Anomaly Intrusion Detection Using Incremental Learning of an Infinite Mixture Model with Feature Selection

Wentao Fan[1], Nizar Bouguila[1], and Hassen Sallay[2]

[1] Concordia University, Montreal, QC, Canada
wenta_fa@encs.concordia.ca, nizar.bouguila@concordia.ca
[2] Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia
hmsallay@imamu.edu.sa

Abstract. We propose an incremental nonparametric Bayesian approach for clustering. Our approach is based on a Dirichlet process mixture of generalized Dirichlet (GD) distributions. Unlike classic clustering approaches, our model does not require the number of clusters to be pre-defined. Moreover, an unsupervised feature selection scheme is integrated into the proposed nonparametric framework to improve clustering performance. By learning the proposed model using an incremental variational framework, the number of clusters as well as the features weights can be automatically and simultaneously computed. The effectiveness and merits of the proposed approach are investigated on a challenging application namely anomaly intrusion detection.

Keywords: Mixture models, clustering, Dirichlet process, generalized Dirichlet, feature selection, variational inference, intrusion detection.

1 Introduction

Huge volumes of data are routinely generated by organizations, scientific activities, internet traffic and so on. An important problem is to model these data to improve the process of making automatic decisions [12]. A widely used approach for data modeling and knowledge discovery is clustering. Clustering can be defined as the task of partitioning a given data set $\mathcal{X} = \{\boldsymbol{X}_1, \dots, \boldsymbol{X}_N\}$ containing N vectors into M homogenous clusters $\mathcal{C}_1, \dots, \mathcal{C}_M$ such that $\mathcal{C}_j \cap \mathcal{C}_l = \emptyset$, and $\cup_{j=1}^{M} \mathcal{C}_j = \mathcal{X}$. Finite mixture models have been widely applied for clustering during the last two decades [11]. Within finite mixture modeling, selecting the number of components that best describes the underlying data without over- or under-fitting is one of the most challenging problems. This obstacle can be removed by extending finite mixtures to the infinite case through Dirichlet processes [13]. Infinite mixtures allow a natural approach for data clustering. Unlike finite mixtures, the number of clusters does not need to be specified by the practitioner in advance and can be automatically inferred from the dataset. Several approaches have been proposed to learn mixture models. In particular, variational inference has received a lot of attention recently [5,4,1,6]. Variational

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 364–373, 2013.

inference is a deterministic approximation learning technique that only requires a modest amount of computational power in contrast to other well-developed approaches such as Markov chain Monte Carlo (MCMC) techniques, and has a tractable learning process as well. Generally real-world problems involve dynamic data sets where the volume of data continuously grows. Thus, it is crucial to adopt an incremental way to learn the statistical model used for clustering.

In this paper, we adopt an incremental version of variational inference proposed by [7] to learn infinite generalized Dirichlet (GD) mixtures with unsupervised feature selection. The employment of the GD as the basic distribution in our mixture model is motivated by its favorable performance when dealing with non-Gaussian data [2,3]. The advantages of our framework are summarized as following: First, the difficulty of choosing the appropriate number of components is avoided by assuming that there is an infinite number of components. Second, thanks to its incremental nature, it is very efficient when dealing with sequentially arriving data, which is an important factor for real-time applications. Third, within the proposed framework, the model parameters and features saliencies can be estimated simultaneously and automatically. The effectiveness of our approach is illustrated through a challenging task namely anomaly intorsion detection. The rest of this paper is organized as follows. Section 2 reviews briefly the infinite GD mixture model with unsupervised feature selection. In Section 3, we develop an incremental variational inference framework for model learning. Section 4 is devoted to the experimental results. Finally, conclusion follows in Section 5.

2 Infinite GD Mixture Model with Feature Selection

In this section, we review briefly the infinite generalized Dirichlet (GD) mixture model with feature selection, which is constructed using a stick-breaking Dirichlet process framework. If a D-dimensional random vector $\boldsymbol{Y} = (Y_1, \dots, Y_D)$ is sampled from a mixture of GD distributions with infinite number of components:

$$p(\boldsymbol{Y}|\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{j=1}^{\infty} \pi_j \mathrm{GD}(\boldsymbol{Y}|\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j) \tag{1}$$

where $\boldsymbol{\pi}$ represents the mixing coefficients with the constraints that are positive and sum to one. Here we adopt the Dirichlet process framework with a stick-breaking representation [15], where the mixing coefficients $\{\pi_j\}$ are constructed by recursively breaking a unit length stick into an infinite number of pieces as $\pi_j = \lambda_j \prod_{k=1}^{j-1}(1 - \lambda_k)$. The stick breaking variable λ_j is distributed according to $\lambda_j \sim \mathrm{Beta}(1, \zeta)$, where ζ is a positive real number and is the concentration parameter of the Dirichlet process. In Eq. (1), $\boldsymbol{\alpha}_j = (\alpha_{j1}, \dots, \alpha_{jD})$ and $\boldsymbol{\beta}_j = (\beta_{j1}, \dots, \beta_{jD})$ are the positive parameters of the GD distribution $\mathrm{GD}(\boldsymbol{Y}|\boldsymbol{\alpha}_j, \boldsymbol{\beta})$ associated with component j, where $\mathrm{GD}(\boldsymbol{X}|\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j)$ is given by

$$\mathrm{GD}(\boldsymbol{Y}|\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j) = \prod_{l=1}^{D} \frac{\Gamma(\alpha_{jl} + \beta_{jl})}{\Gamma(\alpha_{jl})\Gamma(\beta_{jl})} Y_l^{\alpha_{jl}-1} \left(1 - \sum_{k=1}^{l} Y_k\right)^{\gamma_{jl}} \tag{2}$$

where $\sum_{l=1}^{D} Y_l < 1$ and $0 < y_l < 1$ for $l = 1, \ldots, D$, $\gamma_{jl} = \beta_{jl} - \alpha_{jl+1} - \beta_{jl+1}$ for $l = 1, \ldots, D-1$, and $\gamma_{jD} = \beta_{jD} - 1$. $\Gamma(\cdot)$ is the gamma function defined by $\Gamma(x) = \int_0^\infty u^{x-1}e^{-u}du$. It is noteworthy that, in practice the features $\{Y_l\}$ are generally not equally significant for the clustering task since some features may be"noise" and do not contribute to clustering process. Therefore, feature selection may act as a crucial role to improve the learning performance. Before incorporating feature selection into our framework, we leverage a handy mathematical property of the GD distribution which is introduced in [3], to transform the original data points into another D-dimensional space with independent features. Then, we can rewrite the infinite GD mixture model as

$$p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{j=1}^{\infty} \pi_j \prod_{l=1}^{D} \text{Beta}(X_l|\alpha_{jl}, \beta_{jl}) \tag{3}$$

where $X_l = Y_l$ and $X_l = Y_l/(1 - \sum_{k=1}^{l-1} Y_k)$ for $l > 1$. $\text{Beta}(X_l|\alpha_{jl}, \beta_{jl})$ is a Beta distribution parameterized with $(\alpha_{jl}, \beta_{jl})$. Accordingly, the independence between the features in the new space becomes a fact rather than an assumption as considered in previous approaches [8,4]. In this work, we adopt an unsupervised feature selection scheme suggested in [8]: the lth feature is irrelevant if its distribution is independent of the class labels, that is, if it follows a common density. Thus, we can rewrite the mixture density in Eq. (3) as

$$p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\phi}, \boldsymbol{\sigma}, \boldsymbol{\tau}) = \sum_{j=1}^{\infty} \prod_{l=1}^{D} \big[\text{Beta}(X_l|\alpha_{jl}, \beta_{jl})\big]^{\phi_l} \big[\text{Beta}(X_l|\sigma_l, \tau_l)\big]^{1-\phi_l} \tag{4}$$

where $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_D)$ is a set of binary parameters and known as the feature relevance indicator, such that $\phi_l = 0$ if feature l is irrelevant (i.e. noise) and follows a Beta distribution: $\text{Beta}(X_l|\sigma_l, \tau_l)$. The prior of $\boldsymbol{\phi}$ is defined as:

$$p(\boldsymbol{\phi}|\boldsymbol{\epsilon}) = \prod_{l=1}^{D} \epsilon_{l_1}^{\phi_l} \epsilon_{l_2}^{1-\phi_l} \tag{5}$$

where each ϕ_l is a Bernoulli variable such that $p(\phi_l = 1) = \epsilon_{l_1}$ and $p(\phi_l = 0) = \epsilon_{l_2}$. Here the vector $\boldsymbol{\epsilon}$ denotes the features saliencies (i.e. the probabilities that the features are relevant) where $\boldsymbol{\epsilon}_l = (\epsilon_{l_1}, \epsilon_{l_2})$ and $\epsilon_{l_1} + \epsilon_{l_2} = 1$. Furthermore, we place a Dirichlet prior $\text{Dir}(\cdot)$ over $\boldsymbol{\epsilon}$ with positive parameter $\boldsymbol{\varphi}$ as: $p(\boldsymbol{\epsilon}) = \prod_{l=1}^{D} \text{Dir}(\boldsymbol{\epsilon}_l|\boldsymbol{\varphi})$. In mixture modeling, it is convenient to introduce a variable $\boldsymbol{Z} = (Z_1, \ldots, Z_N)$ for an observed dataset $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N)$, where Z_i is an assignment variable of the mixture component with which the data point $\boldsymbol{X}_i$ is associated. The marginal distribution over $\boldsymbol{Z}$ is given by

$$p(\boldsymbol{Z}|\boldsymbol{\lambda}) = \prod_{i=1}^{N} \prod_{j=1}^{\infty} \left[\lambda_j \prod_{k=1}^{j-1} (1 - \lambda_k)\right]^{\mathbf{1}[Z_i = j]} \tag{6}$$

where $\mathbf{1}[\cdot]$ is an indicator function which has the value of 1 when $Z_i = j$ and 0 otherwise. Next, we need to introduce prior distributions over unknown random variables. In this work, the Gamma distribution $\mathcal{G}(\cdot)$ is adopted to approximate a conjugate prior over parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\sigma}$ and $\boldsymbol{\tau}$, by assuming that these Beta parameters are statistically independent: $p(\boldsymbol{\alpha}) = \mathcal{G}(\boldsymbol{\alpha}|\boldsymbol{u}, \boldsymbol{v})$, $p(\boldsymbol{\beta}) = \mathcal{G}(\boldsymbol{\beta}|\boldsymbol{p}, \boldsymbol{q})$, $p(\boldsymbol{\sigma}) = \mathcal{G}(\boldsymbol{\sigma}|\boldsymbol{g}, \boldsymbol{h})$, $p(\boldsymbol{\tau}) = \mathcal{G}(\boldsymbol{\tau}|\boldsymbol{s}, \boldsymbol{t})$.

3 Incremental Variational Model Learning

In this work, we adopt a variational incremental learning approach introduced in [7] to learn the proposed model. According to this approach, data instances can be sequentially processed in small batches where each one may contain one or more data points. There are two phases involved: a model building phase and a compression phase. In the model building phase, the current model with observed data points is optimized. The goal of the compression phase is to determine which mixture component that groups of data points should be assigned to.

3.1 Model Building Phase

Given an observed dataset $\mathcal{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N)$, let $\Theta = \{\boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\phi}, \boldsymbol{\epsilon}, \boldsymbol{\lambda}\}$ be the set of random variables. In variational learning, the main goal is to determine a proper approximation $q(\Theta)$ for the real posterior distribution $p(\Theta|\mathcal{X})$ by maximizing the free energy $\mathcal{F}(\mathcal{X}, q)$:

$$\mathcal{F}(\mathcal{X}, q) = \int q(\Theta) \ln[p(\mathcal{X}, \Theta)/q(\Theta)] d\Theta \tag{7}$$

In our framework, motivated by [1], we truncate the variational distribution $q(\Theta)$ at a value of M, such that $\lambda_M = 1$, $\pi_j = 0$ when $j > M$, and $\sum_{j=1}^{M} \pi_j = 1$. It is noteworthy that the truncation level M is a variational parameter which can be freely initialized and will be optimized automatically during the learning process [1]. Next, we adopt a factorization assumption to factorize $q(\Theta)$ into disjoint tractable factors as: $q(\Theta) = q(\boldsymbol{Z})q(\boldsymbol{\alpha})q(\boldsymbol{\beta})q(\boldsymbol{\sigma})q(\boldsymbol{\tau})q(\boldsymbol{\phi})q(\boldsymbol{\epsilon})q(\boldsymbol{\lambda})$. Then, we can obtain the following update equations for these factors by maximizing the free energy $\mathcal{F}(\mathcal{X}, q)$ with respect to each of them:

$$q(\boldsymbol{Z}) = \prod_{i=1}^{N} \prod_{j=1}^{M} r_{ij}^{\mathbf{1}[Z_i=j]}, \qquad q(\boldsymbol{\lambda}) = \prod_{j=1}^{M} \mathrm{Beta}(\lambda_j|a_j, b_j) \tag{8}$$

$$q(\boldsymbol{\alpha}) = \prod_{j=1}^{M} \prod_{l=1}^{D} \mathcal{G}(\alpha_{jl}|u_{jl}^*, v_{jl}^*), \qquad q(\boldsymbol{\beta}) = \prod_{j=1}^{M} \prod_{l=1}^{D} \mathcal{G}(\beta_{jl}|c_{jl}^*, d_{jl}^*) \tag{9}$$

$$q(\boldsymbol{\sigma}) = \prod_{l=1}^{D} \mathcal{G}(\sigma_l|g_l^*, h_l^*), \qquad q(\boldsymbol{\tau}) = \prod_{l=1}^{D} \mathcal{G}(\tau_l|s_l^*, t_l^*) \tag{10}$$

$$q(\boldsymbol{\phi}) = \prod_{i=1}^{N} \prod_{l=1}^{D} f_{il}^{\phi_{il}} (1 - f_{il})^{(1-\phi_{il})}, \qquad q(\boldsymbol{\epsilon}) = \prod_{l=1}^{D} \mathrm{Dir}(\boldsymbol{\epsilon}_l|\boldsymbol{\varphi}_l^*) \tag{11}$$

where we have calculated

$$r_{ij} = \frac{\exp(\rho_{ij})}{\sum_{j=1}^{M}\exp(\rho_{ij})}, \qquad \varphi_{l_1}^{*} = \varphi_{l_1} + \sum_{i=1}^{N}\langle\phi_{il}\rangle, \qquad \varphi_{l_2}^{*} = \varphi_{l_2} + \sum_{i=1}^{N}\langle 1-\phi_{il}\rangle \tag{12}$$

$$\rho_{ij} = \sum_{l=1}^{D}\langle\phi_{il}\rangle[\widetilde{\mathcal{R}}_{jl} + (\bar{\alpha}_{jl}-1)\ln X_{il} + (\bar{\beta}_{jl}-1)\ln(1-X_{il})] + \big\langle\ln\lambda_j\big\rangle + \sum_{k=1}^{j-1}\big\langle\ln(1-\lambda_k)\big\rangle$$

$$f_{il} = \frac{\exp(\widetilde{f}_{il})}{\exp(\widetilde{f}_{il}) + \exp(\widehat{f}_{il})}, \qquad a_j = 1 + \sum_{i=1}^{N}\langle Z_i = j\rangle, \qquad b_j = \zeta_j + \sum_{i=1}^{N}\sum_{k=j+1}^{M}\langle Z_i = k\rangle$$

$$\widetilde{f}_{il} = \sum_{j=1}^{M}\big\langle Z_i = j\big\rangle\big[\widetilde{\mathcal{R}}_{jl} + (\bar{\alpha}_{jl}-1)\ln X_{il} + (\bar{\beta}_{jl}-1)\ln(1-X_{il})\big] + \langle\ln\epsilon_{l_1}\rangle$$

$$\widehat{f}_{il} = \widetilde{\mathcal{F}}_l + (\bar{\sigma}_l - 1)\ln X_{il} + (\bar{\tau}_l - 1)\ln(1-X_{il}) + \langle\ln\epsilon_{l_2}\rangle$$

$$u_{jl}^{*} = u_{jl} + \sum_{i=1}^{N} r_{ij}\langle\phi_{il}\rangle\bar{\alpha}_{jl}\big[\psi(\bar{\alpha}_{jl}+\bar{\beta}_{jl}) - \psi(\bar{\alpha}_{jl}) + \bar{\beta}_{jl}\psi'(\bar{\alpha}_{jl}+\bar{\beta}_{jl})(\langle\ln\beta_{jl}\rangle - \ln\bar{\beta}_{jl})\big]$$

$$c_{jl}^{*} = c_{jl} + \sum_{i=1}^{N} r_{ij}\langle\phi_{il}\rangle\bar{\beta}_{jl}[\psi(\bar{\alpha}_{jl}+\bar{\beta}_{jl}) - \psi(\bar{\beta}_{jl}) + \bar{\alpha}_{jl}\psi'(\bar{\alpha}_{jl}+\bar{\beta}_{jl})(\big\langle\ln\alpha_{jl}\big\rangle - \ln\bar{\alpha}_{jl})]$$

$$v_{jl}^{*} = v_{jl} - \sum_{i=1}^{N}\big\langle Z_i = j\big\rangle\big\langle\phi_{il}\big\rangle\ln X_{il}, \qquad d_{jl}^{*} = d_{jl} - \sum_{i=1}^{N}\big\langle Z_i = j\big\rangle\big\langle\phi_{il}\big\rangle\ln(1-X_{il})$$

In the above equations, $\psi(\cdot)$ is the digamma function, and $\langle\cdot\rangle$ is the expectation evaluation. $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{F}}$ are the lower bounds of $\mathcal{R} = \big\langle\ln\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\big\rangle$ and $\mathcal{F} = \big\langle\ln\frac{\Gamma(\sigma+\tau)}{\Gamma(\sigma)\Gamma(\tau)}\big\rangle$, respectively. Since these expectations are intractable, we use the second-order Taylor series expansion to find their lower bounds. The hyperparameters of $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ are calculated in a similar way as for the hyperparameters of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The expected values in the above formulas are given by $\langle Z_i = j\rangle = r_{ij}$, $\langle\phi_{il}\rangle = f_{il}$, $\bar{\alpha}_{jl} = \langle\alpha_{jl}\rangle = u_{jl}^{*}/v_{jl}^{*}$, $\bar{\beta}_{jl} = c_{jl}^{*}/d_{jl}^{*}$,$\big\langle\ln\lambda_j\big\rangle = \Psi(a_j) - \Psi(a_j+b_j)$, $\big\langle\ln(1-\lambda_j)\big\rangle = \Psi(b_j) - \Psi(a_j+b_j)$, $\langle\ln\epsilon_{l_1}\rangle = \psi(\varphi_1^{*}) - \psi(\varphi_1^{*}+\varphi_2^{*})$, $\big\langle\ln\alpha_{jl}\big\rangle = \Psi(u_{jl}^{*}) - \ln v_{jl}^{*}$, and $\big\langle\ln\beta_{jl}\big\rangle = \Psi(c_{jl}^{*}) - \ln d_{jl}^{*}$.

After convergence, the observed data points are clustered into M groups according to corresponding responsibilities r_{ij}. Following [7], these newly formed groups of data points are denoted as "clumps", and these clumps are subject to the constraint that all data points $\boldsymbol{X}_i$ in the clump m share the same $q(Z_i) \equiv q(Z_m)$ which is a key factor in the following compression phase.

3.2 Compression Phase

In the compression phase, we attempt to determine clumps that possibly belong to the same mixture component while taking into account future arriving data. Suppose that we have already observed N data points, and our goal is to make an inference at some target time T where $T \geq N$. This is fulfilled by scaling the current observed data to the target size T, which is equivalent to using the variational posterior distribution of the observed data N as a predictive model of the future data [7]. Therefore, we can obtain the modified free energy for the compression phase as the following

$$\mathcal{F} = \sum_{j=1}^{M}\sum_{l=1}^{D}\Big[\Big\langle \ln\frac{p(\alpha_{jl})}{q(\alpha_{jl})}\Big\rangle + \Big\langle \ln\frac{p(\beta_{jl})}{q(\beta_{jl})}\Big\rangle\Big] + \sum_{l=1}^{D}\Big[\Big\langle \ln\frac{p(\sigma_{l})}{q(\sigma_{l})}\Big\rangle + \Big\langle \ln\frac{p(\tau_{l})}{q(\tau_{l})}\Big\rangle + \Big\langle \ln\frac{p(\epsilon_{l})}{q(\epsilon_{l})}\Big\rangle\Big]$$
$$+ \sum_{j=1}^{M}\Big\langle \ln\frac{p(\lambda_{j})}{q(\lambda_{j})}\Big\rangle + \frac{T}{N}\sum_{m}|n_m|\big[\ln\sum_{j=1}^{M}\exp(\rho_{mj}) + \ln\sum_{l=1}^{D}\exp(f_{ml})\big] \tag{13}$$

where $\frac{T}{N}$ is the data magnification factor and $|n_m|$ represents the number of data points in clump m. The corresponding update equations for maximizing this free energy function are

$$r_{mj} = \frac{\exp(\rho_{mj})}{\sum_{j=1}^{M}\exp(\rho_{mj})}, \qquad f_{ml} = \frac{\exp(\widetilde{f}_{ml})}{\exp(\widetilde{f}_{ml}) + \exp(\widehat{f}_{ml})}, \qquad \vartheta = \frac{T}{N}\sum_{m}|n_m| \tag{14}$$
$$\varphi^*_{l_1} = \varphi_{l_1} + \frac{T}{N}\sum_{m}|n_m|\langle\phi_{ml}\rangle, \qquad \varphi^*_{l_2} = \varphi_{l_2} + \frac{T}{N}\sum_{m}|n_m|\langle 1-\phi_{ml}\rangle$$
$$\rho_{mj} = \sum_{l=1}^{D} f_{ml}[\widetilde{\mathcal{R}}_{jl} + (\bar{\alpha}_{jl}-1)\ln X_{ml} + (\bar{\beta}_{jl}-1)\ln(1-X_{ml})] + \langle\ln\lambda_j\rangle + \sum_{k=1}^{j-1}\langle\ln(1-\lambda_k)\rangle$$
$$a_j = 1 + \vartheta\langle Z_m = j\rangle, \qquad b_j = \zeta_j + \vartheta\sum_{k=j+1}^{M}\langle Z_m = k\rangle$$
$$\widetilde{f}_{ml} = \sum_{j=1}^{M}\big\langle Z_m = j\big\rangle\big[\widetilde{\mathcal{R}}_{jl} + (\bar{\alpha}_{jl}-1)\ln X_{ml} + (\bar{\beta}_{jl}-1)\ln(1-X_{ml})\big] + \langle\ln\epsilon_{l_1}\rangle$$
$$\widehat{f}_{ml} = \widetilde{\mathcal{F}}_l + (\bar{\sigma}_l - 1)\ln X_{ml} + (\bar{\tau}_l - 1)\ln(1-X_{ml}) + \langle\ln\epsilon_{l_2}\rangle$$
$$u^*_{jl} = u_{jl} + \vartheta r_{mj}\langle\phi_{ml}\rangle\bar{\alpha}_{jl}\big[\psi(\bar{\alpha}_{jl}+\bar{\beta}_{jl}) - \psi(\bar{\alpha}_{jl}) + \bar{\beta}_{jl}\psi'(\bar{\alpha}_{jl}+\bar{\beta}_{jl})(\langle\ln\beta_{jl}\rangle - \ln\bar{\beta}_{jl})\big]$$
$$c^*_{jl} = c_{jl} + \vartheta r_{mj}\langle\phi_{ml}\rangle\bar{\beta}_{jl}[\psi(\bar{\alpha}_{jl}+\bar{\beta}_{jl}) - \psi(\bar{\beta}_{jl}) + \bar{\alpha}_{jl}\psi'(\bar{\alpha}_{jl}+\bar{\beta}_{jl})(\big\langle\ln\alpha_{jl}\big\rangle - \ln\bar{\alpha}_{jl})]$$
$$v^*_{jl} = v_{jl} - \vartheta r_{mj}\big\langle\phi_{ml}\big\rangle\ln X_{ml}, \qquad d^*_{jl} = d_{jl} - \vartheta r_{mj}\big\langle\phi_{ml}\big\rangle\ln(1-X_{ml})$$

where $\langle X_{ml}\rangle$ represents the average over all data points contained in clump m. In the compression phase, the first step is to hard assign each clump or data point to the component with the highest responsibility r_{mj} obtained from the model building phase as

$$I_m = \arg\max_j r_{mj} \tag{15}$$

where $\{I_m\}$ represent which component the clump (or data point) m belongs to in the compression phase. Next, we cycle through each component and split it into two subcomponents along its principal component. This splitting process can be refined by updating Eqs. (14). After convergence criterion is reached for refining the split, the clumps are then assigned to one of the two candidate components. Among all the potential splits, we choose the one that results in the largest change in the free energy (Eq. (13)). We iterate this splitting process until a stopping criterion is satisfied. Based on [7], a stoping criterion for the splitting process can be expressed as a limit on the amount of memory required to store the components. In our case, the memory cost for the mixture model is $\mathcal{MC} = 5DN_c$, where $5D$ is the number of parameters contained in a D-variate GD component with feature selection, while N_m denotes the number of components. Thus, We

Algorithm 1

```
Choose the initial truncation level $M$.
Initialize hyper-parameters: $u_{jl}$, $v_{jl}$, $c_{jl}$, $d_{jl}$, $g_l$, $h_l$, $s_l$, $t_l$, $\zeta_j$, $\varphi_{l_1}$ and $\varphi_{l_2}$.
Initialize the values of $r_{ij}$ by $K$-Means algorithm.
while More data to be observed do
    Perform the model building phase through Eqs. (8)$\sim$(11).
    Initialize the compression phase using Eq. (15).
    while $\mathcal{MC} \geq \mathcal{C}$ do
        for $j = 1$ to $M$ do
            if $evaluated(j) =$ false then
                Split component $j$ and refine this split using Eqs (14).
                $\Delta\mathcal{F}(j) =$ change in Eq. (13).
                $evaluated(j) =$ true.
            end if
        end for
        Split component $j$ with the largest value of $\Delta\mathcal{F}(j)$.
        $M = M + 1$.
    end while
    Discard the currently observed data points.
    Save the resultant components for next learning round.
end while
```

can define an upper limit on the component memory cost $\mathcal{C}$, and the compression phase stops when $\mathcal{MC} \geq \mathcal{C}$. As a result, the computational time and the space requirement is bounded in each learning round. After the compression phase, the currently observed data points are discarded while the resultant components are treated in the same way as data points in the next round of leaning. The proposed incremental variational inference algorithm for infinite GD mixture model with feature selection is summarized in Algorithm 1.

4 Anomaly Intrusion Detection

The construction of intrusion detection models has been the topic of extensive research in the past. The main goal is to protect networks against criminals. Indeed, the target of Intrusion Detection Systems (IDSs) is to discover inappropriate, incorrect, or anomalous activities within computers or networks and this can be considered as classification problem in the context of machine learning (see, [9], for instance and references therein). In general, IDSs can be broadly divided into two main categories: misuse detection and anomaly detection systems [14]. In contrast to the signature-based misuse detection, the anomaly detection has the superiority of being able to detect new or unknown attacks. In this experiment, we evaluate the effectiveness of the proposed incremental infinite GD mixture model with feature selection (referred as *InGD-Fs*) by applying it to tackle the problem of anomaly intrusion detection. In our case, the truncation level M is initialized as 20. Our specific choice for the initial values of the hyperparameters is: $(u_{jl}, v_{jl}, c_{jl}, d_{jl}, g_l, h_l, s_l, t_l, \zeta_j, \varphi_{l_1}, \varphi_{l_2}) = (1, 0.01, 1, 0.01, 1, 0.01, 1, 0.01, 0.1, 0.1, 0.1)$.

4.1 Databases and Experimental Design

We investigate our approach on two challenging publicly available databases known as the KDD Cup 1999 Data[1] and the Kyoto traffic Data[2]. In our case, a 10 percent subset of the KDD database is adopted. Specifically, the training set consists of 494,020 data instances of which 97,277 are normal and 396,743 are attacks, while the testing set contains 292,393 data instances of which 60,593 are normal and 231,800 are attacks. Each instance in this data set is composed of 41 features. This database has five categories in total including one 'Normal' and four attack classes namely: DOS, R2L, U2R and Probing. The Kyoto database consists of real traffic data obtained from several types of honeypots by the Kyoto University. In our experiment, the Kyoto database contains 784,000 21-dimensional instances where 395,368 are normal sessions and 388,632 are attacks. In this application, the training data are used to learn the current model, where the testing data instances are supposed to be obtained sequentially in an online fashion. It is noteworthy that the features in the two original databases are on quite different scales, we therefore require to normalize the databases so that one feature would not dominate the others. By finding the maximum and minimum values of a given feature X_l in a data instance $\boldsymbol{X}$, we can transform the feature into the range $[0, 1]$ by $X_l = \frac{X_l - \min(X_l)}{\max(X_l) - \min(X_l)}$, where X_l is set to a smallest value if the maximum is equal to the minimum.

4.2 Experimental Results

We run the the proposed *InGD-Fs* 20 times to investigate its performance. For comparison, we have also applied three other mixture-modeling approaches: the infinite GD mixture model without feature selection (*InGD*), the finite GD mixture model without feature selection(*FiGD*) and the infinite Gaussian mixture model with feature selection (*InGM-Fs*). In order to provide a fair comparison, all of these tested approaches are learned through the incremental variational inference. The results of applying different approaches on the KDD99 database and Kyoto database are shown in Table 1, in terms of the average classification accuracy rate (Accuracy) and the false positive (FP) rate. According to this table, it is obvious that our approach (*InGD-Fs*) has the best performance among all the tested approaches by providing the highest accuracy rate and the lowest FP rate for both databases. There are several important conclusions which can be drawn from this table: First, the fact that *InGD-Fs* outperforms *InGD* proves that feature selection is a significant factor for improving clustering performance; Second, *InGD* has better results than *FiGD*, which demonstrates

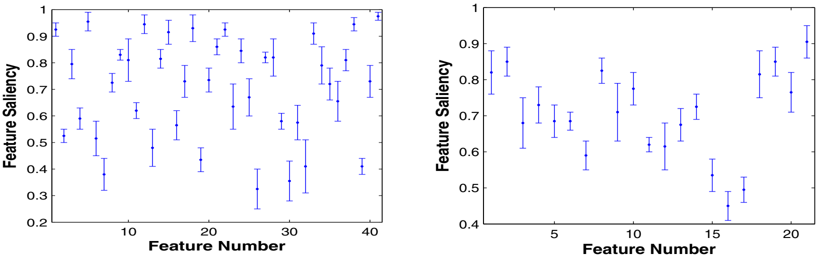
[1] `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`

[2] `http://www.takakura.com/Kyoto_data/`

the advantage of using infinite mixture models over finite ones. Third, *InGM-Fs* provides the worst performance among all tested approaches which verifies that the GD mixture model has better modeling capability than the Gaussian for compactly supported data. In addition, the saliencies of the 41 features in the KDD 99 database and 21 features in the Kyoto database calculated by the *InGD-Fs* over 20 runs are illustrated in Fig. 1. As shown in this figure, it is clear that the different features do not contribute equally in the classification, since they have different relevance degrees.

Table 1. Average classification accuracy rate (Accuracy) and false positive (FP) rate computed using different approaches

	KDD data		Kyoto data	
	Accuracy (%)	FP (%)	Accuracy (%)	FP (%)
InGD-Fs	86.73	6.27	81.34	11.78
InGD	84.18	7.14	78.61	13.52
FiGD	82.52	9.63	76.59	16.37
InGM-Fs	79.45	13.91	75.01	18.23



KDD data

Kyoto data

Fig. 1. Features saliencies obtained using the proposed *InGD-Fs* approach

5 Conclusion

In this paper, we have proposed an incremental clustering algorithm that allows the simultaneous computation of the number of clusters and features weights during execution. Our approach is based on an incremental variational learning of the infinite GD mixture model with unsupervised feature selection. The effectiveness of the proposed approach has been evaluated on a challenging real application namely anomaly intrusion detection. Future works could be devoted to the inclusion of a localized feature selection scheme, such as the one proposed in [10], to improve further the generalization capabilities of our framework.

Acknowledgments. The first two authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC). The third author would like to thank King Abdulaziz City for Science and Technology (KACST) for their funding support under grant 08-INF36-8.

References

1. Blei, D., Jordan, M.: Variational inference for Dirichlet process mixtures. Bayesian Analysis 1, 121–144 (2005)
2. Bouguila, N., Ziou, D.: A hybrid SEM algorithm for high-dimensional unsupervised learning using a finite generalized Dirichlet mixture. IEEE Transactions on Image Processing 15(9), 2657–2668 (2006)
3. Boutemedjet, S., Bouguila, N., Ziou, D.: A hybrid feature extraction selection approach for high-dimensional non-Gaussian data clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(8), 1429–1443 (2009)
4. Constantinopoulos, C., Titsias, M., Likas, A.: Bayesian feature and model selection for Gaussian mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(6), 1013–1018 (2006)
5. Corduneanu, A., Bishop, C.M.: Variational Bayesian model selection for mixture distributions. In: Proc. of the 8th International Conference on Artificial Intelligence and Statistics (AISTAT), pp. 27–34 (2001)
6. Fan, W., Bouguila, N., Ziou, D.: Variational learning for finite Dirichlet mixture models and applications. IEEE Transactions on Neural Netw. Learning Syst. 23(5), 762–774 (2012)
7. Gomes, R., Welling, M., Perona, P.: Incremental learning of nonparametric Bayesian mixture models. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)
8. Law, M.H.C., Figueiredo, M.A.T., Jain, A.K.: Simultaneous feature selection and clustering using mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), 1154–1166 (2004)
9. Lee, W., Stolfo, S.J., Mok, K.W.: Adaptive intrusion detection: A data mining approach. Artificial Intelligence Review 14(6), 533–567 (2000)
10. Li, Y., Dong, M., Hua, J.: Simultaneous localized feature selection and model detection for Gaussian mixtures. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 953–960 (2009)
11. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley, New York (2000)
12. Mitchell, T.M.: Machine learning and data mining. Communications of the ACM 42(11), 30–36 (1999)
13. Neal, R.M.: Markov chain sampling methods for Dirichlet process mixture models. Journal of Computational and Graphical Statistics 9(2), 249–265 (2000)
14. Northcutt, S., Novak, J.: Network Intrusion Detection: An Analyst's Handbook. New Riders Publishing (2002)
15. Sethuraman, J.: A constructive definition of Dirichlet priors. Statistica Sinica 4, 639–650 (1994)