A Model-based Approach for Text Clustering with Outlier Detection

Jianhua Yin[†], Jianyong Wang[†‡]
[†]Department of Computer Science, Tsinghua National Laboratory for Information Science and Technology (TNList),
Tsinghua University, Beijing, China
[‡]Jiangsu Collaborative Innovation Center for Language Ability, Jiangsu Normal University, Xuzhou 221009, China
jhyin12@gmail.com, jianyong@tsinghua.edu.cn

Abstract—**Text clustering is a challenging problem due to the high-dimensional and large-volume characteristics of text datasets. In this paper, we propose a collapsed Gibbs Sampling algorithm for the Dirichlet Process Multinomial Mixture model for text clustering (abbr. to GSDPMM) which does not need to specify the number of clusters in advance and can cope with the high-dimensional problem of text clustering. Our extensive experimental study shows that GSDPMM can achieve significantly better performance than three other clustering methods and can achieve high consistency on both long and short text datasets. We found that GSDPMM has low time and space complexity and can scale well with huge text datasets. We also propose some novel and effective methods to detect the outliers in the dataset and obtain the representative words of each cluster.**

I. Introduction

Text clustering [1] is a widely studied problem with many applications such as document organization, summarization, classification, and browsing. The biggest challenge of text clustering is the high-dimensional problem of the text data. This corresponds to the fact that the text lexicon is rather large (with the order of 10^5), while each document contains only a small number of words (with the order of 10^2). As discussed in [2], the similarity between high-dimensional vectors will lose their effectiveness and statistical significance because of irrelevant attributes, this is also called the dimensionality curse.

In [3], we introduced a collapsed Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model (GSDMM) to deal with the short text clustering problem. GSDMM defines the probability of a document choosing each cluster with a metric similar to that of the Naive Bayes Classifier [4]. In detail, GSDMM evaluates the frequency of the words in a document appearing in each cluster as some kind of similarity between the document and the clusters. As GSDMM only considers the words in the current document, it will not be affected by other irrelevant words in the vocabulary, and can deal with the high-dimensional problem of text clustering. GSDMM can also infer the number of clusters automatically as long as the initial number of clusters is larger than the true number of clusters in the dataset. However, it is difficult to set a proper initial number of clusters K for GSDMM as we do not know the true number, as a result, we may have to choose a really large K to ensure safety which will result in the complexity of GSDMM to be large.

In this paper, we use the Dirichlet Process Multinomial Mixture (abbr. to DPMM) model to model the generative process of documents, which can be considered as an infinite extension of the Dirichlet Multinomial Mixture (DMM) model. The DPMM model assumes that there is an infinite number of latent clusters, but only a finite number of them are used to generate the observed documents. The advantage of the DPMM model is that the number of clusters K is not required as an input parameter but grows with the data. Yu et al. [5] proposed a blocked Gibbs sampling algorithm for an approximation of the DPMM model, however, their method is slow to converge. Zhang [6] used the DPMM model for online document clustering and used empirical Bayes method [7] to estimate those parameters instead of taking a full Bayesian approach. Neal [8] introduced a collapsed Gibbs Sampling algorithm for the general Dirichlet Process Mixture (DPM) model. Specifically, MacEachern [9] and Neal [10] proposed the collapsed Gibbs Sampling algorithm for the Dirichlet Process Normal Mixture model and the Dirichlet Process Bernoulli Mixture model, respectively. However, these mixture models are not appropriate for text clustering. Differently, we are the first to propose the collapsed Gibbs Sampling algorithm for the Dirichlet Process Multinomial Mixture model (abbr. to GSDPMM) for text clustering, which is fast to converge and can scale well with huge text datasets. GSDPMM can infer the number of clusters automatically and can deal with the high-dimensional problem of text clustering. We also propose some novel and effective methods to detect the outliers in the dataset and obtain the representative words of each cluster.

In the experimental study, we compared GSDPMM with K-means [11], LDA [12], and GSDMM [3]. We found that GSDPMM can achieve significantly better performance than these methods on both long and short text datasets. We run each algorithm 20 times on each dataset and found that the standard deviations of the results of GSDPMM are quite small, which indicates that GSDPMM has high consistency.

The contributions of this paper are summarized as follows:

- We are among the first to use the Dirichlet Process Multinomial Mixture (DPMM) model for text clustering, and our experimental study has validated its effectiveness. We find it can cope with the high-dimensional problem of text clustering.
- To the best of our knowledge, we are the first to propose the collapsed Gibbs sampling algorithm for the DPMM model for text clustering, which can achieve very good performance on both short and long text clustering. Meanwhile, this algorithm has low time and space complexity and can infer the number of

978-1-5090-2020-1/16/$31.00 © 2016 IEEE

clusters automatically.

- We propose some novel and effective methods to detect the outliers in the dataset and obtain the representative words of each cluster.

The remainder of this paper is organized as follows. In Section II, we review the related work for text clustering. In Section III, we introduce the Dirichlet Process Multinomial Mixture (DPMM) model, then we introduce the GSDPMM algorithm in Section IV. In Section V, we describe the design of experiments to evaluate the performance GSDPMM compared with other three clustering models, and study the special properties of GSDPMM. We finally present conclusions and future work in Section VI.

II. Related Work

The clustering methods can be generally categorized into the following two categories: similarity-based clustering and model-based clustering.

A. Similarity-based Clustering

Similarity-based clustering methods mostly use the vector space model to represent data points and choose some similarity metric to measure the similarity between data points.

Partitional algorithms like K-means [11] and K-medoids [13] are one kind of similarity-based clustering methods that formulate clustering as an optimization problem: find K cluster centers and assign the data points to the nearest cluster center, so that the squared distances from the cluster are minimized. The advantage of partitional algorithms is that they are efficient and easy to implement. However, the number of clusters need to be specified in advance, and they are sensitive to the initialization.

Hierarchical algorithms [14] are another kind of similarity-based clustering methods that recursively find nested clusters either in agglomerative mode or divisive mode. The hierarchical algorithms are particularly useful to support a variety of searching methods because they naturally create a tree-like hierarchy which can be leveraged for the search process [15]. The drawback of hierarchical algorithms is that they need to assume the true number of clusters or a similarity threshold, and they cannot scale well with large data sets.

Density-based algorithms [16][17] define the clusters as areas of higher density than the remainder of the dataset. The advantage of density-based algorithms is that they do not need to specify the number of clusters in advance, and can detect the outliers of the dataset. However, they have limitations in handling high-dimensional data like text. Because the feature space of high-dimensional data is usually sparse, density-based algorithms have difficulty to distinguish high-density regions from low-density regions [18].

B. Model-based Clustering

Model-based methods assume data points are generated by a mixture model, and then use techniques like EM [19] or Gibbs Sampling [20] to estimate the parameters of the mixture model, so as to obtain the clustering results.

The most widely used model-based clustering method is the Gaussian Mixture Model (GMM) [21], which assumes that data points are generated by a mixture of Gaussian distributions. However, the complexity of GMM is too large for high-dimensional data like text. Nigam et al. [22] proposed an EM-based algorithm for the Dirichlet Multinomial Mixture (DMM) model for classification with both labeled and unlabeled documents. When only unlabeled documents are provided, this algorithm turns out to be a clustering model. In [3], we introduced a collapsed Gibbs Sampling algorithm for the DMM model (GSDMM) to deal with the short text clustering problem. GSDMM can cope with the high-dimensional problem of short texts, and can also infer the number of clusters automatically as long as the initial number of clusters is larger than the true number of clusters. However, it is difficult to set a proper initial number of clusters K for GSDMM as we do not know the true number of clusters, as a result, we may have to choose a really large K to ensure safety which will result in the complexity of GSDMM to be large.

Yu et al. [5] proposed a blocked Gibbs sampling algorithm for an approximation of the Dirichlet Process Multinomial Mixture (DPMM) model, however, their method is slow to converge. Zhang [6] used the DPMM model for online document clustering and used empirical Bayes method [7] to estimate the parameters instead of taking a full Bayesian approach. Neal [8] introduced a collapsed Gibbs Sampling algorithm for the general Dirichlet Process Mixture (DPM) model. Specifically, MacEachern [9] and Neal [10] proposed the collapsed Gibbs Sampling algorithm for the Dirichlet Process Normal Mixture model and the Dirichlet Process Bernoulli Mixture model, respectively. However, these mixture models are not appropriate for text clustering. Differently, we are the first to propose the collapsed Gibbs Sampling algorithm for the DPMM model (GSDPMM) for text clustering, which is fast to converge and can achieve very good performance on both short and long text clustering.

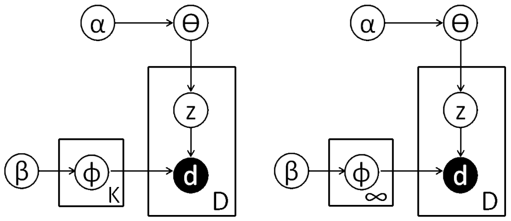
Topic models like LDA [12] and PLSA [23] are probabilistic generative models that can model texts and identify latent semantics underlying the text collection. LDA assumes that each document is generated by choosing a distribution over topics and then choosing each word in the document from a topic selected according to this distribution. Teh et al. [24] proposed the Hierarchical Dirichlet Process (HDP) model as an extension of LDA which can automatically determine the appropriate number of topics needed. Different from LDA, we assume that each document is generated by only one topic (cluster) and the words in the document are generated independently when the document's cluster assignment is known. We find that this model is more effective for the text clustering task, and our extensive experimental study shows that GSDPMM can achieve significantly better performance than LDA on both long and short text datasets.

III. The DPMM Model

The Dirichlet Process Multinomial Mixture (DPMM) model can be considered as an infinite extension of the Dirichlet Multinomial Mixture (DMM) model [22]. The graphical representation of the DMM model is shown in Figure 1a, which

TABLE I: NOTATIONS

V	size of the vocabulary
D	number of documents in the corpus
$\bar{L}$	average length of documents
$\vec{d}$	documents in the corpus
$\vec{z}$	cluster assignments of each document
I	number of iterations
m_z	number of documents in cluster z
n_z	number of words in cluster z
n_z^w	number of occurrences of word w in cluster z
N_d	number of words in document d
N_d^w	number of occurrences of word w in document d



(a) DMM (b) DPMM

Fig. 1: Graphical Models

is equivalent to the following generative process:

$$\Theta|\alpha \sim Dir(\alpha) \tag{III.1}$$

$$z_d|\Theta \sim Mult(\Theta) \qquad d = 1, ..., D \tag{III.2}$$

$$\Phi_k|\beta \sim Dir(\beta) \qquad k = 1, ..., K \tag{III.3}$$

$$d|z_d, \{\Phi_k\}_{k=1}^{K} \sim p(d|\Phi_{z_d}) \tag{III.4}$$

Here, "$X \sim S$" means "X is distributed according to S", so the right side is a specification of distribution.

The DMM model is a probabilistic generative model for documents, and embodies two assumptions about the generative process: (1) the documents are generated by a mixture model [25], and (2) there is a one-to-one correspondence between mixture components and clusters. When generating document d, the DMM model first selects a mixture component (cluster) z_d according to the mixture weights (weights of clusters), Θ, in Equation III.2. Then document d is generated by the selected mixture component (cluster) from distribution $p(d|\Phi_{z_d})$ in Equation III.4. The weight vector of the clusters, Θ, is generated by a Dirichlet distribution with a hyper-parameter α, as in Equation III.1. The cluster parameters Φ_z are also generated by a Dirichlet distribution with a hyper-parameter β, as in Equation III.3.

The DMM model becomes the DPMM model when we let K go to infinity, whose graphical representation is shown in Figure 1b. The generative process of the DPMM model is as follows:

$$\Theta|\alpha \sim GEM(1, \alpha) \tag{III.5}$$

$$z_d|\Theta \sim Mult(\Theta) \qquad d = 1, ..., D \tag{III.6}$$

$$\Phi_k|\beta \sim Dir(\beta) \qquad k = 1, ..., \infty \tag{III.7}$$

$$d|z_d, \{\Phi_k\}_{k=1}^{\infty} \sim p(d|\Phi_{z_d}) \tag{III.8}$$

In the DPMM model, the Dirichlet prior $Dir(\alpha)$ is replaced by a stick-breaking construction, $\Theta \sim GEM(1, \alpha)$ [26]. Different from the DMM model, the number of clusters in the DPMM model is no longer a fixed value K.

In this paper, the probability of document d generated by cluster z_d is defined as follows:

$$p(d|\Phi_{z_d}) = \prod_{w \in d} Mult(w|\Phi_{z_d}) \tag{III.9}$$

Here, we make the Naive Bayes assumption: the words in a document are generated independently when the document's cluster assignment z_d is known. We also assume that the probability of a word is independent of its position within the document.

IV. APPROACH

A. Choosing an Existing Cluster

Before discussing the collapsed Gibbs sampling algorithm for the Dirichlet Process Multinomial Mixture (DPMM) model, we first look at the case of the Dirichlet Multinomial Mixture (DMM) model. The documents $\vec{d} = \{d_i\}_{i=1}^{D}$ are observed and the cluster assignments $\vec{z} = \{z_i\}_{i=1}^{D}$ are latent. Because conjugate priors are used, we can integrate out Θ and Φ. Then we can sample z_d from distribution $p(z_d = z|\vec{z}_{\neg d}, \vec{d}, \alpha, \beta)$, which is the probability of document d choosing cluster z given the information of other documents and their cluster assignments. Factorize this conditional distribution, we have:

$$\begin{aligned} & p(z_d = z|\vec{z}_{\neg d}, \vec{d}, \alpha, \beta) \\ & \propto p(z_d = z|\vec{z}_{\neg d}, \vec{d}_{\neg d}, \alpha, \beta) p(d|z_d = z, \vec{z}_{\neg d}, \vec{d}_{\neg d}, \alpha, \beta) \end{aligned} \tag{IV.1}$$

$$\propto p(z_d = z|\vec{z}_{\neg d}, \alpha) p(d|z_d = z, \vec{d}_{z,\neg d}, \beta) \tag{IV.2}$$

Here, we use the Bayes Rule in Equation IV.1, and apply the properties of D-Separation [19] in Equation IV.2. The first term in Equation IV.2 indicates the probability of document d choosing cluster z when we know the cluster assignments of other documents. The second term in Equation IV.2 can be considered as a predictive probability of document d given $\vec{d}_{z,\neg d}$, i.e., the other documents currently assigned to cluster z.

We will first derive the first term in Equation IV.2 as follows:

$$p(z_d = z|\vec{z}_{\neg d}, \alpha)$$

$$= \int p(z_d = z, \Theta|\vec{z}_{\neg d}, \alpha) d\Theta \tag{IV.3}$$

$$= \int p(\Theta|\vec{z}_{\neg d}, \alpha) p(z_d = z|\vec{z}_{\neg d}, \Theta, \alpha) d\Theta \tag{IV.4}$$

$$= \int p(\Theta|\vec{z}_{\neg d}, \alpha) p(z_d = z|\Theta) d\Theta \tag{IV.5}$$

Here, Equation IV.3 exploits the Sum Rule of Probability [19]. We use the Product Rule of Probability [19] in Equation IV.4 and apply the properties of D-Separation in Equation IV.5. The first term in Equation IV.5 is the posterior distribution of Θ and the second term in Equation IV.5 is the following multinomial distribution: $Mult(z_d = z|\Theta) = \Theta_z$.

Next, we derive the first term in Equation IV.5 as follows:

$$
\begin{aligned}
&p(\Theta|\vec{z}_{\neg d},\alpha)\\
&= \frac{p(\Theta|\alpha)p(\vec{z}_{\neg d}|\Theta)}{\int p(\Theta|\alpha)p(\vec{z}_{\neg d}|\Theta)d\Theta} \qquad \text{(IV.6)}\\
&= \frac{\frac{1}{\Delta(\alpha)}\prod_{k=1}^{K}\Theta_k^{\alpha/K-1}\prod_{k=1}^{K}\Theta_k^{m_{k,\neg d}}}{\int \frac{1}{\Delta(\alpha)}\prod_{k=1}^{K}\Theta_k^{\alpha/K-1}\prod_{k=1}^{K}\Theta_k^{m_{k,\neg d}}d\Theta} \qquad \text{(IV.7)}\\
&= \frac{1}{\Delta(\vec{m}_{\neg d}+\alpha/K)}\prod_{k=1}^{K}\Theta_k^{m_{k,\neg d}+\alpha/K-1} \qquad \text{(IV.8)}\\
&= Dir(\Theta|\vec{m}_{\neg d}+\alpha/K) \qquad \text{(IV.9)}
\end{aligned}
$$

Here, Equation IV.6 exploits the Bayes Rule.

Then, we can derive the first term in Equation IV.2 as follows:

$$
\begin{aligned}
&p(z_d=z|\vec{z}_{\neg d},\alpha)\\
&= \int Dir(\Theta|\vec{m}_{\neg d}+\alpha/K)Mult(z_d=z|\Theta)d\Theta \qquad \text{(IV.10)}\\
&= \int \frac{1}{\Delta(\vec{m}_{\neg d}+\alpha/K)}\Theta_z\prod_{k=1,k\neq z}^{K}\Theta_k^{m_{k,\neg d}+\alpha/K-1}d\Theta \qquad \text{(IV.11)}\\
&= \frac{\Delta(\vec{m}+\alpha/K)}{\Delta(\vec{m}_{\neg d}+\alpha/K)} \qquad \text{(IV.12)}\\
&= \frac{\prod_{k=1}^{K}\Gamma(m_k+\alpha/K)}{\Gamma(\sum_{k=1}^{K}(m_k+\alpha/K))}\frac{\Gamma(\sum_{k=1}^{K}(m_{k,\neg d}+\alpha/K))}{\prod_{k=1}^{K}\Gamma(m_{k,\neg d}+\alpha/K)} \qquad \text{(IV.13)}\\
&= \frac{\Gamma(m_{z,\neg d}+\alpha/K+1)}{\Gamma(m_{z,\neg d}+\alpha/K)}\frac{\Gamma(D-1+\alpha)}{\Gamma(D+\alpha)} \qquad \text{(IV.14)}\\
&= \frac{m_{z,\neg d}+\alpha/K}{D-1+\alpha} \qquad \text{(IV.15)}
\end{aligned}
$$

In Equation IV.12, we adopt the Δ function used in [27], which is defined as $\Delta(\vec{\alpha}) = \frac{\prod_{k=1}^{K}\Gamma(\alpha)}{\Gamma(\sum_{k=1}^{K}\alpha)}$. Using the property of Γ function: $\Gamma(x+1) = x\Gamma(x)$, we can get Equation IV.15 from Equation IV.14. In Equation IV.15, $m_{z,\neg d}$ is the number of documents in cluster z without considering document d, and D is the total number of documents in the dataset. Equation IV.15 indicates that document d will tend to choose larger clusters when we only consider the cluster assignments of the other documents.

The second term in Equation IV.2 considers the words of the documents in each cluster which actually indicates some kind of similarity between document d and cluster z.

$$
\begin{aligned}
&p(d|z_d=z,\vec{d}_{z,\neg d},\beta)\\
&= \int p(d,\Phi_z|z_d=z,\vec{d}_{z,\neg d},\beta)d\Phi_z \qquad \text{(IV.16)}\\
&= \int p(\Phi_z|z_d=z,\vec{d}_{z,\neg d},\beta)p(d|\Phi_z,z_d=z,\vec{d}_{z,\neg d},\beta)d\Phi_z \qquad \text{(IV.17)}\\
&= \int p(\Phi_z|\vec{d}_{z,\neg d},\beta)p(d|\Phi_z,z_d=z)d\Phi_z \qquad \text{(IV.18)}
\end{aligned}
$$

Here, Equation IV.16 exploits the Sum Rule of Probability [19]. We use the Product Rule of Probability in Equation IV.17 and apply the properties of D-Separation [19] to obtain Equation IV.18.

Next, we try to derive the first term in Equation IV.18 as follows:

$$
\begin{aligned}
&p(\Phi_z|\vec{d}_{z,\neg d},\beta)\\
&= \frac{p(\Phi_z|\beta)p(\vec{d}_{z,\neg d}|\Phi_z)}{\int p(\Phi_z|\beta)p(\vec{d}_{z,\neg d}|\Phi_z)d\Phi_z} \qquad \text{(IV.19)}\\
&= \frac{\frac{1}{\Delta(\beta)}\prod_{t=1}^{V}\Phi_{z,t}^{\beta-1}\prod_{t=1}^{V}\Phi_{k,t}^{n_{z,\neg d}^{t}}}{\int \frac{1}{\Delta(\beta)}\prod_{t=1}^{V}\Phi_{z,t}^{\beta-1}\prod_{t=1}^{V}\Phi_{k,t}^{n_{z,\neg d}^{t}}d\Phi_z} \qquad \text{(IV.20)}\\
&= \frac{1}{\Delta(\vec{n}_z+\beta)}\prod_{t=1}^{V}\Phi_{z,t}^{n_{z,\neg d}^{t}+\beta-1} \qquad \text{(IV.21)}\\
&= Dir(\Phi_z|\vec{n}_{z,\neg d}+\beta) \qquad \text{(IV.22)}
\end{aligned}
$$

Here, Equation IV.19 exploits the Bayes Rule.

Then, we can obtain the second term in Equation IV.2 as follows:

$$
\begin{aligned}
&p(d|z_d=z,\vec{d}_{z,\neg d},\beta)\\
&= \int Dir(\Phi_z|\vec{n}_{z,\neg d}+\beta)\prod_{w\in d}Mult(w|\Phi_z)d\Phi_z \qquad \text{(IV.23)}\\
&= \int \frac{1}{\Delta(\vec{n}_{z,\neg d}+\beta)}\prod_{t=1}^{V}\Phi_{z,t}^{n_{z,\neg d}^{t}+\beta-1}\prod_{w\in d}\Phi_{z,w}^{n_d^w}d\Phi_z \qquad \text{(IV.24)}\\
&= \frac{\Delta(\vec{n}_z+\beta)}{\Delta(\vec{n}_{z,\neg d}+\beta)} \qquad \text{(IV.25)}\\
&= \frac{\prod_{t=1}^{V}\Gamma(n_z^t+\beta)}{\Gamma(\sum_{t=1}^{V}(n_z^t+\beta))}\frac{\Gamma(\sum_{t=1}^{V}(n_{z,\neg d}^t+\beta))}{\prod_{t=1}^{V}\Gamma(n_{z,\neg d}^t+\beta)} \qquad \text{(IV.26)}\\
&= \frac{\prod_{w\in d}\prod_{j=1}^{N_d^w}(n_{z,\neg d}^w+\beta+j-1)}{\prod_{i=1}^{N_d}(n_{z,\neg d}+V\beta+i-1)} \qquad \text{(IV.27)}
\end{aligned}
$$

Because the Γ function has the following property: $\frac{\Gamma(x+m)}{\Gamma(x)} = \prod_{i=1}^{m}(x+i-1)$, we can get Equation IV.27 from Equation IV.26. In Equation IV.27, N_d^w and N_d are the number of occurrences of word w in document d and the total number of words in document d, respectively, and $N_d = \sum_{w\in d}N_d^w$. Besides, $n_{z,\neg d}^w$ and $n_{z,\neg d}$ are the number of occurrences of word w in cluster z and the total number of words in cluster z without considering document d, respectively, and $n_{z,\neg d} = \sum_{w=1}^{V}n_{z,\neg d}^w$. We can notice that Equation IV.27 actually evaluates some kind of similarity between document d and cluster z, and document d will tend to choose a cluster whose documents share more words with it.

Finally, we have the probability of document d choosing cluster z_d given the information of other documents and their cluster assignments as follows:

$$
\begin{aligned}
&p(z_d=z|\vec{z}_{\neg d},\vec{d},\alpha,\beta)\\
&\propto \frac{m_{z,\neg d}+\alpha/K}{D-1+\alpha}\frac{\prod_{w\in d}\prod_{j=1}^{N_d^w}(n_{z,\neg d}^w+\beta+j-1)}{\prod_{i=1}^{N_d}(n_{z,\neg d}+V\beta+i-1)} \qquad \text{(IV.28)}
\end{aligned}
$$

We can generalize the DMM model to the DPMM model by letting K go to infinity. By doing so, the probability of document d choosing one of the existing K clusters evolves from Equation IV.28 to Equation IV.29 as follows:

$$
\begin{aligned}
&p(z_d=z|\vec{z}_{\neg d},\vec{d},\alpha,\beta)\\
&\propto \frac{m_{z,\neg d}}{D-1+\alpha}\frac{\prod_{w\in d}\prod_{j=1}^{N_d^w}(n_{z,\neg d}^w+\beta+j-1)}{\prod_{i=1}^{N_d}(n_{z,\neg d}+V\beta+i-1)} \qquad \text{(IV.29)}
\end{aligned}
$$

The first part of Equation IV.29 relates to Rule 1 of GSDPMM (Choose a cluster with more documents), as it will have larger value when $m_{z,\neg d}$ (the number of documents in cluster z) is larger. This is also known as the "richer gets richer" or "clustering tendency", which will lead larger clusters to get larger [26]. As a result, the first part of Equation IV.29 conforms to the completeness objective of clustering (all members of a ground true group are assigned to the same cluster) [28].

The second part of Equation IV.29 relates to Rule 2 of GSDPMM (Choose a cluster whose documents share more words with the current document), which is actually a product of N_d parts that correspond to the N_d words of document d. For each word w in document d, the corresponding part measures the fraction of the occurrences of word w in cluster z. When cluster z has more documents that share same words with document d, the second part of Equation IV.29 will be larger, and document d will be more likely to choose cluster z. As a result, the second part of Equation IV.29 conforms to the homogeneity objective of clustering (each cluster contains only members of a single ground true group) [28].

GSDPMM can use parameter β to balance the two parts of Equation IV.29, in other words, to balance the completeness and homogeneity of the clustering results. When β is larger, the second part of Equation IV.29 is less sensitive to $n^w_{z,\neg d}$ (the number of words in cluster z without considering document d), and its influence to Equation IV.29 will be smaller. On the other hand, the influence of the first part of Equation IV.29 will get larger. In other words, GSDPMM will focus more on the completeness objective when β is larger.

In addition, notice that all the information we need to compute Equation IV.29 are $m_{z,\neg d}$ (the number of documents in cluster z), $n_{z,\neg d}$ (the number of words in cluster z without considering document d), and $n^w_{z,\neg d}$ (the number of occurrences of word w in cluster z without considering document d), we only need to update these values before and after a new sample z_d is drawn with complexity linear to the length of document d. In Section IV-C, we will discuss the time and space complexity of GSDPMM, and show that they are $O(D\bar{L})$ and $O(KD\bar{L})$ respectively, where $\bar{L}$ is the average length of the documents.

B. Choosing a New Cluster

We denote a new cluster as $K+1$, and derive the conditional probability of document d choosing a new cluster as follows:

$$\begin{aligned}
&p(z_d = K+1|\vec{z}_{\neg d}, \vec{d}, \alpha, \beta)\\
&\propto p(z_d = K+1|\vec{z}_{\neg d}, \vec{d}_{\neg d}, \alpha, \beta)p(d|z_d = K+1, \vec{z}_{\neg d}, \vec{d}_{\neg d}, \alpha, \beta) && \text{(IV.30)}\\
&= p(z_d = K+1|\vec{z}_{\neg d}, \alpha)p(d|z_d = K+1, \beta) && \text{(IV.31)}
\end{aligned}$$

Here, we use the Bayes Rule to obtain Equation IV.30, and apply the properties of D-Separation [19] to get Equation IV.31. The first term in Equation IV.31 indicates the probability of document d choosing a new cluster when we know the cluster assignments of other documents. The second term in Equation IV.31 can be considered as the predictive probability of document d being generated by the new cluster.

We can derive the first term in Equation IV.31 as follows:

$$\begin{aligned}
&p(z_d = K+1|\vec{z}_{\neg d}, \alpha)\\
&= 1 - \sum_{k=1}^{K} p(z_d = k|\vec{z}_{\neg d}, \alpha) && \text{(IV.32)}\\
&= 1 - \frac{\sum_{k=1}^{K} m_{k,\neg d}}{D-1+\alpha} && \text{(IV.33)}\\
&= 1 - \frac{D-1}{D-1+\alpha} && \text{(IV.34)}\\
&= \frac{\alpha}{D-1+\alpha} && \text{(IV.35)}
\end{aligned}$$

Then, we derive the second term in Equation IV.31 as follows:

$$\begin{aligned}
&p(d|z_d = K+1, \beta)\\
&= \int p(d, \Phi_{K+1}|z_d = K+1, \beta)d\Phi_{K+1} && \text{(IV.36)}\\
&= \int p(\Phi_{K+1}|z_d = K+1, \beta)p(d|\Phi_{K+1}, z_d = K+1, \beta)d\Phi_{K+1} && \text{(IV.37)}\\
&= \int p(\Phi_{K+1}|\beta)p(d|\Phi_{K+1}, z_d = K+1)d\Phi_{K+1} && \text{(IV.38)}\\
&= \int Dir(\Phi_{K+1}|\beta) \prod_{w \in d} Mult(w|\Phi_{K+1})d\Phi_{K+1} && \text{(IV.39)}\\
&= \int \frac{1}{\Delta(\beta)} \prod_{t=1}^{V} \Phi_{K+1,t}^{\beta-1} \prod_{w \in d} \Phi_{K+1,w}^{N_d^w} d\Phi_{K+1} && \text{(IV.40)}\\
&= \frac{\Delta(\vec{n}_{K+1} + \beta)}{\Delta(\beta)} && \text{(IV.41)}\\
&= \frac{\prod_{t=1}^{V} \Gamma(n_{K+1}^t + \beta)}{\Gamma(\sum_{t=1}^{V}(n_{K+1}^t + \beta))} \frac{\Gamma(\sum_{t=1}^{V} \beta)}{\prod_{t=1}^{V} \Gamma(\beta)} && \text{(IV.42)}\\
&= \frac{\prod_{w \in d} \prod_{j=1}^{N_d^w}(\beta + j - 1)}{\prod_{i=1}^{N_d}(V\beta + i - 1)} && \text{(IV.43)}
\end{aligned}$$

Here, Equation IV.36 uses the Sum Rule of Probability and Equation IV.37 exploits the Product Rule of Probability. We apply the properties of D-Separation [19] to get Equation IV.39. Because the Γ function has the following property: $\frac{\Gamma(x+m)}{\Gamma(x)} = \prod_{i=1}^{m}(x+i-1)$, we can get Equation IV.43 from Equation IV.42.

Now, we have the probability distribution of a document choosing a new cluster as follows:

$$\begin{aligned}
&p(z_d = K+1|\vec{z}_{\neg d}, \vec{d}, \alpha, \beta)\\
&\propto \frac{\alpha}{D-1+\alpha} \frac{\prod_{w \in d} \prod_{j=1}^{N_d^w}(\beta + j - 1)}{\prod_{i=1}^{N_d}(V\beta + i - 1)} && \text{(IV.44)}
\end{aligned}$$

where D is the total number of documents and V is the size of the vocabulary. N_d^w and N_d are the number of occurrences of word w in document d and the total number of words in document d, respectively, and $N_d = \sum_{w \in d} N_d^w$.

The first part of Equation IV.44 relates to Rule 1 of GSDPMM (Choose a cluster with more documents), and α is the pseudo number of documents in the new cluster. The second part of Equation IV.44 relates to Rule 2 of GSDPMM (Choose a cluster whose documents share more words with the current document), and β is the pseudo number of occurrences of each word in the new cluster. When α is larger, the documents are more likely to choose a new cluster. In Section V-D, we show that GSDPMM can detect the outliers in the documents.

C. The GSDPMM Algorithm

The detail of the collapsed Gibbs Sampling for the DPMM model (abbr. to GSDPMM) is shown in Algorithm 1, and the meaning of the variables is shown in Table I. The input for the GSDPMM algorithm is D documents, $\vec{d} = \{d_i\}_{i=1}^{D}$, and the output is the number of clusters K and the cluster assignment for each document, $\vec{z} = \{z_i\}_{i=1}^{D}$.

We assign all documents in a single cluster in the initialization step, then we traverse the documents for I iterations. In each iteration, each document will re-choose one of the existing clusters or a new cluster in turn. As this process goes on, some new clusters will be generated and finally the documents will be clustered well into these clusters. In Section V-E, we found that GSDPMM can achieve good and stable performance with only five iterations on three different datasets, which means GSDPMM can converge fast.

GSDPMM uses only four large data structures: n_z^w having dimension KV (number of occurrences of word w in each cluster), n_z having dimension K (number of words in each cluster), m_z having dimension K (number of documents in each cluster), $\vec{z}$ having dimension D (cluster assignments of each document). We can see none of the above structures needs much space. Actually, when dealing with huge datasets, GSDPMM spends most space to store the words of each document with complexity $O(D\bar{L})$, where $\bar{L}$ is the average length of the documents. This is an advantage of GSDPMM compared to methods that use the vector space model [29] to represent the documents. Because the vector space model needs to represent each document as a vector of size V which is the size of the dictionary. Therefore, the space complexity of methods using the vector space model is $O(DV)$, which is much larger than that of GSDPMM, because V is often larger than 10^5 in text datasets.

In each iteration, we compute the probability of document d choosing each of the K existing clusters using Equation IV.29, as well as the probability of document d choosing a new cluster using Equation IV.44. Then we sample a cluster for document d from a multinomial distribution with the above $K + 1$ probabilities as parameters. The time complexity of Equation IV.29 and Equation IV.44 are both linear to the average length of documents, $\bar{L}$. Therefore, the time complexity of each iteration of GSDPMM is $O(KD\bar{L})$.

Algorithm 1: GSDPMM

```
Data: Document vector $\vec{d}$.
Result: Number of clusters $K$, cluster assignments of
        each document $\vec{z}$.
begin
    //Initialization
    $K = K_0$ (Default $K_0 = 1$)
    Zero all count variables $m_z, n_z$, and $n_z^w$
    for each document $d \in [1, D]$ do
        Sample a cluster for $d$:
        $z_d \leftarrow z \sim Multinomial(1/K)$
        $m_z \leftarrow m_z + 1$ and $n_z \leftarrow n_z + N_d$
        for each word $w \in d$ do
            $n_z^w \leftarrow n_z^w + N_d^w$
    //Collapsed Gibbs Sampling
    for $i \in [1, I]$ do
        for each document $d \in [1, D]$ do
            Record the current cluster of $d$: $z = z_d$
            $m_z \leftarrow m_z - 1$ and $n_z \leftarrow n_z - N_d$
            for each word $w \in d$ do
                $n_z^w \leftarrow n_z^w - N_d^w$
            if $n_z == 0$ then
                //Remove the empty cluster
                $K \leftarrow K - 1$
                Re-arrange cluster indices so that
                $1, ..., K$ are active (i.e., non-empty);
            Compute the probability of document $d$
            choosing each of the $K$ existing clusters
            with Equation IV.29 or a new cluster with
            Equation IV.44;
            Sample cluster index $z$ for document $d$ from
            a multinomial distribution with the above
            $K + 1$ probabilities as parameters;
            if $z \in [1, K]$ then
                //An existing cluster is chosen
                $m_z \leftarrow m_z + 1$ and $n_z \leftarrow n_z + N_d$
                for each word $w \in d$ do
                    $n_z^w \leftarrow n_z^w + N_d^w$
            else
                //A new cluster is chosen
                $K \leftarrow K + 1$
                Initialize $m_K, n_K$, and $n_K^w$ as zero
```

D. Discussion

1) The High-dimensional Problem: In traditional similarity-based clustering methods, we need to represent documents with vectors whose length are the vocabulary size and define similarity between documents with some distance metric like cosine similarity. As these vectors are high-dimensional, the similarity between these documents will lose their effectiveness and statistical significance because of irrelevant attributes [2]. Therefore, clustering algorithms based on similarity measure between documents may no longer be effective in the high-dimensional space. Differently, GSDPMM evaluates the frequency of the words in a document appearing in each cluster as some kind of similarity between the document and the clusters. As GSDPMM only considers the words in the current document, it will not be affected by other irrelevant words in the vocabulary, and can deal with the high-dimensional problem of text clustering.

2) Outlier Detection: If a document is an outlier of the dataset, this document will have a really low probability of choosing an existing cluster. Because none of these existing clusters have documents that share many words with the outlier document, and Equation IV.29 will be small for any existing cluster. On the other hand, Equation IV.44 will have a relatively larger value for this outlier document, and this document will tend to choose a new cluster. In later iterations, other documents are less likely to choose this new cluster as they do not share many words with the outlier document, and this new cluster will tend to have only one document (the outlier) in the result. Therefore, the clusters with only one document in the result of GSDPMM can be regarded as outliers in the dataset. In Section V-D, we find that GSDPMM can achieve both really high recall and precision for the outlier detection task.

3) One Document Belongs to One Cluster: Although we assume that each document belongs to only one cluster, GSDPMM can still obtain the probability of each document belonging to each cluster with Equation IV.29. This means GSDPMM is a soft-clustering algorithm. Besides, GSDPMM has two advantages compared with algorithms (like LDA) which assume that each document is a distribution over topics as follows. First, in each iteration, GSDPMM only needs to sample a cluster for each document, while LDA needs to sample a topic for each word. This means GSDPMM has lower time complexity than LDA. Second, when sampling a topic for a word, LDA considers the popularity of the topic in the current document and the popularity of the current word in the topic. When we first see a document, we do not know the popularity of each topic in the document, and the first rule is useless. On the other hand, GSDPMM considers the popularity of each cluster in the whole dataset and the popularity of the document's words in the cluster. Even we first see a document, we already have the current popularity of each cluster in the dataset, and the two rules can both be useful. This illustrates why GSDPMM can converge faster than LDA.

4) Representation of Clusters: As we know, $\Phi_{z,w}$ corresponds to the probability of word w being generated by cluster z, and can be regarded as the importance of word w to cluster z. With the GSDPMM algorithm, we can estimate the number of clusters, K, and the cluster assignment z_d for each document d. For each topic $z = 1, ..., K$, we can derive the posterior of Φ_z as follows:

$$
\begin{aligned}
& p(\Phi_z|\vec{d}, \vec{z}, \alpha, \beta) \\
& = p(\Phi_z|\vec{d}_z, \beta) && \text{(IV.45)} \\
& = \frac{p(\Phi_z|\beta)p(\vec{d}_z|\Phi_z)}{\int p(\Phi_z|\beta)p(\vec{d}_z|\Phi_z)d\Phi_z} && \text{(IV.46)} \\
& = \frac{\frac{1}{\Delta(\beta)}\prod_{w=1}^{V}\Phi_{z,w}^{\beta-1}\prod_{w=1}^{V}\Phi_{k,w}^{n_z^w}}{\int\frac{1}{\Delta(\beta)}\prod_{w=1}^{V}\Phi_{z,w}^{\beta-1}\prod_{w=1}^{V}\Phi_{k,w}^{n_z^w}d\Phi_z} && \text{(IV.47)} \\
& = \frac{1}{\Delta(\vec{n}_z+\beta)}\prod_{w=1}^{V}\Phi_{z,w}^{n_z^w+\beta-1} && \text{(IV.48)} \\
& = Dir(\Phi_z|\vec{n}_z+\beta) && \text{(IV.49)}
\end{aligned}
$$

where $\vec{n}_z = \{n_z^w\}_{w=1}^{V}$, and n_z^w is the number of occurrences of word w in the zth cluster. Here, Equation IV.45 uses the properties of D-Separation, and Equation IV.46 exploits the Bayes Rule.

Using the expectation of the Dirichlet distribution, we can estimate $\Phi_{z,w}$ as follows:

$$\hat{\Phi}_{z,w} = \frac{n_z^w + \beta}{n_z + V\beta} \qquad \text{(IV.50)}$$

where n_z^w is the number of occurrences of word w in cluster z, and n_z is the total number of words in cluster z. It is interesting to notice that Equation IV.50 actually defines the fraction of occurrences of word w in cluster z, and is highly related to the second part of Equation IV.29. If word w has a relatively high value of $\hat{\Phi}_{z,w}$, it can be regarded as the representative word of cluster z. In Section V-G, we present the top ten representative words for each cluster that GSDPMM finds on a dataset, and find that these words can perfectly represent those clusters.

TABLE II: STATISTICS OF TEXT DATASETS (D:Number of Documents, K:Number of Clusters, V: Vocabulary Size, Avg Len: Average Length of the Documents)

Dataset	D	K	V	Avg Len
20NG	18,846	20	181,754	137.85
Tweet	2,472	89	5,098	8.56
TSet	11,109	152	8,111	6.23
SSet	11,109	152	18,478	22.20
TSSet	11,109	152	19,672	28.43

V. EXPERIMENTAL STUDY

A. Experimental Setup

1) Data Sets: We used three real text datasets in the experimental study, which are available on Github[1]:

- **20NG**[2] . This dataset consists of 18,846 documents from 20 major newsgroups. This is a classical dataset for the evaluation of text clustering methods. The average length of the documents in this dataset is 137.85.
- **TweetSet**. This dataset consists of 2,472 tweets that are highly relevant to 89 queries. The relevance between tweets and queries are manually labelled in the 2011 and 2012 microblog tracks at the Text REtrieval Conference [3] . The average length of the tweets in this dataset is 8.56.
- **Google News**. This dataset consists of the titles and snippets of 11,109 news articles about 152 events [3]. This dataset is further divided into three datasets: TitleSet(TSet), SnippetSet(SSet), and TitleSnippetSet(TSSet). The TSet and SSet only contain the titles and snippets, respectively, while the TSSet contains both the titles and snippets.

For all datasets, the preprocessing process includes converting all letters into lowercase, removing stop words, and stemming. After preprocessing, the statistics of these text datasets are shown in Table II. We can see the average length of the documents in 20NG is much larger than that of the TweetSet and Google News datasets. We plan to evaluate the performance of clustering methods on short and long texts.

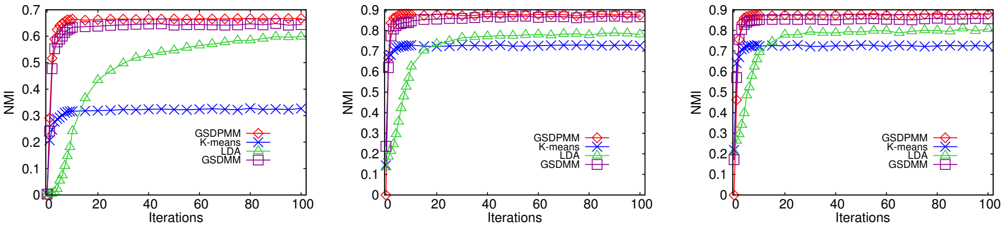
2) Evaluation Metrics: The Normalized Mutual Information (NMI) is widely used to evaluate the quality of the clustering results. NMI measures the amount of statistical information shared by the random variables representing the cluster assignments and the ground truth groups of the documents. In general, NMI is defined as follows [30]:

$$NMI = \frac{\sum_{h,l} n_{h,l}\log\left(\frac{n \cdot n_{h,l}}{n_h \cdot n_l}\right)}{\sqrt{(\sum_h n_h \log\frac{n_h}{n})(\sum_l n_l \log\frac{n_l}{n})}} \qquad \text{(V.1)}$$

[1] https://github.com/jackyin12/GSDMM/

[2] http://qwone.com/~jason/20Newsgroups/

[3] http://trec.nist.gov/data/microblog.html



(a) 20NG (b) TweetSet (c) TitleSet

Fig. 2: Comparison of the convergence speed of the clustering methods.

TABLE III: NMI Results of the Clustering Methods.

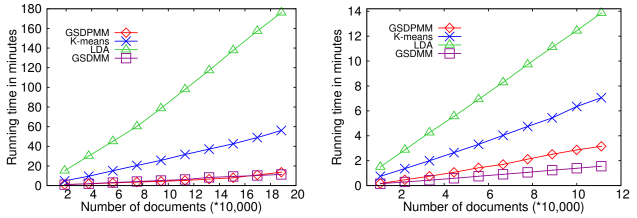
	K	GSDPMM	K-means	LDA	GSDMM
20NG	10	**.667** $\pm$ .004	.235 $\pm$ .008	.585 $\pm$ .013	.613 $\pm$ .007
	20	**.667** $\pm$ .004	.321 $\pm$ .006	.602 $\pm$ .012	.642 $\pm$ .004
	50	**.667** $\pm$ .004	.348 $\pm$ .006	.617 $\pm$ .013	.656 $\pm$ .002
Tweet	50	**.875** $\pm$ .005	.696 $\pm$ .008	.775 $\pm$ .012	.844 $\pm$ .006
	89	**.875** $\pm$ .005	.725 $\pm$ .007	.797 $\pm$ .011	.862 $\pm$ .008
	150	**.875** $\pm$ .005	.742 $\pm$ .006	.811 $\pm$ .012	.871 $\pm$ .004
TSet	100	**.873** $\pm$ .002	.687 $\pm$ .005	.769 $\pm$ .012	.830 $\pm$ .004
	152	**.873** $\pm$ .002	.721 $\pm$ .009	.784 $\pm$ .015	.852 $\pm$ .009
	200	**.873** $\pm$ .002	.730 $\pm$ .008	.806 $\pm$ .013	.868 $\pm$ .006
SSet	100	**.891** $\pm$ .004	.739 $\pm$ .006	.848 $\pm$ .005	.854 $\pm$ .004
	152	**.891** $\pm$ .004	.756 $\pm$ .006	.850 $\pm$ .006	.867 $\pm$ .008
	200	**.891** $\pm$ .004	.768 $\pm$ .007	.862 $\pm$ .004	.885 $\pm$ .005
TSSet	100	**.912** $\pm$ .003	.803 $\pm$ .009	.867 $\pm$ .004	.879 $\pm$ .009
	152	**.912** $\pm$ .003	.837 $\pm$ .004	.881 $\pm$ .003	.898 $\pm$ .004
	200	**.912** $\pm$ .003	.841 $\pm$ .005	.904 $\pm$ .005	.910 $\pm$ .003

where n_h is the number of documents in group h, n_l is the number of documents in cluster l, and $n_{h,l}$ is the number of documents in group h as well as in cluster l. When the clustering results perfectly match the ground truth groups, the NMI value will be one. While when the clustering results are randomly generated, the NMI value will be close to zero.

Homogeneity, Completeness, and V-Measure are used in [28]. Homogeneity represents the objective that each cluster contains only members of a ground truth group and completeness represents the objective that all members of a ground truth group are assigned to the same cluster. V-measure is an entropy-based measure which explicitly measures how successfully the criteria of homogeneity and completeness have been satisfied, and is actually equivalent to Normalized Mutual Information (NMI) [31]. In the experimental study, we use NMI, Homogeneity, and Completeness to evaluate the performance of the clustering methods.

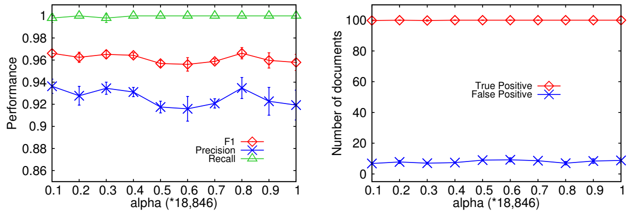
3) Methods for Comparison: In the experimental study, we compare GSDPMM with the following three clustering methods:

- **K-means**. K-means [11] is probably the most widely used method for clustering. Following [1], we set the similarity metric as cosine similarity. To cope with the problem of falling into local maximum, we set the number of initializations at 10 for each run of K-means.
- **LDA**. We treat the topics found by LDA [12] as clusters and assign each document to the cluster with the highest value in its topic proportion vector. Following [32], we set $\alpha = K/50$ and $\beta = 0.1$ where K is the number of topics assumed by LDA.
- **GSDMM**[1]. This is the state-of-the-art clustering method for short text clustering which is actually the collapsed Gibbs sampling algorithm for the Dirichlet Multinomial Mixture model. Following [3], we set $\alpha = 0.1$ and $\beta = 0.1$ for GSDMM.



(a) Long Text Dataset (b) Short Text Dataset

Fig. 3: Scalability



(a) Precision, Recall and F1 (b) True and False Outliers

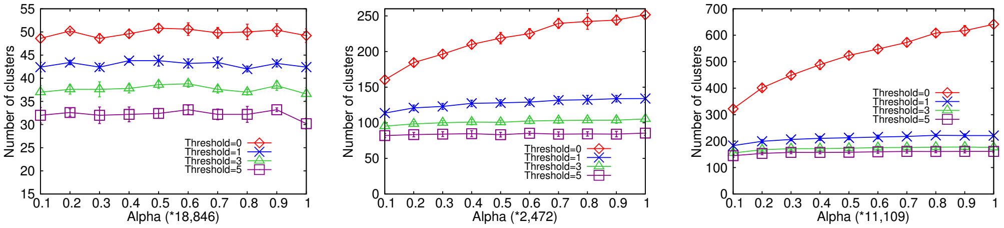
Fig. 4: Outlier detection with different values of α.

For GSDPMM, we set $K = 1$, $\beta = 0.02$, and $\alpha = 0.1 \times D$, where D is the number of documents in the dataset..

B. Comparison with Existing Methods

In this part, we try to investigate the convergence speed and performance of GSDPMM compared with K-means [11], LDA [12], and GSDMM [3]. Figure 2 shows the convergence speed on the clustering methods on 20NG, TweetSet, and TitleSet, and we have similar results on SnippetSet and TitleSnippetSet with the TitleSet. For GSDPMM, K-means, and GSDMM, we set K at the true number of clusters in each dataset. One observation is that the convergence speed of GSDPMM, K-means, and GSDMM are faster than the convergence speed of LDA. Another observation is that all clustering methods converge faster on the short text datasets (TweetSet and GoogleNews) than the long text dataset (20NG).

We report the mean and standard deviation of the NMI of the results by running each method for 20 independent trials on each dataset in Table III. For GSDPM, K-means, and GSDMM, we set the number of iterations at 10. While for LDA, we set the number of iterations at 100, because LDA is slow to converge as shown in Figure 2. We set the initial number of clusters at one for GSDPMM. As for other algorithms, we set three different initial number of clusters for each dataset. For each algorithm, we run 20 independent trials on each dataset, and report the mean and standard deviation of the NMI of the results in Table III. From Table III, we can see that

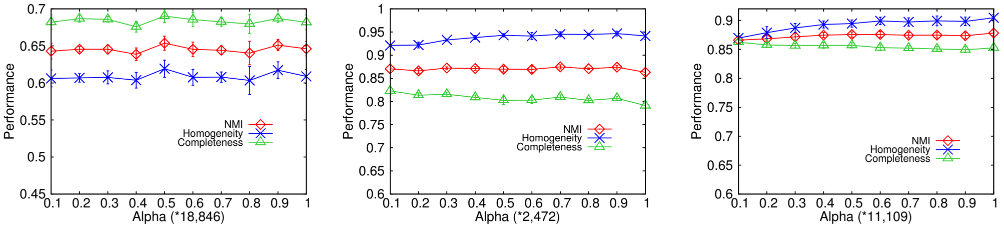


(a) 20NG

(b) TweetSet

(c) TitleSet

Fig. 5: Number of clusters found by GSDPMM with different values of α.



(a) 20NG

(b) TweetSet

(c) TitleSet

Fig. 6: Performance of GSDPMM with different values of α.

GSDPMM always achieves the highest performance compared with the other three methods on all datasets. Meanwhile, the standard deviations of the 20 independent trials of GSDPMM are quite small which means GSDPMM has high consistency. An interesting observation is that all methods perform better on short text datasets (TweetSet and GoogleNews) than the long text dataset (20NG). One possible reason is that TweetSet and GoogleNews are easier for clustering because they are about events and have smaller dictionary as shown in Table II. In addition, notice that these methods have better performance on TSSet and SSet than TSet. Another observation is that K-means, LDA, and GSDMM all achieve better performance when K is larger than the true number of clusters.

C. Scalability

In this part, we compare the scalability of GSDPMM with K-means [11], LDA [12], and GSDMM [3]. All algorithms were implemented in java and conducted on a machine running on 64bit Ubuntu Server 12.04 LTS version with an Interl Xeon E5310 1.60GHz processor and 19GB memory. We copied 20NG and TitleSet ten times respectively to construct two datasets called LongSet and ShortSet. We set K at 50 and 200 on the LongSet and ShortSet respectively for K-means, LDA, and GSDMM. We set the number of iterations at 10 for GSDPMM, K-means, and GSDMM. For LDA, we set the number of iterations at 100, because LDA needs about 100 iterations to get converged.

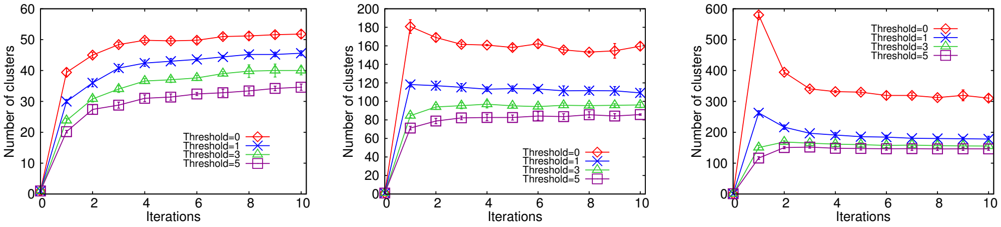
Figure 3 shows the running time of the four clustering methods on these two datasets with different number of documents, and we can see that the running time of these methods are all linear to the number of documents. An observation is that LDA is much slower than GSDPMM, and there are two reasons for this. First, LDA needs to sample a topic (cluster) for each word of the document, while GSDPMM only needs to sample a cluster for a document. Second, LDA needs more iterations to converge because LDA only considers the current word when choosing a topic for the word, while GSDPMM considers all words in the documents when choosing a topic for the document. Another observation is that GSDPMM has similar running time with GSDMM on LongSet, while it is slower than GSDMM on ShortSet. However, we should note that it is difficult to set a proper K for GSDMM as we do not know the true number, as a result, we may have to choose a really large K to ensure safety which will lead the complexity of GSDMM to be large.

D. Outlier Detection

In this part, we try to investigate the ability of GSDPMM for the outlier detection task. We manually generated 100 outlier documents for the 20NG dataset. Each document has 138 unique words that are not in the dictionary of 20NG. We mixed these outlier documents into 20NG, and obtained a dataset called Outlier20NG. We fix the number of iterations of GSDPMM at 5 and beta at 0.02. Then, we run GSDPMM on Outlier20NG with different values of α, and label the clusters with only one document as outliers. The standard deviations of the performance of outlier detection is reported by running GSDPMM for 20 independent trials. The results are shown in Figure 4.

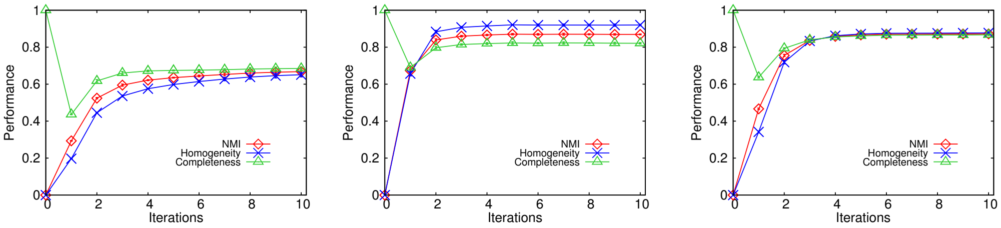
From Figure 4, we can see that GSDPMM can almost detect all the 100 outliers, and less than 10 documents in the original 20NG dataset are detected as outliers. Therefore, GSDPMM can achieve both really high recall and precision for the outlier detection task. After examining the "False Positive" outliers, we find that they are really different from the other documents in their ground truth group, which indicates that these documents are the potential true outliers in the original 20NG dataset.

Figure 5 shows the number of clusters found by GSDPMM with different values of α, where we set different thresholds for the size of the clusters. For example, when the threshold equals one, the clusters with only one document will not be counted. An interesting observation is that the number of clsuters found by GSDPMM grows with α on the TweetSet and TitleSet, while remains stable on 20NG. One possible explanation is



(a) 20NG (b) TweetSet (c) TitleSet

Fig. 7: Number of clusters found by GSDPMM with different number of iterations. We set different thresholds for the size of the clusters to discard the outliers detected by GSDPMM. For example, when the threshold is set to be three, the clusters with no more than three document will be discarded.



(a) 20NG (b) TweetSet (c) TitleSet

Fig. 8: Performance of GSDPMM with different number of iterations. We can see that the performance of GSDPMM grows quickly and gets stable within about five iterations.

that there are more potential outliers in these two short text datasets than 20NG. These potential outliers are more likely to be detected when we enlarge α, because α influence the probability of a document choosing a new cluster.

Figure 6 shows the performance of GSDPMM with different values of α. It is interesting to notice that NMI on the TweetSet and TitleSet remains stable, although the number of clusters found by GSDPMM grows with α on these two datasets, which indicates that GSDPMM is robust to outliers. Another interesting observation is that when we remove the clusters with only one document, the number of clusters found by GSDPMM is stable with different α in all the datasets.

E. *Infer the Number of Clusters*

In this part, we try to investigate the ability of GSDPMM to infer the number of clusters automatically. We set $\alpha = 0.1 \times D$ (D is the number of documents in the dataset), $K = 1$, and $\beta = 0.02$ for all the datasets.

Figure 7 shows the number of clusters found by GSDPMM with different number of iterations, and we set different thresholds for the size of the clusters to discard the outliers detected by GSDPMM. For example, when the threshold is set to be three, the clusters with no more than three document will be discarded. We only reported the results of 20NG, TweetSet, and TitleSet here, and we have similar results on the SnippetSet and TitleSnippetSet. From Figure 7, we can see the number of clusters found by GSDPMM on these datasets all get stable after several iterations. This means we can just assign all documents in a single cluster in the initialization, and the number of clusters will grow to a reasonable level after several iterations. In other words, GSDPMM can infer the number of clusters automatically.

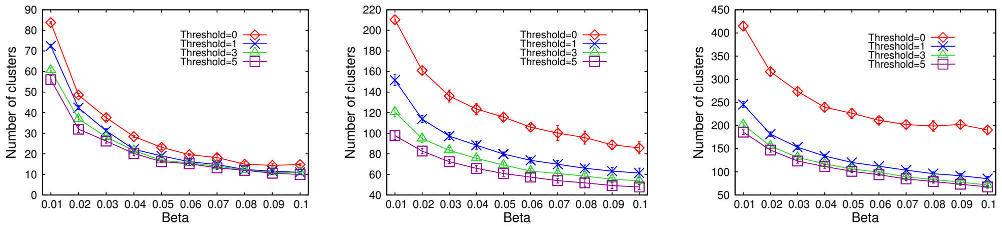
An observation is that the number of clusters found on TweetSet and TitleSet grows to a large number after one iteration, then drops and gets stable. While the number of clusters found on 20NG grows slightly and gets stable. One possible reason is that there are more ground truth clusters in the TweetSet and TitleSet, and many of them are small clusters. As a result, the documents of these two datasets are more likely to choose new clusters when we run GSDPMM, and the number of clusters grows fast in the first iteration. Then, these documents will tend to choose high quality clusters that contain similar documents with them, and the number of clusters will drop and finally gets stable.

Figure 8 shows the performance of GSDPMM with different number of iterations. One observation is that in the initialization, the homogeneity is 0 and the completeness is 1, as all the documents are assigned in the same cluster. Another observation is that the homogeneity grows and the completeness drops after one iteration. The reason is that many documents will choose new generated clusters that have similar documents with them, as a result, the purity of the clusters gets larger and the homogeneity grows. On the other hand, the ground truth groups are not in a single cluster any more, because the number of clusters gets larger, and the completeness drops. In addition, notice that the performance of GSDPMM grows quickly and gets stable within about five iterations, which illustrates that GSDPMM can converge fast.

F. *Balance Completeness and Homogeneity*

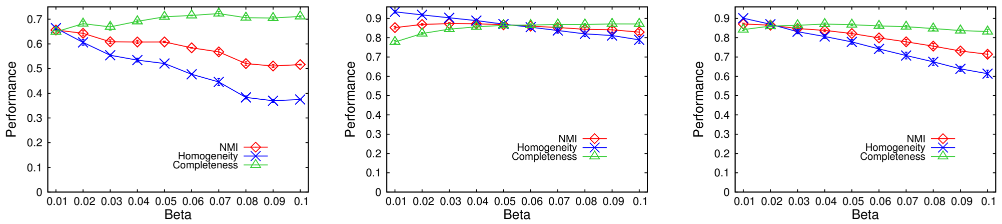
In this part, we try to investigate the influence of β to the results of GSDPMM. Here, we fix α at $0.1 \times D$ (D is the number of documents in the dataset), and the number of iterations at 5.

Figure 9 shows the number of clusters found by GSDPMM with different values of β on 20NG, TweetSet, and TitleSet, and we have similar results on SnippetSet and TitleSnippetSet with the TitleSet. We can see that the number of clusters found by GSDPMM drops when β gets larger. The reason is that β can balance the two rules of GSDPMM as discussed in Section IV-A. When β is larger, the probability of a document



(a) 20NG (b) TweetSet (c) TitleSet

Fig. 9: Number of clusters found by GSDPMM with different values of β.



(a) 20NG (b) TweetSet (c) TitleSet

Fig. 10: Performance of GSDPMM with different values of β.

choosing a cluster is less sensitive to the second part of Equation IV.29. As a result, Rule 1 (Choose a cluster with more documents) plays a more important rule than Rule 2 (Choose a cluster whose documents share more words with the current document), and GSDPMM will result in less number of clusters and achieve larger completeness.

Figure 10 shows the performance of GSDPMM with different values of β on 20NG, TweetSet, and TitleSet. An observation is that GSDPMM can achieve larger completeness with larger β and can achieve larger homogeneity with smaller β, which means GSDPMM can balance completeness and homogeneity with β.

G. Representation of Clusters

In this part, we try to investigate the ability of GSDPMM to obtain the representative words of each cluster. We run GSDPMM on 20NG, and set $\alpha = 0.1 \times D$ (D is the number of documents in the dataset). The number of iterations is set at 5 and $\beta = 0.02$. Table IV shows the top ten representative words for the 18 clusters found by GSDPMM. These clusters contain totally 16,266 documents which is about 86.3% of the documents in 20NG. We can see that these representative words can perfectly represent these clusters.

An interesting observation is that cluster "Hockey" and cluster "Baseball" have similar representative words like "game", "team", "year", and "player". The reason is that they are both about sports, and they share many words about sports. While GSDPMM can separate these two clusters really well, "Hockey" contains 893 documents in which 96.9% are from the ground truth "Hockey" group, and "Baseball" contains 906 documents in which 98.3% are from the ground truth "Baseball" group.

Another observation is that we have two clusters labeled as "Graphics", while their representative words are not similar. The reason is that ground truth "Graphics" group has two kinds of documents, one is more application-related, and the other is more research-related. GSDPMM separates these documents into two clusters.

VI. Conclusion

In this paper, we propose a collapsed Gibbs Sampling algorithm for the Dirichlet Process Multinomial Mixture model for text clustering (abbr. to GSDPMM) that can cope with the high-dimensional problem of text clustering, and has low time and space complexity. We also propose some novel and effective methods to detect the outliers in the dataset and obtain the representative words of each cluster. Our extensive experimental study shows that GSDPMM can achieve significantly better performance than three other clustering methods, and can achieve high consistency on both long and short text datasets. We found that GSDPMM can infer the number of clusters automatically and can scale well with huge text datasets.

We should note that GSDPMM has potentially well performance for incremental clustering. We can first group a number of documents into clusters with GSDPMM. Then each time a new document arrives, we can classify it into one of the existing clusters or a new cluster using Equation IV.29 and Equation IV.44, then we can update the corresponding statistics. In this way, new clusters can be easily detected and outdated documents can be easily removed. In future, we plan to study how to apply GSDPMM for incremental clustering.

Acknowledgment

We thank the anonymous reviewers for their valuable and constructive comments. This work was supported in part by National Basic Research Program of China (973 Program) under Grant No. 2014CB340505, National Natural Science Foundation of China under Grant No. 61532010 and 61272088, and Tsinghua University Initiative Scientific Research Program.

References

[1] C. C. Aggarwal and C. Zhai, "A survey of text clustering algorithms," in *Mining Text Data*. Springer, 2012, pp. 77–128.

TABLE IV: The Top Ten Representative Words for the 18 Clusters Found by GSDPMM on 20NG.

“Hardware”	“Electronics”	“Politics”	“Religion”	“Space”	“Windows”	“Crypt”	“Hockey ”	“Baseball”
card	noise	people	god	space	window	drive	game	game
drive	frequency	government	christian	nasa	file	state	team	year
problem	pink	state	people	year	problem	nntp	hockey	team
scsi	hz	time	jesus	henry	font	shaft	year	player
system	inch	fbi	bible	time	do	john	play	baseball
mac	max	well	church	launch	program	stafford	player	time
mb	khz	gun	time	system	help	winona	season	hit
window	price	fire	christ	gov	work	cpu	playoff	good
work	pc	good	life	shuttle	printer	distribution	nhl	win
bit	diagonal	thing	sin	orbit	system	au	goal	well
“Forsale”	“Atheism”	“Mideast ”	“Medical”	“Graphics”	“Windows.X”	“Motorcycle”	“Guns”	“Graphics”
sale	god	israel	doctor	image	window	bike	gun	computer
offer	people	israeli	patient	file	widget	dod	people	book
mail	atheist	arab	pitt	graphic	application	sun	firearm	system
state	religion	jew	time	window	color	dog	law	conference
nntp	science	people	people	system	problem	ve	crime	mail
computer	moral	muslim	gordon	server	mit	ride	weapon	paper
cd	thing	jewish	bank	ftp	display	time	control	navy
game	morality	state	problem	program	server	rider	handgun	science
shipping	belief	palestinian	good	sun	event	motorcycle	rate	graphic
drive	keith	war	disease	version	program	helmet	criminal	siggraph

[2] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, “When is “nearest neighbor” meaningful?” in *ICDT*. Springer, 1999, pp. 217–235.

[3] J. Yin and J. Wang, “A dirichlet multinomial mixture model-based approach for short text clustering,” in *SIGKDD*, 2014, pp. 233–242.

[4] A. McCallum, K. Nigam *et al.*, “A comparison of event models for naive bayes text classification,” in *AAAI-98 workshop on learning for text categorization*, vol. 752. Citeseer, 1998, pp. 41–48.

[5] G. Yu, R. Huang, and Z. Wang, “Document clustering via dirichlet process mixture model with feature selection,” in *SIGKDD*, 2010, pp. 763–772.

[6] J. Zhang, Z. Ghahramani, and Y. Yang, “A probabilistic model for online document clustering with application to novelty detection,” in *Advances in Neural Information Processing Systems*, 2004, pp. 1617–1624.

[7] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis (2nd ed.)*. CHAPMAN & HALL/CRC, 2003.

[8] R. M. Neal, “Markov chain sampling methods for dirichlet process mixture models,” *JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS*, vol. 9, no. 2, pp. 249–265, 2000.

[9] S. N. Maceachern, “Estimating normal means with a conjugate style dirichlet process prior,” *Communications in Statistics: Simulation and Computation*, vol. 23, no. 3, pp. 727–741, 1994.

[10] R. M. Neal, “Bayesian mixture modeling,” in *Maximum Entropy and Bayesian Methods*. Springer, 1992, pp. 197–211.

[11] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Applied statistics*, pp. 100–108, 1979.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *J. Mach. Learn. Res.*, 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944937

[13] H.-S. Park and C.-H. Jun, “A simple and fast algorithm for k-medoids clustering,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[14] P. Willett, “Recent trends in hierarchic document clustering: a critical review,” *Information Processing and Management*, vol. 24, no. 5, pp. 577–597, 1988.

[15] E. M. Voorhees, “Implementing agglomerative hierarchic clustering algorithms for use in document retrieval,” *Information Processing and Management*, vol. 22, no. 6, pp. 465–476, 1986.

[16] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “Optics: Ordering points to identify the clustering structure,” in *SIGMOD*, 1999, pp. 49–60.

[17] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise.” in *SIGKDD*, 1996, pp. 226–231.

[18] A. K. Jain, “Data clustering: 50 years beyond k-means,” *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[19] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.

[20] C. P. Robert and G. Casella, *Monte Carlo statistical methods*. Citeseer, 2004, vol. 319.

[21] D. Reynolds, “Gaussian mixture models,” *Encyclopedia of Biometrics*, pp. 659–663, 2009.

[22] K. Nigam, A. McCallum, S. Thrun, and T. M. Mitchell, “Text classification from labeled and unlabeled documents using em,” *Machine Learning*, vol. 39, no. 2/3, pp. 103–134, 2000.

[23] T. Hofmann, “Probabilistic latent semantic indexing,” in *SIGIR*, 1999, pp. 50–57.

[24] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, “Hierarchical dirichlet processes,” *Journal of the american statistical association*, vol. 101, no. 476, 2006.

[25] G. McLachlan and K. Basford, *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

[26] Y. W. Teh, “Dirichlet process,” in *Encyclopedia of machine learning*. Springer, 2010, pp. 280–287.

[27] G. Heinrich, “Parameter estimation for text analysis,” Technical Report, 2009.

[28] A. Rosenberg and J. Hirschberg, “V-measure: A conditional entropy-based external cluster evaluation measure,” in *EMNLP-CoNLL*, 2007, pp. 410–420.

[29] G. Salton, A. Wong, and C. S. Yang, “A vector space model for automatic indexing,” *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[30] A. Strehl and J. Ghosh, “Cluster ensembles—a knowledge reuse framework for combining multiple partitions,” *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[31] H. Becker, “Identification and characterization of events in social media,” Ph.D. dissertation, COLUMBIA UNIVERSITY, 2011.

[32] T. L. Griffiths and M. Steyvers, “Finding scientific topics,” *Proceedings of the National academy of Sciences of the United States of America*, vol. 101, no. Suppl 1, pp. 5228–5235, 2004.