparison between AGM and other popular solutions is given and both synthetic and real data sets extracted from challenging applications such as intrusion detection, spam filtering and image categorization are evaluated to show the merits of the proposed approach.

Keywords Asymmetric Gaussian Mixture · RJMCMC · Intrusion Detection · Spam Filtering · Image Categorization · Dimensionality Reduction

Shuai Fu
Concordia Institute for Information Systems Engineering
Concordia University, Montreal, Canada
E-mail: f_shuai@encs.concordia.ca

Nizar Bouguila
Concordia Institute for Information Systems Engineering
Concordia University, Montreal, Canada
E-mail: nizar.bouguila@concordia.ca

1 Introduction

Over past decades, many statistical data mining approaches have been proposed to address challenging data modeling analysis problems given the fact that the volume of data is dramatically increasing due to the usage of Internet. Meanwhile, modern machine-learning-based techniques perform in both generative and discriminative ways which can be divided into two main streams, classification-based supervised and clustering-based unsupervised ones. Compared to supervised solutions, unsupervised approach has no assumption on the number of groups, therefore, friendly to newly added data and patterns which makes it more suitable for increasing database analysis. Moreover, it also immunizes against learning biases and overfitting problems that commonly exist in most supervised approaches if model training is inappropriate. Consequently, there has been an increasing trend of applying finite mixtures into different domains involving statistical modeling of data, such as astronomy, ecology, bioinformatics, pattern recognition, computer vision and machine learning [10]. Our work is based on asymmetric Gaussian mixture (AGM) model [19] and reversible jump Markov chain Monte Carlo (RJMCMC) learning algorithm [33]. Previous efforts reveal the fact that AGM outperforms classic Gaussian mixture model (GMM) by taking asymmetric datasets into consideration which provides more flexibility [18]. Our RJMCMC implementation is based on a hybrid sampling-based approach which takes advantages of both Metropolis-Hastings (MH) and Gibbs sampling methods [17], therefore, simplifies mathematical complexity and extends adaptability of the model. Moreover, without giving a fixed components number in advance, RJMCMC applies a dynamic data-based strategy to identify the optimal components number throughout iterations which makes the model learning a self-adaptive process. To achieve better fitting outcomes, feature selection process is involved to handle high-dimensional vectors of features and the analysis and discussion of deploying AGM to both synthetic datasets and real applications is given in the later chapters.

1.1 Finite Mixture Models

As upgrade of single-mathematical-model-based methodologies, mixture models [38,37,4] can be seen as a superimposition of certain mixture components sharing dependencies with each other, therefore, lead to outstanding performance especially for high-dimensional and multi-cluster datasets. Finite mixture models can be described by

$$p(X|\Theta) = \sum_{j=1}^{M} p_j p(X|\Theta_j) \tag{1}$$

where X reprensents a vector in a given dataset and Θ defines the mixture parameters set (for each mixture compoent, the sub-parameter set is described

by $\Theta_j, j = 1, \ldots, M$) as well as component weight p_j ($0 < p_j \leq 1$ and $\sum_{j=1}^{M} p_j = 1$).

1.2 Finite Mixture Models

Probability density function (PDF) selection has an important role in finite mixture model because it significantly affects the capability of representing the data. Improper PDF selection will cause incorrect outcomes such as wrong components number and poor data fitting. Gaussian mixture model (GMM) [33] demonstrated satisfactory fitting abilities on most real applications whose datasets are Gaussian-like. However, under more general circumstances regarding to non-Gaussian or asymmetric datasets, asymmetric Gaussian mixture (AGM) model [19] leads to a better accuracy by introducing two variance parameters for both left and right parts of asymmetric Gaussian distribution, providing more flexibility for variant real applications. Therefore, the justification of choosing AGM model and its merits will be discussed in the following chapters.

1.3 Bayesian Learning Framework

Estimating the parameters of mixture models could be a challenging task. The maximum-likelihood-based expectation maximization (EM) [15] algorithm is one of the most popular parameter learning approaches. However, the disadvantages of EM algorithm are also obvious. Given the fact that EM approximates values of mixture parameters in a deterministic way this could cause slow convergence and compromise the usability of the algorithm. Furthermore, bad initialization and overfitting problems [9, 5] will also significantly affect its accuracy. Therefore, fully Bayesian learning algorithms, such as Markov Chain Monte Carlo (MCMC) based implementations, are found to be useful to eliminate overfitting problems in mixture parameter learning by introducing prior and posterior distributions for mixture parameters. In our work, the learning process is accomplished by a hybrid MCMC algorithm, which is well known as Metropolis-Hastings within Gibbs sampling [9, 11], based on both Metropolis-Hastings [23] and Gibbs sampling [21] methods because the main difficulty of classic MCMC method is that, under some circumstances, direct sampling is not always straightforward. Moreover, we reinforce the learning algorithm by introducing reversible jump MCMC (RJMCMC) [33] methodology to increase the flexibility of AGM model by allowing model transfer throughout iterations via increasing (component birth/split step) and decreasing (component death/merge step) mixture components. Because of the stochastic sampling-based learning process, learning iterations could end up with different number of components so we choose marginal likelihood [9] to perform model selection in order to evaluate fitting results between models.

1.4 Dimensionality Reduction

One of the most important tasks in data mining, pattern recognition, computer vision and machine learning applications is that, the existence of outliers and irrelevant features severely compromises the clustering outcomes. Therefore, many dimensionality reduction methodologies have been proposed [8,12] such as feature extraction and selection which try to remove these unneeded features in order to improve the performance of the modeling [32,25] while feature extraction is based on transformations or combinations of the original features [30]. Indeed, feature selection methods identify relevant features in the original representation space. Recently, a volume of literature [36,16] has shown that selecting relevant features leads to more accurate modeling results. However, this problem is not trivial especially in the unsupervised context dealing with labelless data sets. For this reason, previous researches [27,18,20] were devoted to extend unsupervised feature selection to mixture-based clustering. In this article, we extended the RJMCMC-based simultaneous Bayesian clustering and feature selection approach proposed in [18] to asymmetric Gaussian mixture model in order to improve the modeling performance on a challenging image categorization application.

1.5 Overview

The rest of this article is organized as follows:

Chapter 2 introduces the Asymmetric Gaussian mixture model and its sampling based Bayesian learning framework. In particular, a self-adapted reversible jump MCMC implementation which has no assumption concerning the number of components and, therefore, the AGM model itself could be transferred between iterations. Furthermore the self-adapted learning process treats components number as an extra parameter and adjusts it throughout iterations by automatically increasing (component birth/death step) and decreasing (component merge/split step) according to current status, therefore, enables model transfer which significantly improves the learning performance.

Chapter 3 is devoted to feature selection since the AGM model assumes that all the features of observations have the same weight of importance and carry pertinent information which is not always the case and many of those features can be irrelevant for clustering purpose. In order to tackle this problem and define relevance and importance of features, feature selection techniques should be taken into consideration. A challenging UIUC sports event database is selected for validation of the proposed approach.

Chapter 4 concludes and summarizes the article and points out future research directions.

2 Asymmetric Gaussian Mixtures with Reversible Jump MCMC and Applications

This chapter presents a novel intrusion detection classifier based on asymmetric Gaussian mixture (AGM) model and reversible jump Markov chain Monte Carlo (RJMCMC) learning algorithm. Previous efforts reveal the fact that AGM outperforms classic Gaussian mixture model (GMM) by taking asymmetric datasets into consideration which provides more flexibility. Our RJMCMC implementation is based on a hybrid sampling-based approach which takes advantages of both Metropolis-Hastings (MH) and Gibbs sampling methods, therefore, simplifies mathematical complexity and extends adaptability of the model. Moreover, without giving a fixed components number in advance, RJMCMC applies a dynamic data-based strategy to identify the optimal components number throughout iterations which makes the model learning a self-adaptive process. Since the model is nondeterministic, Laplace approximation based marginal likelihood is calculated for multiple runs as model selection procedure to improve the correctness and fitting accuracy. Both synthetic and real datasets are applied to our model to discover its merits and the test results will be evaluated and compared with other popular solutions.

2.1 Asymmetric Gaussian Mixture Model

The likelihood function of AGM model [19] with M mixture components can be illustrated as follows:

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N}\sum_{j=1}^{M} p_j p(X_i|\xi_j) \tag{2}$$

where $\mathcal{X} = (X_1, ..., X_N)$ reprensents the dataset with N observations, $\Theta = \{p_1, ..., p_M, \xi_1, ..., \xi_M\}$ defines the mixture parameters set of AGM mixture model including component weight p_j ($0 < p_j \leq 1$ and $\sum_{j=1}^{M} p_j = 1$) and asymmetric Gaussian distribution (AGD) parameters set ξ_j for mixture component j. Assuming the dataset $\mathcal{X}$ is d-dimensional, for each observation $X_n = (x_{n1}, ..., x_{nd}) \in \mathcal{X}$, the probability density function [19] for j-th component of the model can be defined as follows:

$$p(X|\xi_j) \propto \prod_{k=1}^{d} \frac{1}{(\sigma_{l_{jk}} + \sigma_{r_{jk}})} \times \begin{cases} \exp\left[-\frac{(x_k-\mu_{jk})^2}{2(\sigma_{l_{jk}})^2}\right] & if\ x_k < \mu_{jk} \\ \exp\left[-\frac{(x_k-\mu_{jk})^2}{2(\sigma_{r_{jk}})^2}\right] & if\ x_k \geqslant \mu_{jk} \end{cases} \tag{3}$$

parameters set of component j is $\xi_j = (\mu_j, \sigma_{lj}, \sigma_{rj})$ where $\mu_j = (\mu_{j1}, ..., \mu_{jd})$ is the mean, $\sigma_{lj} = (\sigma_{lj1}, ..., \sigma_{ljd})$ and $\sigma_{rj} = (\sigma_{rj1}, ..., \sigma_{rjd})$ represents the left and right standard deviation vectors of AGD .

We bring a M-dimensional membership vector Z to each observation $X_i \in \mathcal{X}, Z_i = (Z_{i1}, ..., Z_{iM})$, indicating which specific component X_i belongs to [10], such that:

$$Z_{ij} = \begin{cases} 1 & \text{if } X_i \text{ belongs to component } j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

that being said, $Z_{ij} = 1$ only when observation X_i has the highest probability of belonging to component j and accordingly, for other components, $Z_{ij} = 0$.

Hence, the complete likelihood function can be obtained by combining Eq. (2) and Eq. (4) as follows:

$$p(\mathcal{X}, Z|\Theta) = \prod_{i=1}^{N}\prod_{j=1}^{M}(p_j p(X_i|\xi_j))^{Z_{ij}} \tag{5}$$

2.2 Bayesian 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 $\pi(Z|\Theta, \mathcal{X})$ [17]:

$$Z \sim \pi(Z|\Theta^{(t-1)}, \mathcal{X}) \tag{6}$$

the number of observations belonging to a specific component j can be calculated using Z as follows:

$$n_j^{(t-1)} = \sum_{i=1}^{N} Z_{ij} \ (j = 1, ..., M) \tag{7}$$

thus $n^{(t-1)} = (n_1^{(t-1)}, ..., n_M^{(t-1)})$ 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 [7,6]

$$\pi(p_1, \ldots, p_M) \sim \mathcal{D}(\gamma_1, ..., \gamma_M) \tag{8}$$

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

$$p(p_1, \ldots, p_M|Z) \sim \mathcal{D}(\gamma_1 + n_1^{(t-1)}, ..., \gamma_M + n_M^{(t-1)}) \tag{9}$$

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:

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

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

$$\sigma_{rj}^{(t)} \sim \mathcal{N}_d(\sigma_{rj}^{(t-1)}, \Sigma) \tag{12}$$

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:

$$r = \frac{p(\mathcal{X}|\Theta^{(t)})\pi(\Theta^{(t)})q(\Theta^{(t-1)}|\Theta^{(t)})}{p(\mathcal{X}|\Theta^{(t-1)})\pi(\Theta^{(t-1)})q(\Theta^{(t)}|\Theta^{(t-1)})} \tag{13}$$

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. 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. (3) (5) (10) (11) (12) (14) and (15), equation (13) can be written as follows:

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

Once acceptance ratio r is derived by Eq. (16), we compute acceptance probability $\alpha = min[1, r]$ [29]. 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 $t = 1, \ldots$
   Gibbs sampling part
   (a) Generate $Z^{(t)}$ from Eq. (6)
   (b) Compute $n_j^{(t)}$ from Eq. (7)
   (c) Generate $p_j^{(t)}$ from Eq. (9)
   Metropolis-Hastings part
   (d) Sample $\xi_j^{(t)}$ $(\mu_j^{(t)}, \sigma_{lj}^{(t)}, \sigma_{rj}^{(t)})$ from Eqs. (10) (11) (12)
   (e) Compute acceptance ratio r from Eq. (13)
   (f) Generate $\alpha = min[1, r]$ and $u \sim U_{[0,1]}$
   (g) If $u \geq \alpha$ then $\xi^{(t)} = \xi^{(t-1)}$

2.3 Reversible Jump Markov Chain Monte Carlo

We reinforce the learning algorithm by introducing reversible jump MCMC (RJMCMC) [5] methodology to increase the flexibility of AGM model because traditional MH-within-Gibbs algorithm assumes that the component number M is given and persistent throughout the learning process. However, because of bad initialization or just information leakage, M could be inaccurate or unknown. Under these circumstances, RJMCMC algorithm presents its merits

by providing extra four independent steps (birth/death steps and merge/split steps) into learning process which could change component number M, therefore, brings more generalities.

In practice, within every RJMCMC learning iteration, the current component number m is considered as an extra parameter which has a proposed Poisson prior $\mathcal{P}(\lambda)$ with $\lambda = 4$ particularly in our case [33]. Accordingly, let M_{min} and M_{max} denote the minimum and maximum number of components M, and assume the probabilities of performing birth/split and death/merge steps are b_m and $d_m = 1 - b_m$ for $m = M_{min}, \ldots, M_{max}$ respectively. Obviously, $b_{M_{max}} = 0$ and $d_{M_{min}} = 0$. Correspondingly, $d_{M_{max}} = 1 - b_{M_{max}} = 1$ and $b_{M_{min}} = 1 - d_{M_{min}} = 1$. For $m = M_{min} + 1, \ldots, M_{max} - 1$, for simplification purpose, we choose the same value for both b_m and d_m as $b_m = d_m = 0.5$. Within every iteration, we generate a random value $u' \sim U_{[0,1]}$ respectively for the four RJMCMC steps. If $b_m >= u'$ or $d_m >= u'$, birth/split or death/merge steps should be performed correspondingly [33].

Merge and Split Steps: Randomly choose two components (j_1, j_2) satisfying that $\mu_{j_1} < \mu_{j_2}$ with no other μ_j in the interval $[\mu_{j_1}, \mu_{j_2}]$. The newly merged component j' will contain the observations that previously belonged to both component j_1 and j_2. Meanwhile, reduce current value of component number m to $m - 1$, then calculate mixture weight and parameters for j' as follows:

$$\begin{aligned} p_{j'} &= p_{j_1} + p_{j_2} \\ p_{j'}\mu_{j'} &= p_{j_1}\mu_{j_1} + p_{j_2}\mu_{j_2} \\ p_{j'}(\mu_{j'}^2 + \sigma_{j'l}^2) &= p_{j_1}(\mu_{j_1}^2 + \sigma_{j_1 l}^2) + p_{j_1}(\mu_{j_1}^2 + \sigma_{j_1 l}^2) \\ p_{j'}(\mu_{j'}^2 + \sigma_{j'r}^2) &= p_{j_1}(\mu_{j_1}^2 + \sigma_{j_1 r}^2) + p_{j_1}(\mu_{j_1}^2 + \sigma_{j_1 r}^2) \end{aligned} \tag{17}$$

As a reverse of merge step, we split component j' into two (j_1 and j_2) with 3 degrees of freedom ($u_1 \sim Beta(2,2), u_2 \sim Beta(2,2), u_3 \sim Beta(1,1)$) and, accordingly, increase m to $m+1$. Therefore, mixture parameters for split components can be calculated as follows:

$$\begin{aligned} p_{j_1} &= p_{j'}u_1, p_{j_2} = p_{j'}u_2 \\ \mu_{j_1} &= \mu_{j'} - \frac{u_2(\sigma_{j'l} + \sigma_{j'r})}{2}\sqrt{\frac{p_{j_2}}{p_{j_1}}} \\ \mu_{j_2} &= \mu_{j'} + \frac{u_2(\sigma_{j'l} + \sigma_{j'r})}{2}\sqrt{\frac{p_{j_1}}{p_{j_2}}} \\ \sigma_{j_1 l}^2 &= u_3(1 - u_2^2)\sigma_{j'l}^2\frac{p_{j'}}{p_{j_1}} \\ \sigma_{j_1 r}^2 &= u_3(1 - u_2^2)\sigma_{j'r}^2\frac{p_{j'}}{p_{j_1}} \\ \sigma_{j_2 l}^2 &= (1 - u_3)(1 - u_2^2)\sigma_{j'l}^2\frac{p_{j'}}{p_{j_2}} \\ \sigma_{j_2 r}^2 &= (1 - u_3)(1 - u_2^2)\sigma_{j'r}^2\frac{p_{j'}}{p_{j_2}} \end{aligned} \tag{18}$$

In order to decide whether the merge and split steps should be accepted or not, the acceptance probability [33] can be derived as follows:

$$\begin{aligned}\mathcal{A} = &\frac{p(\mathcal{X}, Z|\Theta')}{p(\mathcal{X}, Z|\Theta)} \frac{m'\mathcal{P}(m'|\lambda)}{\mathcal{P}(m|\lambda)} \frac{p_{j_1}^{\gamma-1+n_1} p_{j_2}^{\gamma-1+n_2}}{p_{j'}^{\gamma-1+n_1+n_2} Beta(\gamma, m\gamma)} \\ &\times \sqrt{\frac{\kappa}{2\pi}} \exp[-\frac{1}{2}\kappa(\mu_{j_1} - \xi) + (\mu_{j_2} - \xi) + (\mu_{j'} - \xi)] \\ &\times \frac{\beta^\alpha}{\Gamma(\alpha)} (\frac{\sigma_{j_1 l}^2 \sigma_{j_1 r}^2 \sigma_{j_2 l}^2 \sigma_{j_2 r}^2}{\sigma_{j' l}^2 \sigma_{j' r}^2})^{-\alpha-1} \\ &\times \exp[-\beta(\sigma_{j_1 l}^2 + \sigma_{j_1 r}^2 + \sigma_{j_2 l}^2 + \sigma_{j_2 r}^2 - \sigma_{j' l}^2 - \sigma_{j' r}^2)] \\ &\times \frac{d_{m'}}{b_m P_{alloc}} [Beta(\mu_1|2,2) Beta(\mu_2|2,2) Beta(\mu_3|1,1)]^{-1} \\ &\times \frac{p_{j'} |\mu_{j_1} - \mu_{j_2}| \sigma_{j_1 l}^2 \sigma_{j_1 r}^2 \sigma_{j_2 l}^2 \sigma_{j_2 r}^2}{\mu_2(1-\mu_2^2)\mu_3(1-\mu_3)\sigma_{j' l}^2 \sigma_{j' r}^2} \end{aligned} \tag{19}$$

where Θ' and $m' = m + 1$ denote the mixture parameters set and the component number respectively before merge or after split steps. κ is a known hyperparameter and ξ is the midpoint of the variation interval of the involved data observations. Besides, P_{alloc} is the probability of which this particular allocation is made. Therefore, the acceptance probability for merge step is $\min(1, \mathcal{A})$ and, correspondingly, for split step is $\min(1, \mathcal{A}^{-1})$.

Birth and Death Steps: Compared to merge and split steps, birth and death steps are relatively straightforward because the newborn and dead components are empty ones which means parameter re-calculation is not needed. Mixture weight p_{new} in birth step can be obtained by sampling from Beta distribution $p_{new} \sim Beta(1, m)$ and mixture parameters can be derived from the priors as follows [14]:

$$\mu \sim \mathcal{N}(\xi, \kappa^{-1}), \quad \sigma_l^{-2}, \sigma_r^{-2} \sim \Gamma(\alpha, \beta), \quad \beta \sim \Gamma(g, h) \tag{20}$$

where hyperparameters κ, α, g and h are chosen according to the data. For death step, an empty component should be randomly selected and deleted among the existing components if there is any. Otherwise, this step will be skipped. After birth and death steps, mixture weights p_j should be re-scaled so that all weights sum to 1. Acceptance probability for birth and death steps is also required as the one for merge and split steps whose definition is as follows:

$$\begin{aligned}\mathcal{A}' = &\frac{\mathcal{P}(m'|\lambda)}{\mathcal{P}(m|\lambda)} \frac{1}{Beta(m\gamma, \gamma)} p_{j'}^{\gamma-1} (1 - p_{j'})^{N+m\gamma-m} \\ &\times m' \frac{d_{m'}}{(m_0+1)b_m} \frac{1}{Beta(p_{j'}|1, m)} (1 - p_{j'})^m \end{aligned} \tag{21}$$

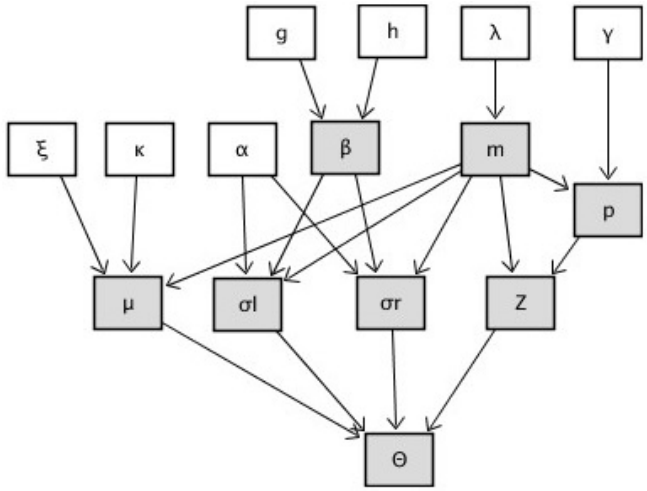

Fig. 1 DAG of RJMCMC parameter learning Bayesian network

where m_0 is the amount of empty components. Thus, the probabilities of occurrence of birth and death steps are $\min(1, \mathcal{A}')$ and $\min(1, \mathcal{A}'^{-1})$ [33].

Finally, Figure 1 describes the dependencies between constants and variables involved in the Bayesian network of RJMCMC mixture parameter learning, and then, a typical learning procedure of AGM can be summarized as follows:

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

1. Initialization
2. Step t: For $t = 1, \ldots$
   Gibbs sampling part
   (a) Generate $Z^{(t)}$ from Eq. (4)
   (b) Compute $n_j^{(t)}$ from Eq. (7)
   (c) Generate $p_j^{(t)}$ from Eq. (9)
   Metropolis-Hastings part
   (d) Sample $\xi_j^{(t)}$ $(\mu_j^{(t)}, \sigma_{lj}^{(t)}, \sigma_{rj}^{(t)})$ from Eqs. (10) (11) (12)
   (e) Compute acceptance ratio r from Eq. (13)
   (f) Generate $\alpha = min[1, r]$ and $u \sim U_{[0,1]}$
   (g) If $u \geq \alpha$ then $\xi^{(t)} = \xi^{(t-1)}$
   RJMCMC part
   (h) Generate $u' \sim U_{[0,1]}$. If $b_m >= u'$, perform split or birth step, then calculate acceptance probability $\mathcal{A}$. If the step is accepted, set $m = m + 1$.
   (i) Generate $u' \sim U_{[0,1]}$. If $d_m >= u'$, perform merge or death step, then calculate acceptance probability $\mathcal{A}'$. If the step is accepted, set $m = m - 1$.

2.4 Model Selection

Theoretically, RJMCMC learning process should always be able to derive the optimal components number M. However, because of the stochastic sampling, improper proposal distributions or bad initialization parameters, learning result based on a single estimation run is not always satisfactory. In order to establish a robust parameter estimation algorithm, we evaluate the estimation outputs derived from multiple RJMCMC runs with different initial values of components number by calculating their marginal likelihood with the Laplace approximation [9] on the logarithm scale which is defined as follows:

$$\log(p(\mathcal{X}|M)) = \log(p(\mathcal{X}|\hat{\Theta}, M)) + \log(\pi(\hat{\Theta}|M)) + \frac{N_p}{2}\log(2\pi) + \frac{1}{2}\log(|H(\hat{\Theta})|) \tag{22}$$

where $\hat{\Theta}$ denotes the proposed optimal parameter set derived from a specific learning process and $\pi(\hat{\Theta}|M)$ is the prior density of mixture parameters as well as its Hessian matrix $H(\hat{\Theta})$ which is asymptotically equal to the posterior covariance matrix.

2.5 Experimental Results

Firstly, 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 [35] as testing database. K-means algorithm is used for initialization and the results analysis will be based on statistics derived from confusion matrix. Then, the proposed approach will be deployed to the Spambase spam filtering database contains multiple spam textual features including spam word/character dictionaries and profiles of uninterrupted capital letter sequences.

2.5.1 Synthetic Data

The main goals of this section are feasibility analysis and efficiency evaluation of the AGM learning algorithm. The number of observations is set to 300 grouped into two clusters ($M = 2$). Hyperparameters are set to $\gamma_j = 1$ [34] for sampling mixture weight p_j from Eq. (9). 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 [9]. The probability density functions are plotted for both

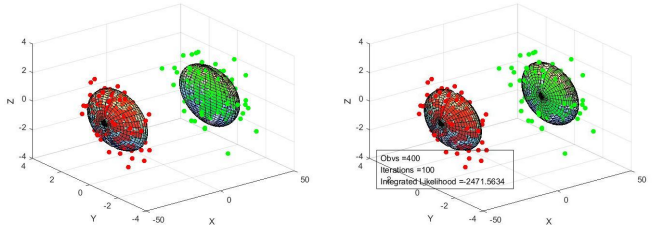

Fig. 2 Original synthetic data grouping and learning results

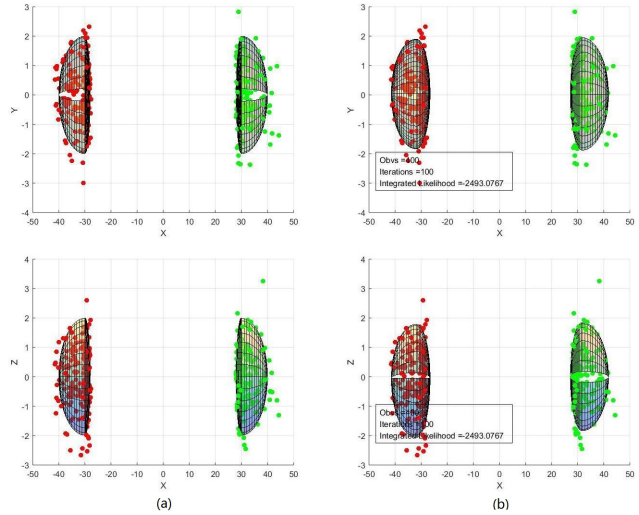

Fig. 3 (a) Original synthetic data grouping; (b) AGM clustering results

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

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 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

2.5.2 Intrusion Detection

Along with the rapid growth of information technologies, personal and commercial behaviors tend to rely on computer network and Internet environments. However, based on the characteristics of networking, exposing sensitive privacy and valuable business secret online is extremely dangerous because accessibility and anonymity make network intrusions hard to be detected and traced, therefore, compromise network security. Cisco 2017 Annual Cybersecurity Report (ACR) [24] pointed out a crucial fact that more than one-third of organizations that experienced a breach in 2016 reported more than 20 percent of customer, opportunity and revenue loss. As a consequence, more than 90 percent of these organizations are improving threat defense technologies and processes by enhancing IT and security functions, increasing security training of employees and implementing risk mitigation techniques. Recently, machine learning-based intrusion detection solutions [13, 1] are drawing more attention because of their efficiency and flexibility.

Earlier intrusion prevention approaches, such as authentication, avoiding programming errors and encryption, were proven as insufficient because along with the increasing of the complexity of network-based software systems, exploitable weaknesses are inevitable due to programming issues. Moreover, authentication and encryption are not always reliable since credentials could be leaked and encryption algorithm could also be compromised by applying powerful hacking techniques to make the attack feasible. In consequence, once intrusion happens, detection will be harder than prevention and sometimes victims could not be even aware of it. Therefore, many supervised data mining solutions were proposed in terms of misuse and anomaly detection systems by establishing known intrusion scenarios, normal usage patterns and the sequential interrelations between user operations to identify intrusion behaviors [28]. However, the disadvantages of supervised intrusion detection systems are significant since predefined patterns and interrelations are inconsistent concerning the system upgrades and newly-founded intrusions which could lead to incessant intrusion detection system adjustment and affect its performance. Furthermore, inductive bias and overfitting problems caused by poor training datasets will also affect the accuracy of the systems. Therefore, researchers are

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

paying more attention to unsupervised solution [3,2] for seeking flexibility and robustness.

Therefore, we select NSL-KDD [35], an improved KDDCUP'99 intrusion-detection data-set, 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:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{23}$$

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 [22] 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 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.

2.5.3 Spam Filtering

Statistics reveal a crucial fact that more than 59% of worldwide e-mail traffic is considered as unsolicited messages, also well known as spams, in 2017 [26]. Most spams are irritating and resource-consuming, and some of them are extremely dangerous in terms of phishing scam, fee fraud, job offer scam, etc,. Since the damages of spam are persistent and significant not only for individuals but also for governments, companies and organizations, many spam filtering technologies have been proposed to address this issue and eliminate unwanted e-mails automatically over recent decades.

Consequently, a well organized Spambase dataset [31] is selected with attributes related to multiple spam textual features including spam word/character dictionaries and profiles of uninterrupted capital letter sequences. Data preprocessing includes Scaling-based data normalization which re-scales numerical values within the range between 0 and 1 and label extraction for generating confusion matrix. To better evaluate the performance and accuracy of AGM model under different initial number of components, the integrated likelihood [9] values are given in Table 5 to identify the best-fit result. Obviously, the result with initial component number $m = 3$ has the largest integrated likelihood value (8.4238e5). Therefore, we select it as the best-fit result and make horizontal comparison with GMM. Statistics in Table 6 reveal the fact that comparing to GMM, AGM provides higher accuracy and precision, additionally, lower false positive rate and false negative rate indicate that AGM outperforms GMM. However, because of the nature of spambase, the performance of both mixture models is not satisfactory since most of spams cannot be identified. Therefore, data-based adjustment of the model might lead to a better result in the future.

Table 5 AGM Statistics

Init. Comp. Number m	***Accuracy***	***Integrated Likelihood***
$m = 1$	55.64%	5.7074e5
$m = 2$	51.21%	4.0543e5
$m = 3$	58.99%	8.4238e5

Table 6 Confusion Matrices and Statistics of GMM and AGM

GMM

	NF [a]	***F*** [b]
NF	35	1778
F	295	2493

AGM

	NF	***F***
NF	249	1564
F	323	2465

	GMM	***AGM***
Accuracy	54.94%	58.99%
Precision	1.93%	13.81%
False Positive Rate	41.63%	38.81%
False Negative Rate	89.39%	56.46%

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

2.5.4 Conclusion

This chapter firstly 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 shall extend the Bayesian learning process and introduce model selection and feature selection methodologies to improve the performance in the case of high-dimensional datasets.

References

1. Ashfaq, R.A.R., Wang, X.Z., Huang, J.Z., Abbas, H., He, Y.L.: Fuzziness based semi-supervised learning approach for intrusion detection system. Information Sciences **378**, 484 – 497 (2017)
2. Azam, M., Bouguila, N.: Unsupervised keyword spotting using bounded generalized gaussian mixture model with ICA. In: 2015 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2015, Orlando, FL, USA, December 14-16, 2015, pp. 1150–1154. IEEE (2015)
3. Bouguila, N.: Bayesian hybrid generative discriminative learning based on finite liouville mixture models. Pattern Recognition **44**(6), 1183–1200 (2011)
4. Bouguila, N.: Count data modeling and classification using finite mixtures of distributions. IEEE Trans. Neural Networks **22**(2), 186–198 (2011)
5. Bouguila, N., Elguebaly, T.: A fully bayesian model based on reversible jump MCMC and finite beta mixtures for clustering. Expert Syst. Appl. **39**(5), 5946–5959 (2012)
6. Bouguila, N., Ziou, D.: Dirichlet-based probability model applied to human skin detection [image skin detection]. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2004, Montreal, Quebec, Canada, May 17-21, 2004, pp. 521–524. IEEE (2004)
7. Bouguila, N., Ziou, D.: A powreful finite mixture model based on the generalized dirichlet distribution: Unsupervised learning and applications. In: 17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, August 23-26, 2004., pp. 280–283. IEEE Computer Society (2004)
8. Bouguila, N., Ziou, D., Boutemedjet, S.: Simultaneous non-gaussian data clustering, feature selection and outliers rejection. In: S.O. Kuznetsov, D.P. Mandal, M.K. Kundu,

S.K. Pal (eds.) Pattern Recognition and Machine Intelligence - 4th International Conference, PReMI 2011, Moscow, Russia, June 27 - July 1, 2011. Proceedings, *Lecture Notes in Computer Science*, vol. 6744, pp. 364–369. Springer (2011)
9. Bouguila, N., Ziou, D., Hammoud, R.I.: On bayesian analysis of a finite generalized dirichlet mixture via a metropolis-within-gibbs sampling. Pattern Anal. Appl. **12**(2), 151–166 (2009)
10. Bouguila, N., Ziou, D., Monga, E.: Practical bayesian estimation of a finite beta mixture through gibbs sampling and its applications. Statistics and Computing **16**(2), 215–225 (2006)
11. Bourouis, S., Mashrgy, M.A., Bouguila, N.: Bayesian learning of finite generalized inverted dirichlet mixtures: Application to object classification and forgery detection. Expert Syst. Appl. **41**(5), 2329–2336 (2014)
12. Boutemedjet, S., Bouguila, N., Ziou, D.: A hybrid feature extraction selection approach for high-dimensional non-gaussian data clustering. IEEE Trans. Pattern Anal. Mach. Intell. **31**(8), 1429–1443 (2009)
13. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys Tutorials **18**(2), 1153–1176 (2016)
14. Casella, G., Robert, C.P., Wells, M.T.: Mixture models, latent variables and partitioned importance sampling. Statistical Methodology **1**(1-2), 1–18 (2004)
15. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society. Series B (methodological) pp. 1–38 (1977)
16. Dy, J.G., Brodley, C.E.: Feature selection for unsupervised learning. Journal of Machine Learning Research **5**, 845–889 (2004)
17. Elguebaly, T., Bouguila, N.: Bayesian learning of finite generalized gaussian mixture models on images. Signal Processing **91**(4), 801–820 (2011)
18. Elguebaly, T., Bouguila, N.: Simultaneous bayesian clustering and feature selection using rjmcmc-based learning of finite generalized dirichlet mixture models. Signal Processing **93**(6), 1531–1546 (2013)
19. Elguebaly, T., Bouguila, N.: Background subtraction using finite mixtures of asymmetric gaussian distributions and shadow detection. Mach. Vis. Appl. **25**(5), 1145–1162 (2014)
20. Elguebaly, T., Bouguila, N.: Simultaneous high-dimensional clustering and feature selection using asymmetric gaussian mixture models. Image and Vision Computing **34**, 27–41 (2015)
21. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In: Readings in Computer Vision, pp. 564–584. Elsevier (1987)
22. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics) **28**(1), 100–108 (1979)
23. Hastings, W.K.: Monte carlo sampling methods using markov chains and their applications. Biometrika **57**(1), 97–109 (1970)
24. Investor.cisco.com: Cisco 2017 annual cybersecurity report (2018). URL https://investor.cisco.com/investor-relations/news-and-events/news/news-details/2017/Cisco-2017-Annual-Cybersecurity-Report-Chief-Security-Officers-Reveal-True-Cost-of-Breaches-And-The-Actions-That-Organizations-Are-Taking/default.aspx
25. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. **97**(1-2), 273–324 (1997)
26. Lab, K.: Spam: share of global email traffic 2014-2017 (2018). URL https://www.statista.com/statistics/420391/spam-email-traffic-share/
27. Law, M.H., Figueiredo, M.A., Jain, A.K.: Simultaneous feature selection and clustering using mixture models. IEEE transactions on pattern analysis and machine intelligence **26**(9), 1154–1166 (2004)
28. Lee, W., Stolfo, S.J.: Data mining approaches for intrusion detection. In: A.D. Rubin (ed.) Proceedings of the 7th USENIX Security Symposium, San Antonio, TX, USA, January 26-29, 1998. USENIX Association (1998)
29. Luengo, D., Martino, L.: Fully adaptive gaussian mixture metropolis-hastings algorithm. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013, pp. 6148–6152. IEEE (2013)

30. Mao, K.Z.: Identifying critical variables of principal components for unsupervised feature selection. IEEE Trans. Systems, Man, and Cybernetics, Part B **35**(2), 339–344 (2005)
31. Mark Hopkins Erik Reeber, G.F.J.S.: Uci machine learning repository: Spambase data set (2018). URL http://archive.ics.uci.edu/ml/datasets/Spambase?ref=datanews.io
32. Raudys, S., Jain, A.K.: Small sample size effects in statistical pattern recognition: Recommendations for practitioners. IEEE Trans. Pattern Anal. Mach. Intell. **13**(3), 252–264 (1991)
33. Richardson, S., Green, P.J.: 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), 731–792 (1997)
34. Stephens, M.: Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods. Annals of statistics pp. 40–74 (2000)
35. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, Ottawa, Canada, July 8-10, 2009, pp. 1–6. IEEE (2009)
36. Tsai, C., Chiu, C.: Developing a feature weight self-adjustment mechanism for a k-means clustering algorithm. Computational Statistics & Data Analysis **52**(10), 4658–4672 (2008)
37. Wen, C.K., Jin, S., Wong, K.K., Chen, J.C., Ting, P.: Channel estimation for massive mimo using gaussian-mixture bayesian learning. IEEE Transactions on Wireless Communications **14**(3), 1356–1368 (2015)
38. Yang, J., Liao, X., Yuan, X., Llull, P., Brady, D.J., Sapiro, G., Carin, L.: Compressive sensing by learning a gaussian mixture model from measurements. IEEE Transactions on Image Processing **24**(1), 106–119 (2015)