Asymmetric Gaussian Mixtures with Reversible Jump MCMC on Spam Filtering

1st Shuai Fu
Faculty of Engineering and Computer Science
Concordia University
Montreal, Canada
f_shuai@encs.concordia.ca

2nd Nizar Bouguila
Concordia Institute for Information Systems Engineering
Concordia University
Montreal, Canada
bouguila@ciise.concordia.ca

***Abstract*—We propose a fully Bayesian learning approach using reversible jump Markov chain Monte Carlo (RJMCMC) for asymmetric Gaussian mixtures (AGM). Compared to classic Gaussian mixture model, AGM doesn't imply that target data is symmetric which brings flexibilities and better fitting results. This paper also introduces a RJMCMC learning implementation based on Metropolis-Hastings (MH) within Gibbs sampling method. As an improvement of traditional sampling-based MCMC learning, RJMCMC has no assumption of components number of data and, therefore, the AGM model itself could be transferred between iterations. For better evaluating models with different mixture components number, the model selection is achieved by calculating integrated likelihood using Laplace approximation to figure out the best-fit components number. We selected both synthetic dataset and a challenging spam filtering application as target datasets to show the merits of the proposed model.**

***Index Terms*—Asymmetric Gaussian Mixture, Metropolis-Hastings, Gibbs sampling, RJMCMC, Laplace approximation, Spam Filtering**

I. Introduction

The statistics reveals a crucial fact that more than 59% of worldwide e-mail traffic is considered as unsolicited messages, also well known as spams, in the year of 2017 [1]. Although most spams are irritating and resource-consuming, some of them are positively 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.

Most modern spam filtering approaches can be classified into two categories: supervised learning and unsupervised learning. As widely deployed solutions for spam filtering, supervised learning includes the following variations: (a) Probabilistic classifiers (eg: Naive Bayes [3], Maximum Entropy Model [4], etc,.), (b) Memory-based learning [5], (c) SVM-based learning [6] and (d) Boosting [7], etc,. Supervised approaches perform well under some circumstances, but compared to unsupervised learning methods, they have significant limitations and drawbacks because of the nature that supervised classifiers cannot identify new spam patterns not presented in their training datasets. Once new patterns are discovered, model adjustment will be needed. Furthermore, poor training datasets could cause inductive bias and overfitting problems which will affect the accuracy of the models. Therefore, unsupervised solutions have been increasingly drawing attention because of its flexibility and robustness.

As widely deployed unsupervised learning approaches, mixture models can be viewed as an improvement of independent methodologies which superimposes a finite number of components while respecting the dependency between data clusters, demonstrating outstanding suitability and generality especially for complex high-dimensional datasets. More precisely, for Gaussian-like datasets, Gaussian mixture model (GMM) [10] is proven as an effective learning approach in several domains such as computer vision, pattern recognition and data mining. In this paper, we show the merits of asymmetric Gaussian mixture (AGM) model [11] for modeling because of its two variance parameters for left and right parts of each distribution in the mixture which brings more accuracy of fitting real datasets which could be asymmetric or even non-Gaussian.

It could be a challenging task of estimating the parameters of mixture models. The maximum-likelihood-based expectation maximization (EM) [12] 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 which could cause slow convergence and compromise the usability of the algorithm. Furthermore, bad initialization and overfitting problems [13] [14] 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 distributions for mixture parameters. In this paper, the learning process is accomplished by a hybrid MCMC algorithm, which is well known as Metropolis-Hastings within Gibbs sampling [13], based on both Metropolis-Hastings (Hastings, 1970) [15] and Gibbs sampling (Geman and Geman, 1984) [16] 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) [14] 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 [13] to perform model selection in order to evaluate fitting results between models.

The rest of this paper is organized as follows. Section 2 illustrates the AGM model including its Bayesian learning and model selection processes. Section 3 focuses on experimental results derived from both synthetic data and real spam filtering database. Finally, Section 4 concludes this paper.

II. Bayesian model

A. Asymmetric Gaussian Mixture Model

The likelihood function of AGM model (Elguebaly and Bouguila, 2013) [11] with M mixture components can be 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 $\mathcal{X} = (X_1, ..., X_N)$ reprensents the dataset with N observations, $\Theta = \{p_1, ..., p_M, \xi_1, ..., \xi_M\}$ defines the mxiture 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 [11] for j-th component of the model can be defined as follows:

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

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 $\sigma_{rj} = (\sigma_{rj1}, ..., \sigma_{rjd})$ are the left and right standard deviation vectors of AGD .

Since AGM is probalistic, for discriminative clustring purpose we introduce a M-dimensional membership vector Z for each observation $X_i \in \mathcal{X}$, $Z_i = (Z_{i1}, ..., Z_{iM})$ which indicates to which specific component X_i belongs (Bouguila, Ziou and Monga, 2006) [9], 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, $Z_{ij} = 0$.

Therefore, the complete likelihood function can be derived by combining q. (1) and Eq. (3) as follows:

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

B. Priors and Posteriors

As discussed before, MH-within-Gibbs based RJMCMC learning algorithm implementation defines priors and posteriors for mixture weighs and parameters to avoid direct sampling. For a specific iteration t, since mixture weight p_j satisfies following rules that $0 < p_j \leq 1$ and $\sum_{j=1}^{M} p_j$ = 1, a nature choice of the prior is Dirichlet distribution [18] as follows:

$$\pi(p_j^{(t)}) \sim \mathcal{D}(\gamma_1, ..., \gamma_M) \tag{5}$$

where γ_j is known hyperparameter. By considering the membership vector Z as a condition, The posterior probability of mixture weight p_j is defined as follows:

$$p(p_j^{(t)}|Z^{(t)}) \sim \mathcal{D}(\gamma_1 + n_1^{(t)}, ..., \gamma_M + n_M^{(t)}) \tag{6}$$

where n_j represents numbers of observations belonging to component j which could be calculated using membership vectors as follows:

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

The same idea applies to the sampling process of mixture parameters. The proposal posterior distribution is $\xi^{(t)} \sim q(\xi|\xi^{(t-1)})$. To be more specific, for parameters of AGM model $\xi^{(t)} = (\mu^{(t)}, \sigma_l^{(t)}, \sigma_r^{(t)})$. We choose d-dimensional Gaussian distributions as posterior distributions respectively:

$$\mu_j^{(t)} \sim \mathcal{N}_d(\mu_j^{(t-1)}, \Sigma) \tag{8}$$

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

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

where Σ is d x d identity matrix which makes the sampling a random walk MCMC process. Correspondingly, the priors are $\mu \sim \mathcal{N}_d(\eta, \Sigma)$ and $\sigma_l, \sigma_r \sim \mathcal{N}_d(\tau, \Sigma)$ given known hyperparameters η and τ.

C. Learning Algorithm

MH-within-Gibbs: As a sampling-based learning algorithm, MH-within-Gibbs method performs random sampling from posteriors of parameters, and then calculate the acceptance ratio r in order to make a decision whether the new samples should be accepted or discarded for next iteration. Because of the usage of membership vector Z, the mixture weight p_j can be derived within Gibbs sampling part. Therefore, it will be excluded from the calculation of the acceptance ratio r which is defined as follows:

$$r = \frac{p(\mathcal{X}|\Theta^{(t)})\pi(\Theta^{(t)})q(\Theta^{(t-1)}|\Theta^{(t)})}{p(\mathcal{X}|\Theta^{(t-1)})\pi(\Theta^{(t-1)})q(\Theta^{(t)}|\Theta^{(t-1)})} \tag{11}$$

Further information about the calculation of acceptance ratio r is explained in Appendix A.

Once acceptance ratio r is derived, we compute acceptance probability $\alpha = min[1, r]$ [19]. 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 $p^{(t)} = p^{(t-1)}$, $\xi^{(t)} = \xi^{(t-1)}$.

RJMCMC moves: Traditional MH-within-Gibbs algorithm has an implicate that the components number M is given and persistent throughout the learning process. However, because of bad initialization or just leaking of information, components number M could be inaccurate or unknown. Under these circumstances, RJMCMC algorithm is found to be useful by providing extra four independent steps (birth/split steps and death/merge steps) into learning process which could change components number M, therefore, brings more generalities.

Letting M_{min} and M_{max} denote the minimum and maximum value of components number M, assuming 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 $m = M_{min} + 1, \ldots, M_{max} - 1$, due to simplification reasons, 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. [10]

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 previously belong to both component j_1 and j_2. At the meanwhile, reduce current value of components number $m = m - 1$, then calculate mixture weight and parameters for j' as follows:

$$\begin{aligned} p_{j'} &= p_{j_1} + p_{j_2} \\ p_{j'}\mu_{j'} &= p_{j_1}\mu_{j_1} + p_{j_2}\mu_{j_2} \\ p_{j'}(\mu_{j'}^2 + \sigma_{j'l}^2) &= p_{j_1}(\mu_{j_1}^2 + \sigma_{j_1 l}^2) + p_{j_1}(\mu_{j_1}^2 + \sigma_{j_1 l}^2) \\ p_{j'}(\mu_{j'}^2 + \sigma_{j'r}^2) &= p_{j_1}(\mu_{j_1}^2 + \sigma_{j_1 r}^2) + p_{j_1}(\mu_{j_1}^2 + \sigma_{j_1 r}^2) \end{aligned} \quad (12)$$

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 $m = m+1$. Therefore, mixture parameters for split components can be calculated as follows:

$$\begin{aligned} p_{j_1} &= p_{j'}u_1, p_{j_2} = p_{j'}u_2 \\ \mu_{j_1} &= \mu_{j'} - \frac{u_2(\sigma_{j'l} + \sigma_{j'r})}{2}\sqrt{\frac{p_{j_2}}{p_{j_1}}} \\ \mu_{j_2} &= \mu_{j'} + \frac{u_2(\sigma_{j'l} + \sigma_{j'r})}{2}\sqrt{\frac{p_{j_1}}{p_{j_2}}} \\ \sigma_{j_1 l}^2 &= u_3(1-u_2^2)\sigma_{j'l}^2\frac{p_{j'}}{p_{j_1}} \\ \sigma_{j_1 r}^2 &= u_3(1-u_2^2)\sigma_{j'r}^2\frac{p_{j'}}{p_{j_1}} \\ \sigma_{j_2 l}^2 &= (1-u_3)(1-u_2^2)\sigma_{j'l}^2\frac{p_{j'}}{p_{j_2}} \\ \sigma_{j_2 r}^2 &= (1-u_3)(1-u_2^2)\sigma_{j'r}^2\frac{p_{j'}}{p_{j_2}} \end{aligned} \quad (13)$$

In order to decide whether the merge and split steps should be accepted or not, we calculate the acceptance probability $\mathcal{A}$ which is described in (Richardson, S. and Green, P.J., 1997) [10]. 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 Beat distribution $p_{new} \sim Beta(1, m)$ and mixture parameters can be derived from the priors as follows [25]:

$$\mu \sim \mathcal{N}(\xi, \kappa^{-1}), \quad \sigma_l^{-2}, \sigma_r^{-2} \sim \Gamma(\alpha, \beta) \quad (14)$$

where hyperparameters κ, α and β are estimated by data and ξ is the midpoint of the observations. For death step, a random choice is made between existing empty components and simply delete the selected one. If there is no empty component, death 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 $\mathcal{A}'$ for birth and death steps is also required as the one for merge and split steps. The probabilities of occurrence of birth and death steps are $\min(1, \mathcal{A}')$ and $\min(1, \mathcal{A}'^{-1})$ [10].

Finally, a typical MH-within-Gibbs learning procedure for AGM model can be summarized as follows:

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. (3)
   b) Compute $n_j^{(t)}$ from Eq. (7)
   c) Generate $p_j^{(t)}$ from Eq. (6)
   Metropolis-Hastings part
   d) Sample $\xi_j^{(t)}$ $(\mu_j^{(t)}, \sigma_{lj}^{(t)}, \sigma_{rj}^{(t)})$ from Eqs. (8) (9) (10)
   e) Compute acceptance ratio r from Eq. (11)

f) Generate $\alpha = min[1, r]$ and $u \sim U_{[0,1]}$
g) If $\alpha \geq u$ 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 $m = m - 1$.

D. Model Selection

Since the RJMCMC learning is stochastic and components number M could be variable, the grouping could end up with different clusters numbers between different attempts. In order to identify the best-fit result, we choose integrated likelihood to achieve model selection as follows:

$$p(\mathcal{X}|M) = \int \pi(\Theta|\mathcal{X}, M)d\Theta = \int p(\mathcal{X}|\Theta, M)\pi(\Theta|M)d\Theta \tag{15}$$

taking the Laplace approximation [13] and logarithm into account, we could rewrite Eq. (15) as follows:

$$\log(p(\mathcal{X}|M)) = \log(p(\mathcal{X}|\Theta, M)) + \log(\pi(\Theta|M)) + \frac{N_p}{2}\log(2\pi) + \frac{1}{2}\log(|H(\Theta)|) \tag{16}$$

where $H(\Theta)$ is the Hassan matrix for mixture parameters set Θ and asymptotically equal to the posterior variance matrix. $\pi(\Theta|M)$ denotes prior distributions for mixture parameters which are defined in Eq. (14). Therefore, the best-fit result should have largest integrated likelihood value derived from Eq. (16)

III. Experimental results

A. 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 [20] as testing database. K-means algorithm is used for initialization and the results analysis will be based on statistics derived from confusion matrix.

B. 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 $\gamma_j = 1$ [21] for sampling mixture weight p_j from Eq. (6). η and τ are considered as d-dimensional zero vectors in prior distributions of mixture parameter ξ.

Different proposed components number ($M' = 1, \ldots, 5$) are tested during the AGM learning process and the statistics are summarized in Table I. In order to select the best number of components, we consider marginal likelihood as described in [13]. 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.

TABLE I
AGM Learning Statistics

Components 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

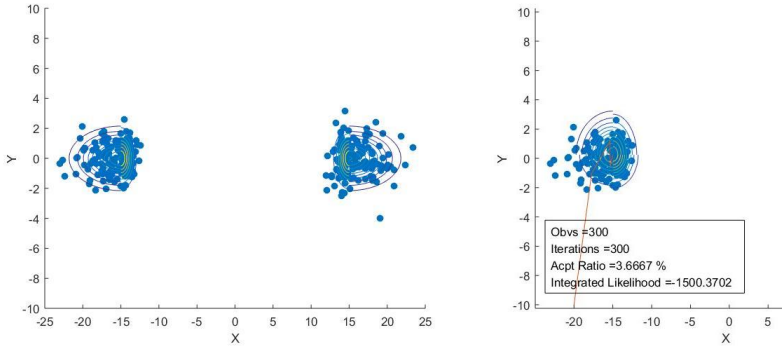

Fig. 1. Original synthetic observations and learning result ($M' = M = 2$)

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 II). 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.

C. Intrusion Detection

Along with the development of information-based industries, network security problems are becoming increasingly

TABLE II
Accuracy Analysis ($M' = M = 2$)

Components number $j = 1$	**Mean** (μ_j)	**Left standard deviation** (σ_{lj})	**Righ devi**
ξ	[-15.00, 0.00]	[10.00, 1.00]	[1.
$\hat{\xi}$	[-14.99, 0.25]	[4.77, 1.13]	[2.
Euclidean Distance	0.246	5.236	
Components number $j = 2$	**Mean** (μ_j)	**Left standard deviation** (σ_{lj})	**Righ devi**
ξ	[15.00, 0.00]	[1.00, 1.00]	[10
$\hat{\xi}$	[14.02, -0.24]	[2.04, 1.04]	[5.
Euclidean Distance	1.010	1.036	

important today. In order to address this challenge, many data mining methodologies were proposed including both classification-based [22] and clustering-based [23] 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 [20], 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:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{17}$$

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 III).

K-means clustering algorithm [24] 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 IV) 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 III
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 IV
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***
Accuracy	60.85%	60.85%
Precision	20.82%	20.98%
False Positive Rate	41.92%	41.90%
False Negative Rate	18.77%	19.16%

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

IV. 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

A. Derivation of Acceptance Ratio r by Eq. (11)

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

$$\begin{aligned} \pi(\Theta) &= \pi(p, \xi) = \pi(\xi) \\ &= \prod_{j=1}^{M} \pi(\mu_j)\pi(\sigma_{lj})\pi(\sigma_{rj}) \\ &= \prod_{j=1}^{M} \mathcal{N}_d(\mu_j|\eta, \Sigma)\mathcal{N}_d(\sigma_{lj}|\tau, \Sigma)\mathcal{N}_d(\sigma_{rj}|\tau, \Sigma) \end{aligned} \tag{18}$$

in Eq. (18), 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:

$$q(\Theta^{(t)}|\Theta^{(t-1)}) = q(\xi^{(t)}|\xi^{(t-1)}) = \prod_{j=1}^{M} \mathcal{N}_d(\mu_j^{(t)}|\mu_j^{(t-1)}, \Sigma)\mathcal{N}_d(\sigma_{lj}^{(t)}|\sigma_{lj}^{(t-1)}, \Sigma)\mathcal{N}_d(\sigma_{rj}^{(t)}|\sigma_{rj}^{(t-1)}, \Sigma) \tag{19}$$

by combining Eqs. (2) (4) (8) (9) (10) (18) and (19), equation (11) can be written as follows:

$$\begin{aligned} r &= \frac{p(\mathcal{X}|\Theta^{(t)})\pi(\Theta^{(t)})q(\Theta^{(t-1)}|\Theta^{(t)})}{p(\mathcal{X}|\Theta^{(t-1)})\pi(\Theta^{(t-1)})q(\Theta^{(t)}|\Theta^{(t-1)})} \\ &= \prod_{i=i}^{N}\prod_{j=1}^{M}\Big(\frac{p(X_i|\mu_j^{(t)}, \sigma_{lj}^{(t)}, \sigma_{rj}^{(t)})}{p(X_i|\mu_j^{(t-1)}, \sigma_{lj}^{(t-1)}, \sigma_{rj}^{(t-1)})} \\ &\times \frac{\mathcal{N}_d(\mu_j^{(t)}|\eta, \Sigma)\mathcal{N}_d(\sigma_{lj}^{(t)}|\tau, \Sigma)\mathcal{N}_d(\sigma_{rj}^{(t)}|\tau, \Sigma)}{\mathcal{N}_d(\mu_j^{(t-1)}|\eta, \Sigma)\mathcal{N}_d(\sigma_{lj}^{(t-1)}|\tau, \Sigma)\mathcal{N}_d(\sigma_{rj}^{(t-1)}|\tau, \Sigma)} \\ &\times \frac{\mathcal{N}_d(\mu_j^{(t-1)}|\mu_j^{(t)}, \Sigma)\mathcal{N}_d(\sigma_{lj}^{(t-1)}|\sigma_{lj}^{(t)}, \Sigma)\mathcal{N}_d(\sigma_{rj}^{(t-1)}|\sigma_{rj}^{(t)}, \Sigma)}{\mathcal{N}_d(\mu_j^{(t)}|\mu_j^{(t-1)}, \Sigma)\mathcal{N}_d(\sigma_{lj}^{(t)}|\sigma_{lj}^{(t-1)}, \Sigma)\mathcal{N}_d(\sigma_{rj}^{(t)}|\sigma_{rj}^{(t-1)}, \Sigma)} \end{aligned} \tag{20}$$

REFERENCES

[1] Global spam volume as percentage of total e-mail traffic from January 2014 to September 2017, m. (2018). Spam statistics: spam e-mail traffic share 2017 — Statista. [online] Statista. Available at: https://www.statista.com/statistics/420391/spam-email-traffic-share/ [Accessed 11 Jan. 2018].

[2] Zhang, L., Zhu, J. and Yao, T. (2004). An evaluation of statistical spam filtering techniques. ACM Transactions on Asian Language Information Processing, 3(4), pp.243-269.

[3] McCallum, A. and Nigam, K., 1998, July. A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization (Vol. 752, pp. 41-48).

[4] Berger, A.L., Pietra, V.J.D. and Pietra, S.A.D., 1996. A maximum entropy approach to natural language processing. Computational linguistics, 22(1), pp.39-71.

[5] Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C.D. and Stamatopoulos, P., 2000. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. arXiv preprint cs/0009009.

[6] Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine learning, 20(3), pp.273-297.

[7] Freund, Y., Schapire, R. and Abe, N., 1999. A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence, 14(771-780), p.1612.

[8] McLachlan, G. and Peel, D. (2000). Finite mixture models. New York: Wiley.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Hastings, W. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika, 57(1), p.97.

[16] 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.

[17] Elguebaly, T. and Bouguila, N. (2011). Bayesian learning of finite generalized Gaussian mixture models on images. Signal Processing, 91(4), pp.801-820.

[18] Marin, J.M. and Mengersen, K.R.C., 2005. Handbook of Statistics: Bayesian modelling and inference on mixtures of distributions, Vol. 25.

[19] 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.

[20] 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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] Casella, G., Robert, C. and Wells, M. (2004). Mixture models, latent variables and partitioned importance sampling. Statistical Methodology, 1(1-2), pp.1-18.