A Probabilistic Approach based on a Finite Mixture Model of Multivariate Beta Distributions

Narges Manouchehri and Nizar Bouguila
Concordia Institute for Information System Engineering (CIISE), Concordia University, Montréal, Canada

Keywords: Mixture Models, Multivariate Beta Distribution, Maximum Likelihood, Clustering.

Abstract: Model-based approaches specifically finite mixture models are widely applied as an inference engine in machine learning, data mining and related disciplines. They proved to be an effective and advanced tool in discovery, extraction and analysis of critical knowledge from data by providing better insight into the nature of data and uncovering hidden patterns that we are looking for. In recent researches, some distributions such as Beta distribution have demonstrated more flexibility in modeling asymmetric and non-Gaussian data. In this paper, we introduce an unsupervised learning algorithm for a finite mixture model based on multivariate Beta distribution which could be applied in various real-world challenging problems such as texture analysis, spam detection and software modules defect prediction. Parameter estimation is one of the crucial and critical challenges when deploying mixture models. To tackle this issue, deterministic and efficient techniques such as Maximum likelihood (ML), Expectation maximization (EM) and Newton Raphson methods are applied. The feasibility and effectiveness of the proposed model are assessed by experimental results involving real datasets. The performance of our framework is compared with the widely used Gaussian Mixture Model (GMM).

1 INTRODUCTION

Over the past couple of decades, machine learning experienced tremendous growth and advancement. Accurate data analysis, extraction and retrieval of information have been largely studied in the various fields of technology (Han and Pei, 2012). Technological improvement led to the generation of huge amount of complex data of different types (Diaz-Rozo and Larranaga, 2018). Various statistical approaches have been suggested in data mining, however data clustering received considerable attention and still is a challenging and open problem (Giordan, 2015). Finite mixture models have been proven to be one of the most strong and flexible tools in data clustering and have seen a real boost in popularity. Multimodal and mixed generated data consists of different components and categories and mixture models proved to be an enhanced statistical approach to discover the latent pattern of data (McCabe, 2015). One of the crucial challenges of modeling and clustering is applying the most appropriate distribution. Most of the literatures on finite mixtures concern Gaussian mixture model (GMM) (Zhou, 2017), (Guha and Shim, 2001), (Gevers, 1999), (Hastie and Tibshirani, 1996), (Luo et al., 2017). However, GMM is not a proper tool to express the latent structure of non-Gaussian data. Recently, other distributions such as Dirichlet and Beta distributions which are more flexible have been considered as a powerful alternative (Giordan, 2015), (Olkin and Trikalinos, 2015), (Fan and Bouguila, 2013), (Cockriel and McDonald, 2018), (Elguebaly and Bouguila, 2013), (Fan et al., 2014), (Klauschies et al., 2018), (Wentao et al., 2013).

In this work, we introduce multivariate Beta mixture model to cluster k dimensional vectors with features defined between zero and one. For learning the parameters, we applied the Expectation-Maximization (EM) algorithm. Our model will be evaluated on two real world applications. The first one is software defect detection. Nowadays, complex software systems are increasingly applied and the rate of software defects is growing correspondingly. Errors, failures and defects may cause serious and costly complications in systems and projects by providing unexpected or unintended results. Hence, prediction of defective modules by statistical methods has become one of the attention-grabbing subjects of many studies using machine learning methods to differentiate fault prone or non-fault prone softwares (Malhotra and Jain, 2012). Spam filtering is our second topic of interest. Evolutionary automated communication by

Manouchehri, N. and Bouguila, N.
A Probabilistic Approach based on a Finite Mixture Model of Multivariate Beta Distributions.
DOI: 10.5220/0007707003730380
In *Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019)*, pages 373-380
ISBN: 978-989-758-372-8

Internet improved the style of everyday communication. Electronic mail is a dominant medium for digital communications as it is convenient, economical and fast. However, unwanted emails take advantage of the Internet. Spam emails are sent to widely and economically advertise a specific product or service, serve online frauds (Malhotra and Jain, 2012) or are carrying a piece of malicious code that might damage the end user machines.

The rest of this paper is organized as follows; In sections 2 and 3, we present our proposed mixture model and model learning, respectively. Model assessment is performed by devoting section 4 to experimental results and the accuracy of our model is estimated by comparing it with Gaussian mixture models. Finally, we conclude this paper in section 5.

2 THE MIXTURE MODEL

In this section, we propose a new mixture model based on a multivariate Beta distribution.

2.1 The Finite Multivariate Beta Distribution

Bivariate and multivariate Beta distributions have been introduced by Olkin and Liu (Olkin and Liu, 2003), (Olkin and Trikalinos, 2015). This article is devoted to our proposed mixture model based on multivariate Beta distribution. In this section, we will briefly introduce the bivariate distribution with three shape parameters and then describe the multivariate case in detail.

Let us consider two correlated random variables X and Y defined by Beta distribution and described as follows:

$$X = \frac{U}{(U+W)} \tag{1}$$

$$Y = \frac{V}{(V+W)} \tag{2}$$

U, V and W are three independent random variables arisen from standard Gamma distribution and parametrized by their shape parameters a, b and c, respectively. Both variables X and Y have positive real values and are less than one. The joint density function of this bivariate distribution is expressed by Equation 3.

$$f(X,Y) = \frac{X^{a-1}Y^{b-1}(1-X)^{b+c-1}(1-Y)^{a+c-1}}{B(a,b,c)(1-XY)^{(a+b+c)}} \tag{3}$$

where

$$B(a,b,c) = \frac{\Gamma(a)\Gamma(b)\Gamma(c)}{\Gamma(a+b+c)} \tag{4}$$

The multivariate Beta distribution is constructed by generalization of above bivariate distributions to k variate distribution. Let $U_1, ..., U_k$ and W be independent random variables each having a Gamma distribution and variable X is defined by Equation 5 where $i = 1, ...k$.

$$X_i = \frac{U_i}{(U_i+W)} \tag{5}$$

The joint density function of $X_1, ..., X_k$ after integration over W is expressed by:

$$f(x_1,...,x_k) = c\frac{\prod_{i=1}^{k} x_i^{a_i-1}}{\prod_{i=1}^{k}(1-x_i)^{(a_i+1)}}\left[1+\sum_{i=1}^{k}\frac{x_i}{(1-x_i)}\right]^{-a} \tag{6}$$

where x_i is between zero and one and:

$$c = B^{-1}(a_1,...,a_k) = \frac{\Gamma(a_1+...+a_k)}{\Gamma(a_1)......\Gamma(a_k)} = \frac{\Gamma(a)}{\prod_{i=1}^{k}\Gamma(a_i)} \tag{7}$$

a_i is the shape parameter of each variable X_i and:

$$a = \sum_{i=1}^{k} a_i \tag{8}$$

2.2 Mixture model

Let us consider $\mathcal{X} = \{\vec{X}_1, \vec{X}_2, ..., \vec{X}_N\}$ be a set of N k-dimensional vectors such that each vector $\vec{X}_n = (X_{n1}, ..., X_{nk})$ is generated from a finite but unknown multivariate Beta mixture model $p\left(\vec{X}|\Theta\right)$. We assume that $\mathcal{X}$ is composed of M different finite clusters and can be approximated by a finite mixture model as below (Bishop, 2006), (Figueiredo and Jain, 2002), (McLachlan and Peel, 2000):

$$p\left(\vec{X}|\Theta\right) = \sum_{j=1}^{M} p_j p(\vec{X}|\vec{\alpha}_j) \tag{9}$$

where $\vec{\alpha}_j = (a_1, ..., a_k)$. The weight of component j is denoted by p_j. All of mixing proportions are positive and sum to one.

$$\sum_{j=1}^{M} p_j = 1 \tag{10}$$

The complete model parameters are denoted by $\{p_1, ..., p_M, \vec{\alpha}_1, ..., \vec{\alpha}_M\}$ and $\Theta = (p_j, \vec{\alpha}_j)$ represents the set of weights and shape parameters of component j.

3 MODEL LEARNING

In this section, we first estimate the initial values of the parameters. Then, the optimal parameters are estimated by developing maximum likelihood estimation within EM algorithm. The initialization phase is based on k-means framework and method of moments.

3.1 Method of Moments for the Finite Multivariate Beta Distribution

Method of moments (MM) is a statistical technique to estimate model's parameter. Considering Equation 11 and Equation 12 as the first two moments, sample mean and variance, the shape and scale parameters of Beta distribution can be estimated using the method of moments by Equation 13 and Equation 14 as follow:

$$E(X) = \bar{x} = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{11}$$

$$Var(X) = \bar{v} = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{x})^2 \tag{12}$$

$$\hat{\alpha} = E(X)\Big(\frac{E(X)}{Var(X)}\big(1-E(X)\big)-1\Big) \tag{13}$$

$$\hat{\beta} = \Big(1-E(X)\Big)\Big(\frac{E(X)}{Var(X)}\big(1-E(X)\big)-1\Big) \tag{14}$$

By the help of the mean and variance of components obtained from k-means phase, the initial parameters are approximated.

3.2 Maximum Likelihood and EM Algorithm

As one of the suggested methods to tackle the problem of finding the parameters of our model, we apply maximum likelihood estimate (ML) approach (Ganesalingam, 1989) and expectation maximization (EM) framework (McCabe, 2015), (McLachlan and Krishnan, 2008) on the complete likelihood. In this technique, the parameters which maximize the probability density function of data are estimated as follow:

$$\Theta^* = \arg\max_{\Theta} \mathcal{L}(\mathcal{X}, \Theta) \tag{15}$$

$$\mathcal{L}(\Theta, \mathcal{X}) = \log\Big(p(\mathcal{X}|\Theta)\Big) = \sum_{n=1}^{N} \log\Big(\sum_{j=1}^{M} p_j p(\vec{X}_n|\vec{\alpha}_j)\Big) \tag{16}$$

Each $\vec{X}_n$ is supposed to be arisen from one of the components. Hence, a set of membership vectors is introduced as $\vec{Z}_n = (\vec{Z}_{n1}, \ldots, \vec{Z}_{nM})$ where:

$$z_{nj} = \begin{cases} 1 & \text{if X belongs to a component } j \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

$$\sum_{j=1}^{M} z_{nj} = 1 \tag{18}$$

This gives the following complete log-likelihood:

$$\mathcal{L}(\Theta, Z, \mathcal{X}) = \sum_{j=1}^{M}\sum_{n=1}^{N} z_{nj}\Big(\log p_j + \log p(\vec{X}_n|\vec{\alpha}_j)\Big) \tag{19}$$

In the EM algorithm, as the first step in Expectation phase, we assign each vector $\vec{X}_n$ to one of the clusters by its posterior probability given by:

$$\hat{Z}_{nj} = p(j|\vec{X}_n, \vec{\alpha}_j) = \frac{p_j p(\vec{X}_n|\vec{\alpha}_j)}{\sum_{j=1}^{M} p_j p(\vec{X}_n|\vec{\alpha}_j)} \tag{20}$$

The complete log-likelihood is computed as:

$$\begin{aligned}
\mathcal{L}(\Theta, Z, \mathcal{X}) &= \sum_{j=1}^{M}\sum_{n=1}^{N} \hat{Z}_{nj}\big(\log p_j + \log p(\vec{X}_n|\vec{\alpha}_j)\big) = \\
&\sum_{j=1}^{M}\sum_{n=1}^{N} \hat{Z}_{nj}\Bigg(\log p_j + \log\Big(\frac{\prod_{i=1}^{k} X_{ni}^{(a_{ji}-1)}}{\prod_{i=1}^{k}(1-X_{ni})^{(a_{ji}+1)}} \times \\
&\Big[1+\sum_{i=1}^{k}\frac{X_{ni}}{(1-X_{ni})}\Big]^{-a_j} \times \frac{\Gamma(\sum_{i=1}^{k} a_{ji})}{\prod_{i=1}^{k}\Gamma(a_{ji})}\Big)\Bigg) = \\
&\sum_{j=1}^{M}\sum_{n=1}^{N} \hat{Z}_{nj}\Bigg(\log p_j + \log\Big(\prod_{i=1}^{k} X_{ni}^{(a_{ji}-1)}\Big) \\
&-\log\Big(\prod_{i=1}^{k}(1-X_{ni})^{(a_{ji}+1)}\Big) + \log\big(\Gamma(a_j)\big) \\
&-\log\prod_{i=1}^{k}\Gamma(a_{ji}) + \log\Big[1+\sum_{i=1}^{k}\frac{X_{ni}}{(1-X_{ni})}\Big]^{-a_j}\Bigg) = \\
&\sum_{j=1}^{M}\sum_{n=1}^{N} \hat{Z}_{nj}\Bigg(\log p_j + \sum_{i=1}^{k}(a_{ji}-1)\big(\log(X_{ni})\big) - \\
&\sum_{i=1}^{k}(a_{ji}+1)\big(\log(1-X_{ni})\big) + \log\big(\Gamma(a_j)\big) - \\
&\sum_{i=1}^{k}\log\big(\Gamma(a_{ji})\big) - a_j\log\Bigg(\Big[1+\sum_{i=1}^{k}\frac{X_{ni}}{(1-X_{ni})}\Big]\Bigg)\Bigg)
\end{aligned} \tag{21}$$

The value of a_j is computed by Equation 13 for each component of mixture model.

To reach our ultimate goal and in maximization step, the gradient of the log-likelihood is calculated with respect to parameters. To solve optimization problem, we need to find a solution for the following equation:

$$\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \Theta} = 0 \tag{22}$$

The first derivatives of Equation 21 with respect to a_{ji} where $i = 1, ..., k$ are given by:

$$\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial a_{ji}} = \sum_{j=1}^{M} \sum_{n=1}^{N} \hat{Z}_{nj} \Bigg(\log(X_{ni}) - \log(1 - X_{ni}) + \Psi(a_j) - \Psi(a_{ji}) - \log \Big[1 + \sum_{i=1}^{k} \frac{X_{ni}}{(1 - X_{ni})} \Big] \Bigg) \tag{23}$$

where $\Psi(.)$ and $\Psi'(.)$ are digamma and trigamma functions respectively defined as follow:

$$\Psi(X) = \frac{\Gamma'(X)}{\Gamma(X)} \tag{24}$$

$$\Psi'(X) = \frac{\Gamma''(X)}{\Gamma(X)} - \frac{\Gamma'(X)^2}{\Gamma(X)^2} \tag{25}$$

As this equation doesn't have a closed form solution, we use an iterative approach named Newton-Raphson method expressed as follow:

$$\hat{\alpha}_j{}^{new} = \hat{\alpha}_j{}^{old} - H_j{}^{-1} G_j \tag{26}$$

where G_j is the first derivatives vector described in Equation 23 and H_j is Hessian matrix.

$$G_j = \left(G_{1j}, ..., G_{kj} \right)^T \tag{27}$$

The Hessian matrix is calculated by computing the second and mixed derivatives of $L(\Theta, Z, \mathcal{X})$.

$$H_j = \begin{bmatrix} \frac{\partial G_{j1}}{\partial a_{j1}} & \cdots & \frac{\partial G_{j1}}{\partial a_{jk}} \\ \vdots & \ddots & \vdots \\ \frac{\partial G_{jk}}{\partial a_{j1}} & \cdots & \frac{\partial G_{jk}}{\partial a_{jk}} \end{bmatrix} = \sum_{i=1}^{N} \hat{Z}_{nj} \times \tag{28}$$

$$\begin{bmatrix} \Psi'(|\vec{a}_j|) - \Psi'(a_{j1}) & \cdots & \Psi'(|\vec{a}_j|) \\ \vdots & \ddots & \vdots \\ \Psi'(|\vec{a}_j|) & \cdots & \Psi'(|\vec{a}_j|) - \Psi'(a_{jk}) \end{bmatrix}$$

where

$$|\vec{a}_j| = a_1 + ... + a_k \tag{29}$$

The estimated values of mixing proportions are expressed by Equation 30 as it has a closed-form solution:

$$p_j = \frac{\sum_{n=1}^{N} p(j | \vec{X}_n, \vec{\alpha}_j)}{N} \tag{30}$$

3.3 Estimation Algorithm

The initialization and estimation framework is described as follows:

1. INPUT: k-dimensional data $\vec{X}_n$ and M.
2. Apply the k-means to obtain initial M clusters.
3. Apply the MOM to obtain $\vec{\alpha}_j$.
4. E- step: Compute $\hat{Z}_{nj}$ using Equation 20.
5. M-step: Update the $\vec{\alpha}_j$ using Equation 26 and p_j using Equation 30.
6. If $p_j < \varepsilon$, discard component j and go to 4.
7. If the convergence criterion passes terminate, else go to 4.

4 EXPERIMENTAL RESULTS

In this section, we estimate the accuracy of our algorithm by testing on two real world applications. As the first step, we normalize our datasets by Equation 31 as one of the assumptions of our distribution is that the values of all observations are positive and less than one.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{31}$$

To assess the accuracy of the algorithm, the observations are assigned to different clusters based on Bayesian decision rule. Afterward, the accuracy is inferred by confusion matrix. At the next step, multivariate Beta mixture and Gaussian mixture model will be compared.

4.1 Software Defect Prediction

Software quality assurance and detection of a fault or a defect in a software program have become one of the topics that have received lots of attention in research and technology. Any failure in software may result in high costs for the system (Bertolino, 2007), (Briand and Hetmanski, 1993), (El Emam and Rai, 2001). The evaluation of the quality of complex software systems is costly and complicated. Consequently, prediction

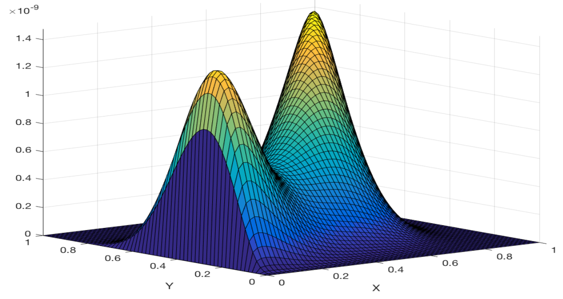

Figure 1: Two-component mixture of bivariate Beta distribution.

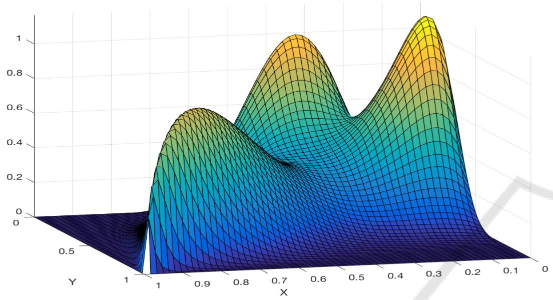

Figure 2: Three-component mixture of bivariate Beta distribution.

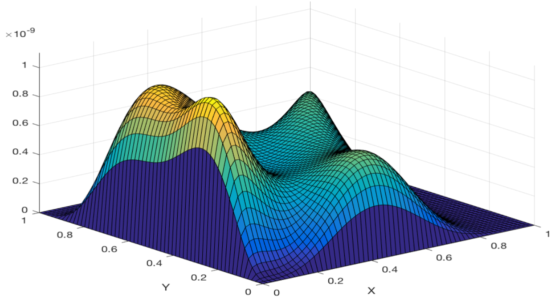

Figure 3: Four-component mixture of bivariate Beta distribution.

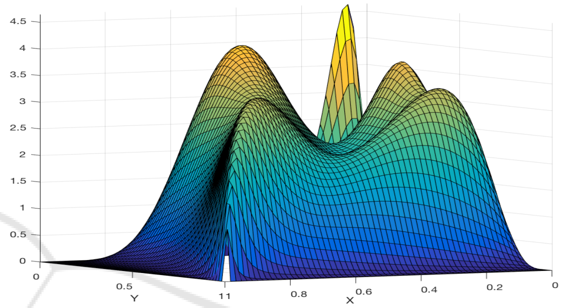

Figure 4: Five-component mixture of bivariate Beta distribution.

of software failures and improving reliability is one of the attractive applications for scientists (Boucher and Badri, 2017), (Kawashima and Mizuno, 2015), (Lyu, 1996), (Koru and Liu, 2005). To tackle this problem, it is critical to define the appropriate metrics to express the attributes of the software modules. There are some metrics (Aleem et al., 2015) for assessing software complexity such as the code size, McCabes cyclomatic and Halsteads complexity (McCabe, 1976). The McCabes metric includes essential, cyclomatic and design complexity and the number of lines of code. While the Halsteads metric consists of base and derived measures and line of code (LOC) (McCabe, 1976), (Shihab, 2014). Prediction models are applied to improve and optimize the quality which is translated to customer satisfaction as a significant achievement for the companies. Finite mixture models as flexible statistical solutions and clustering techniques are considered as powerful tools in this area (Oboh and Bouguila, 2017), (Bouguila and Hamza, 2010), (Kawashima and Mizuno, 2015). Our experiment is performed on three datasets from the PROMISE data repository obtained from NASA software projects and its public MDP (Modular toolkit for Data Processing) which are currently used as benchmark datasets in this area of research (NASA, 2004).

The metrics or features of each dataset are five different lines of code measure, three McCabe metrics, four base Halstead measures, eight derived Halstead measures and a branch-count. The datasets are classified by a binary variable to indicate if the module is defective or not. CM1 as the first dataset is a NASA spacecraft instrument software written in "C". KC1 as the second one, is a "C++" dataset raised from system implementing storage management for receiving and processing ground data. The last case, PC1 is developed using "C" considering functions flight software for earth orbiting satellite. To highlight the basic properties of the datasets, Table 1 is created. As it is shown in Table 2 and Table 3, multivariate Beta mixture model (MBMM) has better performance in all three datasets in comparison with Gaussian mixture model (GMM). For CM1, the accuracy of our model is 98.79% while this value for GMM is 85.94% . KC1 has a more accurate result (94.12%) with MBMM than GMM (88.66%). The performance of models for PC1 is similar by 94.13% and 91.79% of accuracy for MBMM and GMM, respectively. The precision and recall follow the same behavior as accuracy. The multivariate Beta mixture model is capable to reach 97.44% precision and 99.55% of recall for PC1 and KC1, respectively. While GMM has the best preci-

Table 1: Software modules defect properties.

Dataset	Language	Instances	Defects
CM1	C	498	49
KC1	C++	2109	326
PC1	C	1109	77

Table 2: Software modules defect results inferred from the confusion matrix of multivariate Beta mixture model.

Dataset	Accuracy	Precision	Recall
CM1	98.79	99.15	99.55
KC1	94.12	94.69	98.31
PC1	94.13	97.44	95.97

sion and recall in PC1 with 96.06% and 95.23%.

4.2 Spam Detection

Spam filtering as our second real application is one of the major research fields in information systems security. Spams or unsolicited bulk emails pose serious threats. As it was mentioned is some literature up to 75–80% of email messages are spam (Blanzieri and Bryl, 2008) which resulted in heavy financial losses of 50 and 130 billion dollars in 2005, respectively (Galati, 2018), (Lugaresi, 2004), (Wang et al., 2018). Considering serious risks and costly consequences, classification and categorization of email (Ozgur and Gungor, 2012), (Amayri and Bouguila, 2009a), (Amayri and Bouguila, 2009b), (Amayri and Bouguila, 2012) have received a lot of attention. Applying machine learning and pattern recognition techniques capability was enhanced compared to hand-made rules (Amayri and Bouguila, 2010), (Bouguila and Amayri, 2009), (Fan and Bouguila, 2013), (Cormack and Lynam, 2007), (Chang and Meek, 2008), (Hershkop and Stolfo, 2005), (Drake, 2004) .

Our experiment was carried out on a challenging spam data set obtained from UCI machine learning repository, created by Hewlett-Packard Labs (UCI, 1999). This dataset contains 4601 instances and 58 attributes (57 continuous input attributes and 1 nominal class label target attribute). 39.4% of email (1813 instances) are spam and 60.6% (2788) are legitimate. The attributes are extracted from a commonly used technique called Bag of Words (BoW) as one of the main information retrieval methods in natural language processing. In this method, each email is presented by its words disregarding grammar. Most of the attributes in spam base dataset indicate whether a particular word or character was frequently occurring in the e-mail. 48 features include the percentage of words in the e-mail that match the word. 6 attributes are extracted from the percentage of characters in the e-mail that match characters. The rest of the features are the average length of uninterrupted sequences of capital letters, the length of the longest uninterrupted sequence of capital letters and the total number of capital letters in the e-mail. The dataset class denotes whether the e-mail was considered spam or not. To evaluate our framework, first the dataset has been reduced to 3626 instances to have a balanced case. Then it was normalized by Equation 31 as our assumption is that all observation values are between zero and one. Table 4 shows the results of our model performance in comparison with Gaussian mixture model considering their confusion matrix. As we can realize from table 4, multivariate Beta mixture model is more accurate (79.92%) and has higher value in terms of precision and recall, 80.6% and 82.74%, respectively.

Table 3: Software modules defect results according to the confusion matrix of Gaussian mixture model.

Dataset	Accuracy	Precision	Recall
CM1	85.94	92.21	90.88
KC1	88.66	93.99	92.69
PC1	91.79	96.06	95.23

Table 4: Spam filtering results to compare the performance of MBMM and GMM.

Mixture model	Accuracy	Precision	Recall
MBMM	79.92	80.6	82.74
GMM	67.81	78.99	68.29

5 CONCLUSION

In this paper, we have developed a clustering technique and a mixture model in order to propose a new approach to model data and improve clustering accuracy. We have mainly proposed our model using a multivariate Beta distribution which has more flexibility. The work addresses the parameters estimation within a deterministic and efficient method using maximum likelihood estimation. After the presentation of our algorithm for parameters estimation, we evaluated the capability of the proposed statistical mixture model in two real attractive domains and applied confusion matrix as a typical evaluation approach to estimate the accuracy, precision and recall and effectiveness of our solution. As the first real world experiment, we considered a popular and critical application in information security engineering about predicting defects in software modules in the context of three NASA datasets. Our clustering algorithm was developed to discover two groupings based on some software complexity metrics. The proposed methodology has been shown to outperform Gaussian mixture model as a classical approach and our offered solution achieved better results in terms of data modeling capabilities and clustering accuracy. The second application was spam detection using the spam base dataset from the UCI repository. The ultimate goal of our extensive study is developing a powerful classifier as a devoted filter to accurately distinguish spam emails from legitimate emails in order to improve the blocking rate of spam emails and decrease the misclassification rate of legitimate emails. Spam filtering solutions presented in this paper generates acceptable, accurate results in comparison with Gaussian mixture model as the results of our algorithm has higher precision and recall. From the outcomes, we can infer that the multivariate Beta mixture model could be a competitive modeling approach for the software defect and spam prediction problems. In other words, we can say that our model produces enhanced clustering results largely due to its model flexibility.

REFERENCES

Aleem, S., Capretz, L. F., and Ahmed, F. (2015). Benchmarking machine learning technologies for software defect detection. 6(3):11–23.

Amayri, O. and Bouguila, N. (2009a). A discrete mixture-based kernel for svms: Application to spam and image categorization. *Artificial Intelligence Review*, 34(1):73–108.

Amayri, O. and Bouguila, N. (2009b). Online spam filtering using support vector machines. *IEEE Symposium on Computers and Communications*, pages 337–340.

Amayri, O. and Bouguila, N. (2010). A study of spam filtering using support vector machines. *Artificial Intelligence Review*, 34(1):173–108.

Amayri, O. and Bouguila, N. (2012). Unsupervised feature selection for spherical data modeling: Application to image-based spam filtering. *International Conference on Multimedia Communications, Services and Security*, pages 13–23.

Bertolino, A. (2007). Software testing research: Achievements, challenges, dream. *Future of Software Engineering*, page 85–103.

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer, New York.

Blanzieri, E. and Bryl, A. (2008). A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review*, 29:63–92.

Boucher, A. and Badri, M. (2017). Predicting fault-prone classes in objectoriented software. page 306–317.

Bouguila, N. and Amayri (2009). A discrete mixturebased kernel for svms: Application to spam and image categorization. *Information Processing and Management*, 45:631–642.

Bouguila, N., W. J. and Hamza, A. (2010). Software modules categorization through likelihood and bayesian analysis of finite dirichlet mixtures. *Applied Statistics*, 37(2):235–252.

Briand, L. C., B. V. and Hetmanski, C. J. (1993). Developing interpretable models with optimized set reduction for identifying high-risk software components. volume 19, page 1028–1044. IEEE Transactions on Software Engineering.

Chang, M., Y. W. and Meek, C. (2008). Partitioned logistic regression for spam filtering. page 97–105. 14th ACM SIGKDD international conference on knowledge discovery and data mining.

Cockriel, W. M. and McDonald, J. B. (2018). Two multivariate generalized beta families, communications in statistics. *Theory and Methods*, 47(23):5688–5701.

Cormack, G. and Lynam, T. (2007). Online supervised spam filter evaluation. *ACMTransactions on Information Systems*, 25(3):1–31.

Diaz-Rozo, J., B. C. and Larranaga, P. (2018). Clustering of data streams with dynamic gaussian mixture models: An iot application in industrial processes. *IEEE Internet of Things Journal*, 5:3533.

Drake, C., O. J. K. E. (2004). Anatomy of a phishing email. *First conference on email and anti- Spam (CEAS)*, 25(3):1–31.

El Emam, K. Benlarbi, S. G. N. and Rai, S. N. (2001). Comparing casebased reasoning classifiers for predicting high risk software components. *Journal of Systems and Software*, 55(3):301–320.

Elguebaly, T. and Bouguila, N. (2013). Finite asymmetric generalized gaussian mixture models learning for infrared object detection. *Computer Vision and Image Understanding*, 117:1659–1671.

Fan, W. and Bouguila, N. (2013). Variational learning of a dirichlet process of generalized dirichlet distributions for simultaneous clustering and feature selection. *Pattern Recognition*, 46:2754–2769.

Fan, W., Bouguila, N., and Ziou, D. (2014). Variational learning of finite dirichlet mixture models using component splitting. *Neurocomputing*, 129:3–16.

Figueiredo, M. and Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396.

Galati, L. (2018). The anatomy of a phishing attack. *Fairfield County Business Journal*, 54:5.

Ganesalingam, S. (1989). Classification and mixture approaches to clustering via maximum likelihood. *Journal of the Royal Statistical Society: Series C (Applied Statistics*, 38(3):455–466.

Gevers, T., S. A. (1999). Color-based object recognition. *Journal of the Royal Statistical Society: Series C (Applied Statistics*, 32(3):453–464.

Giordan, M., W. R. (2015). A comparison of computational approaches for maximum likelihood estimation of the dirichlet parameters on high-dimensional data. *SORT-Statistics and Operations Research Transactions*, 39(1):109–126.

Guha, S., R. R. and Shim, K. (2001). Cure: an efficient clustering algorithm for large databases. *Information Systems*, 26:35–58.

Han, J., K. M. and Pei, J. (2012). *Data mining: concepts and techniques*. Elsevier, Morgan Kaufmann, Amsterdam; Boston.

Hastie, T. and Tibshirani, R. (1996). Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society Series B (Methodological)*, 58(1):155–176.

Hershkop, S. and Stolfo, S. (2005). Combining email models for false positive reduction. *The eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 98–107.

Kawashima, N. and Mizuno, O. (2015). Predicting fault-prone modules by word occurrence in identifiers. *Software Engineering Research, Management and Applications*, page 87–98.

Klauschies, T., Coutinho, R. M., and Gaedke, U. (2018). A beta distribution-based moment closure enhances the reliability of trait-based aggregate models for natural populations and communities. *Ecological Modelling*, 381:46–77.

Koru, A. G. and Liu, H. (2005). Building effective defect prediction models in practice. *IEEE software*, 22(6):23–29.

Lugaresi, N. (2004). European union vs. spam: a legal response. *First conference on email and anti-Spam (CEAS)*.

Luo, Z., He, W., Liwang, M., Huang, L., Zhao, Y., and Geng, J. (2017). Real-time detection algorithm of abnormal behavior in crowds based on gaussian mixture model. *12th International Conference on Computer Science and Education (ICCSE)*, 20:183.

Lyu, M. R. e. a. (1996). Handbook of software reliability engineering. *IEEE computer society press CA*, 222:183.

Malhotra, R. and Jain, A. (2012). Fault prediction using statistical and machine learning methods for improving software quality. *Journal of Information Processing Systems*, 8(2):1241–262.

McCabe, T. J. (1976). A complexity measure. *IEEE Transactions on software Engineering*, 8(4):308–320.

McCabe, T. J. (2015). *Mixture Models in Statistics*. Elsevier Ltd.

McLachlan, G. and Krishnan, T. (2008). *The EM algorithm and extensions*. Wiley-Interscience.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. New York: Wiley.

NASA (2004). *PROMISE Software Engineering Repository data set*. http://promise.site.uottawa.ca/SERepository/datasetspage.html.

Oboh, S. and Bouguila, N. (2017). Unsupervised learning of finite mixtures using scaled dirichlet distribution and its application to software modules categorization. volume 8, page 1085–1090.

Olkin, I. and Liu, R. (2003). A bivariate beta distribution, statistics and probability letters. volume 62, pages 407–412.

Olkin, I. and Trikalinos, T. A. (2015). Constructions for a bivariate beta distribution. volume 96, pages 54–60.

Ozgur, L. and Gungor, T. (2012). Optimization of dependency and pruning usage in text classification. volume 15, pages 45–58.

Shihab, E. (2014). Practical software quality prediction, in software maintenance and evolution (icsme). page 639–644.

UCI, R. (1999 (accessed 2 August 1999)). *Spambase UCI Repository data set*. https://archive.ics.uci.edu/ml/machine-learningdatabases/spambase/spambase.data.

Wang, S., Zhang, X., Cheng, Y., Jiang, F., Yu, W., and Peng, J. (2018). A fast content-based spam filtering algorithm with fuzzy-svm and k-means. pages 301–307.

Wentao, F., Bouguila, N., and Ziou, D. (2013). Unsupervised hybrid feature extraction selection for high-dimensional non-gaussian data clustering with variational inference. *IEEE Transactions on Knowledge and Data Engineering*, 25(7):1670–1685.

Zhou, J. H., P. C. K. Y. W. (2017). Gaussian mixture model for new fault categories diagnosis. pages 1–6.