APPROXIMATE BAYESIAN INFERENCE FOR COUNT DATA MODELING

FRANCISCO XAVIER SUMBA TORAL

A THESIS

IN

THE DEPARTMENT

OF

ELECTRICAL AND COMPUTER ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF APPLIED SCIENCE (ELECTRICAL AND COMPUTER ENGINEERING)

CONCORDIA UNIVERSITY

MONTRÉAL, QUÉBEC, CANADA

APRIL 2020

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Francisco Xavier Sumba Toral**

Entitled: **Approximate Bayesian Inference for Count Data Modeling**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

Dr. O. Ait Mohamed _____ Chair

Dr. M. Nik-Bakht (BCEE) _____ External Examiner

Dr. H. Rivaz _____ Internal Examiner

Dr. Nizar Bouguila (CIISE) _____ Supervisor

Approved _____

Dr. Y.R. Shayan, Chair.

Department of Electrical and Computer Engineering

_____ 20 _____ _____

Dr. Amir Asif, Dean

Faculty of Engineering and Computer Science

Abstract

Approximate Bayesian Inference for Count Data Modeling

Francisco Xavier Sumba Toral

Bayesian inference allows to make conclusions based on some antecedents that depend on prior knowledge. It additionally allows to quantify uncertainty, which is important in Machine Learning in order to make better predictions and model interpretability. However, in real applications, we often deal with complicated models for which is unfeasible to perform full Bayesian inference. This thesis explores the use of approximate Bayesian inference for count data modeling using Expectation Propagation and Stochastic Expectation Propagation.

In Chapter 2, we develop an expectation propagation approach to learn an EDCM finite mixture model. The EDCM distribution is an exponential approximation to the widely used Dirichlet Compound distribution and has shown to offer excellent modeling capabilities in the case of sparse count data. Chapter 3 develops an efficient generative mixture model of EMSD distributions. We use Stochastic Expectation Propagation, which reduces memory consumption, important characteristic when making inference in large datasets.

Finally, Chapter 4 develops a probabilistic topic model using the generalized Dirichlet distribution (LGDA) in order to capture topic correlation while maintaining conjugacy. We make use of Expectation Propagation to approximate the posterior, resulting in a model that achieves more accurate inference compared to variational inference. We show that latent topics can be used as a proxy for improving supervised tasks.

Acknowledgement

I wish to thank Nizar Bouguila, my supervisor, not only for introducing me in using probability theory to do Machine Learning but also for giving me the independence to do my research. Always I felt stuck our discussions provide me an idea that helped me progress. This freedom greatly allowed me to satisfy my curiosity and feed my ravenous for knowledge.

It has been my pleasure to exchange ideas with everyone I met during this journey, endless thanks to all of you. I, first, have to thank Nuha Zamzami who not only encouraged me when nothing worked but also collaborated with me in Chapters 2 and 3, without her this thesis it would not be possible. I would like to thank all the members of the lab who are always supportive and welcoming. I want to thank Ziyang Song for those vibrant discussions, sharing ideas, and helping me form our own reading club! Thanks to all my coworkers at HeyDay.ai, and more specifically to Jordan Prince for being an incredible mentor and proving me interesting discussions. Thanks everybody for your support!

I learn a lot from different faculty members, but I extremely want to thank professor Yue Li from whom I learned a ton and introduced me to new concepts and ways of doing Machine Learning, thanks for all those interesting conversations. I want also to thank Will Hamilton for being an excellent professor. I had plenty of learning, and I really enjoyed his courses, it clarified many concepts that were useful during this work.

I would like to thank all the institutions that helped me do my research and pursue graduate school. This work was generously supported thanks to a Concordia Merit scholarship, a graduate scholarship from the faculty of Engineering and Computer Science, a research funding assistantship, Mitacs, and Heyday.ai.

Finally, I would like to thank my mother and my grandmother for everything. All the sacrifices they have done motivates me and you are the reason of all of this. Thanks!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays, there is an overwhelming amount of data that continues to increase more and more. These data vary in content (e.g. tweets, news, security recordings, etc) and kind (e.g. documents, images, speech, etc). Moreover, the emergence of the internet has led us to have an interconnected world that facilitates data sharing and generation of new content; thus, countless streams of data are generated daily. A large portion of these data comes as discrete data (count data) such as documents, messages from social media, or features extracted from videos or images.

Extracting knowledge from large datasets of count data allows us to make inferences from a specific problem at hand. Machine learning helps us uncover patterns, but in most cases it is hard or expensive to label these large amounts of count data for a supervised setting. Unsupervised learning, clustering, however, allows to uncover patterns with no need of labels, more specifically when given a group of count data, mixture models allow to incorporate some hidden knowledge and make inferences. In natural language processing, when dealing with text, we can infer statistical regularities as a result of these hidden components that often correspond to groups of data or topics. These learned models or distributions can be later used as a proxy for other machine learning tasks such as classification or semi-supervised settings. Additionally, we want to have interpretable

results with a certain amount of uncertainty. The Bayes framework allows measuring uncertainty under a probabilistic model.

1.1 Approximate Inference

In Bayesian inference, we make use of the Bayes' theorem (Eq. 1) in order to infer a posterior distribution that is a consequence of the likelihood function and some prior knowledge. It not only allows to quantify uncertainty but as more information is available, our initial hypothesis can be updated. In this setting, we often want to make inferences about unknown data or parameters $\mathbf{\Theta}$ given the observed data $\mathcal{X}$, which require the computation of the evidence. Computing the evidence can be unfeasible due to complicated integrals since we need to marginalize the latent variables from the likelihood (i.e. $\int p(\mathbf{\Theta}, \mathcal{X}) d\mathbf{\Theta}$). Thus, instead of calculating the exact posterior, we estimate it.

$$p(\mathbf{\Theta} \mid \mathcal{X}) = \frac{p(\mathbf{\Theta}, \mathcal{X})}{p(\mathcal{X})} = \frac{p(\mathbf{\Theta}, \mathcal{X})}{\int p(\mathbf{\Theta}, \mathcal{X}) d\mathbf{\Theta}} \tag{1}$$

There are many advances that have been done in approximate inference, but approximate methods can be classified in *deterministic* and *sampling* methods. The former evaluates the integral in several locations and constructs an approximate function. The latter relies in the law of large numbers and given enough samples, the integral will converge to the true value.

The rest of this chapter describes some previous work on approximate Bayesian inference that lies groundwork for the remaining chapters.

1.2 Monte Carlo

Monte Carlo (MC) [4,33] can be motivated by the law of large numbers, if we have enough samples from a distribution, its average converges to the expected value. It is a flexible way of approximating sums or integrals when they cannot be computed in closed-form.

The idea is to see any expectation as a sum or integral and then approximate it by the average. However, it is computationally expensive. The estimator is as shown in Eq. 2, where we take N samples, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, from the distribution p and evaluate in a function $f(x_i)$.

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{x}_i) \approx \mathbb{E}_p \left[f(\boldsymbol{x}) \right] \tag{2}$$

Sometimes, when it is not possible to sample from the distribution p, Importance Sampling (IS) or Markov Chain Monte Carlo (MCMC) can be used.

1.2.1 Importance Sampling

In cases when is impractical to sample from the target distribution $p(\boldsymbol{x})$, we can propose a decomposition using a proposal distribution $q(\boldsymbol{x})$ that matches the shape of the distribution and is easier to sample (i.e. $p(\boldsymbol{x})f(\boldsymbol{x}) = q(\boldsymbol{x})p(\boldsymbol{x})f(\boldsymbol{x})/q(\boldsymbol{x})$). We can select any proposal distribution, however, this choice is sensitive to variance, and the optimal choice of the proposal distribution is important. Then, we can compute the expectation by averaging N samples from the proposal distribution (Eq. 3).

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} \frac{p(\boldsymbol{x}_i) f(\boldsymbol{x}_i)}{q(\boldsymbol{x}_i)} \approx \mathbb{E}_q \left[\frac{p(\boldsymbol{x}) f(\boldsymbol{x})}{q(\boldsymbol{x})} \right] \tag{3}$$

1.3 Laplace Approximation

Laplace's method [43] seeks a Gaussian approximation $q(\boldsymbol{x})$ to a probability density function $p(\boldsymbol{x}) = f(\boldsymbol{x})/Z$, where Z is a normalizing constant. The approximate distribution is centered in the mode of $p^{'}(\boldsymbol{x}_0) = 0$. We expand $\log f(\boldsymbol{x})$ about its mode using Taylor expansion:

$$\log f(\boldsymbol{x}) \approx \log f(\boldsymbol{x}_0) - \frac{1}{2} H (\boldsymbol{x} - \boldsymbol{x}_0)^2 \tag{4}$$

where H is the Hessian matrix $H = -\nabla^2 \log f(\boldsymbol{x}) \mid_{\boldsymbol{x}=\boldsymbol{x}_0}$.

By taking the exponent and normalizing, we obtain an approximate Gaussian distribution to $p(\boldsymbol{x})$ that is centered on its mode (Eq. 5).

$$q(\boldsymbol{x}) = \frac{\sqrt{H}}{\sqrt[D]{2\pi}} \exp\{-\frac{H}{2}(\boldsymbol{x} - \boldsymbol{x}_0)^2\} \tag{5}$$

where D is the dimension of $\boldsymbol{x}$.

1.4 Expectation Propagation

Expectation Propagation (EP) [60] is a generalization of Assumned Density Filtering (ADF) [64], which is a one-pass sequential method and is dependent on the order of data points. Unlike ADF, EP reuses data points to perform iterative refinements. In other words, EP handles partitioned data and combines partitions iteratively through message passing. Indeed, EP is more computationally efficient than MCMC [63], and it has shown to be more accurate than Variational Inference (VI) [60, 61].

Having the latent variable $\boldsymbol{\Theta}$, EP approximates a target distribution $p(\boldsymbol{\Theta} \mid \mathcal{X})$, which is commonly the posterior, with a global approximation $q(\boldsymbol{\Theta})$ that belongs to the exponential family. The choice of q depends on the problem but it has to be a simple approximating distribution that can be fitted using small refinements. Thus, in order to apply EP, firstly the target distribution must be factorizable such that the posterior can be split in D sites $p(\boldsymbol{\Theta} \mid \mathcal{X}) \propto p_0(\boldsymbol{\Theta}) \prod_i^D p_i(\mathbf{x}_i \mid \boldsymbol{\Theta})$; the initial site p_0 is commonly represented with the prior distribution and the remaining p_i sites represent the contribution of each term to the likelihood. The approximating distribution must admit a similar factorization, *i.e.* $q(\boldsymbol{\Theta}) \propto \prod_i^D \tilde{p}_i(\boldsymbol{\Theta})$. Therefore, the goal of EP is to refine each of the approximating sites such that they capture the contribution of each of the likelihood sites to the posterior, *i.e.* $\tilde{p}_i(\boldsymbol{\Theta}) \approx p_i(\mathbf{x} \mid \boldsymbol{\Theta})$. Each approximating site has to be initialized and belong to the exponential family. Consequently, each site is refined to create a cavity distribution by

dividing the global approximation over the current approximate site.

$$q^{\setminus i}(\mathbf{\Theta}) \propto \frac{q(\mathbf{\Theta})}{\tilde{p}_i(\mathbf{\Theta})} \tag{6}$$

Additionally, in order to approximate each site, we introduce a new tilted distribution which consists in the product of the cavity distribution and the current site.

$$q_i^*(\mathbf{\Theta}) \propto p_i(\mathbf{\Theta}) q^{\setminus i}(\mathbf{\Theta}) \tag{7}$$

Subsequently, a new posterior is found by minimizing the Kullback Leibler divergence $D_{KL}(q_i^*(\mathbf{\Theta}) \mid\mid q^{new}(\mathbf{\Theta}))$ such that $\tilde{p}_i(\mathbf{\Theta}) \approx p_i(\mathbf{x} \mid \mathbf{\Theta})$. This minimization is equivalent to match the moments of those distributions [4, 61]. We can also notice that this updating scheme creates a coupling for the approximating factors, so updates must be iterated. Finally, the revised approximate site is updated by removing the remaining terms from the current approximation $\tilde{p}_i(\mathbf{\Theta}) \propto q^{new}(\mathbf{\Theta}) / q^{\setminus i}(\mathbf{\Theta})$.

EP can also be seen as a variational method [8, 83] that instead of evaluating the KL divergence from p to q, it evaluates from q to p.

1.5 Stochastic Expectation Propagation

Efficient inference and learning for probabilistic models that scale to large datasets are essential in the Bayesian setting. Thus, a variety of methods have been proposed from sampling approximations [58] to distributional approximations such as Stochastic Expectation Propagation (SEP) [37].

As previously mentioned EP commonly provides more accurate approximations compared to sampling methods [63] and variational inference [60, 61]. Yet, the number of parameters grows with the number of data points, causing memory overheads and making it difficult to scale to large datasets. Besides, ADF [64], which has been introduced

before EP, maintains a global approximating posterior; however, it results in poor estimates. Therefore, [48] proposed an alternative to push EP to large datasets denominated Stochastic Expectation Propagation (SEP). SEP takes the best of these two methods by maintaining a global approximation that is updated locally. It does this by introducing a global site that captures the average effect of the likelihood sites and, as a result avoiding memory overheads.

For the same Bayesian setting where we are given a probabilistic model $p(\mathcal{X} \mid \boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$ drawn from a prior $p_0(\boldsymbol{\theta})$, SEP approximates a target distribution $p(\boldsymbol{\theta} \mid \mathcal{X})$, which is commonly the posterior, with a global approximation $q(\boldsymbol{\theta})$ that belongs to the exponential family. The target distribution must be factorizable such that the posterior can be split in D sites $p(\boldsymbol{\theta} \mid \mathcal{X}) \propto p_0(\boldsymbol{\theta}) \prod_{i=1}^{D} p_i(\boldsymbol{\theta})$; the initial site p_0 is commonly represented with the prior distribution and the remaining p_i sites represent the contribution of each ith item to the likelihood. The approximating distribution must admit a similar factorization as:

$$q(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \tilde{p}(\boldsymbol{\theta})^D \tag{8}$$

Unlike EP, the SEP maintains a global approximating site, $\tilde{p}(\boldsymbol{\theta})^D$, to capture the average effect of a likelihood on the posterior. Thus, we only have to maintain the parameters of the approximate posterior and approximate global site that commonly belongs to the exponential family. Consequently, each site is refined to create a cavity distribution (Eq. 9) by dividing the global approximation over one of the copies of the approximate site.

$$q^{\setminus 1}(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta}) / \tilde{p}(\boldsymbol{\theta}) \tag{9}$$

Additionally, in order to approximate each site, a new tilted distribution (Eq. 10) is introduced using the cavity distribution and the current site.

$$\hat{p}_i(\boldsymbol{\theta}) \propto p_i(\boldsymbol{\theta}) q^{\setminus 1}(\boldsymbol{\theta}) \tag{10}$$

Subsequently, a new posterior is found by minimizing the Kullback Leibler divergence $D_{KL}(\hat{p}_i(\boldsymbol{\theta}) \mid\mid q^{new}(\boldsymbol{\theta}))$ such that $\tilde{p}_i(\boldsymbol{\theta}) \approx p_i(\boldsymbol{\theta})$. This minimization is equivalent to match the moments of those distributions [4,61]. Finally, the revised approximate site is updated by removing the remaining terms from the current approximation by employing damping [31,59] in order to make a partial update since $\tilde{p}_i$ captures the effect of a single likelihood function:

$$\tilde{p}(\boldsymbol{\theta}) = \tilde{p}(\boldsymbol{\theta})^{1-\eta} \left(\frac{q^{new}(\boldsymbol{\theta})}{q^{\setminus w}(\boldsymbol{\theta})} \right)^{\eta} = \tilde{p}(\boldsymbol{\theta})^{1-\eta} \tilde{p}_i(\boldsymbol{\theta})^{\eta} \tag{11}$$

Notice that η is the step size, and when $\eta = 1$, no damping is applied. A natural choice is $\eta = 1/D$.

1.6 Contributions

The key contributions of this thesis were either published or being reviewed in scientific journals or conferences. The contributions are as follows:

1. Creating an EDCM mixture model for count data using EP for inference [80]. We also propose an initialization method for the mixture model which facilitates learning.

2. An improvement of the EDCM mixture model with a distribution with more degrees of freedom named EMSD that captures better count data and models word appearance. We employ SEP for inference that is more appropriate for large datasets [79].

3. We learn the topic model LGDA that replaces the Dirichlet distribution with Generalized Dirichlet distribution modeling topic correlation and show that the learned topics can be used for supervised tasks [77].

1.7 Thesis Structure

The next chapters present three new clustering models for count data that achieve comparable results to its analogous counterparts. In general, we make use of EP, SEP, and other deterministic or sampling methods to compute intractable integrals. First, in Chapter 2, we introduce a mixture model that models the burstiness problem using the EDCM distribution using EP for inference. Chapter 3 extends the mixture model by making use of SEP to learn an EMSD mixture that has more degrees of freedom and captures better word occurrence. Later, in Chapter 4, we introduce a topic model that captures the correlation between topics while maintaining conjugacy. Finally, in Chapter 5 we conclude and point out future directions for this thesis.

Chapter 2

Improving the EDCM mixture model with Expectation Propagation

Bayesian inference is extremely important to challenging scenarios that involve complex probabilistic models, which are usually intractable. In this work, we develop an Expectation Propagation approach to learn EDCM finite mixture models. The EDCM distribution is an exponential approximation to the widely used Dirichlet Compound distribution and has been shown to offer excellent modeling capabilities in the case of sparse count data. Expectation Propagation is a deterministic approach that provides accurate approximations to the full posterior and allows to include prior beliefs in the model as opposed to the maximum-likelihood method which provides point estimates only. We evaluate the validity of our framework on several datasets for sentiment analysis and image recognition. Our proposed model shows comparable to superior results to other approaches in the literature.

2.1 Introduction

Statistical methods are excellent at modeling semantic content of text documents [46]. More specifically, document clustering is widely used in a variety of applications such as

text retrieval or topic modeling [5]. For instance, Latent Dirichlet Allocation (LDA) [12], a very well-know hierarchical topic model, captures the word-topic assignment. In other words, LDA can capture the likeliness of word w appearing in topic k. However, in other settings, it is necessary to know the word appearance dependencies, *i.e.* if word w appears once, it is more probable that the same word w will appear again. This phenomenon is denominated as burstiness, which has shown to be addressed using Dirichlet Compound Multinomial (DCM) distribution [53]. Furthermore, taking into account the sparsity and high-dimensionality of text data, [26] proposed the EDCM distribution which approximates the DCM as a member of the exponential family. EDCM has shown to be more efficient and keep the merits of DCM for modeling word occurrence dependency. Indeed, EDCM distribution has been successfully used to develop a mixture model to efficiently cluster high-dimensional count data in several real-world applications (*e.g.* [26, 41, 62, 87, 91]).

At the core of our proposed method, there is the notion of modeling the behavior of rare words appearing often in a document. The DCM distribution not only captures this behavior [53] but also models text data better than a multinomial distribution. Similarly, different distributions had been used in order to model burstiness while preserving conjugacy; for instance, [88] used the Scaled Dirichlet instead of Dirichlet distribution and other works used Generalized Dirichlet [13] or Beta-Liouville distribution [15]. However, all these models share similar limitations including that they do not belong to the exponential family of distributions and their parameters estimation is slow especially in high-dimensional spaces. The approximation for the DCM distribution, denominated as EDCM, offers fast parameter learning and a helpful intuition for the study of the burstiness phenomenon [26]. Moreover, Bayesian learning commonly involves statistical modeling and inference methods. Parameter learning is one of the encountered challenges in mixture models, and typically the maximum-likelihood method via the Expectation Maximization (EM) algorithm has been used for learning the parameters of an EDCM mixture model [26].

In spite of the maximum-likelihood method has been showing fast parameter learning, this approach suffers from numerous inconveniences such as providing a point estimate, which impacts the accuracy of the learned model [4]. Additionally, the appropriate number of components has to be known in advance, which can be approached by selecting the appropriate model with techniques such as Minimum Message Length criterion (MML) [2,17]. For instance, recent work has developed an MML criterion based on EDCM [87,91] to detect the appropriate number of clusters, but also the authors claim its improvement is due to the prior information introduced by the MML-based criterion. In fact, deterministic Bayesian inference techniques (*e.g.* variational inference or expectation propagation) allow good approximation of the full posterior. Recently, [62] has proposed the use of a sampling method, *i.e.* Markov Chain Monte Carlo (MCMC), for learning an EDCM mixture and has shown the importance of having priors, outperforming previous results. However, sampling methods are computationally expensive [68].

In this work, we study the application of the Bayesian framework for learning the EDCM mixture model. In particular, we propose an approach for an EDCM mixture model using Expectation Propagation (EP) [60] for parameter learning. EP is a generalization of Assumed Density Filtering (ADF) that approximates the model posterior with a tilted distribution using small refinements to approximate the global posterior. EP, a deterministic approximate inference framework, has shown to be more accurate than methods such as variational inference and MCMC [4, 59], and it has shown appropriate generalization in a Gaussian mixture model [61], hierarchical models such as LDA [59] or even infinite mixtures [28]. The contributions of this chapter are summarized as follows: 1) derive foundations to learn an EDCM mixture model using EP; 2) test and evaluate the proposed approach on high-dimensional count data.

The rest of this chapter is organized as follows. First, Section 2.2 revisits the core methods upon our work is built on, such that, we review the family of distributions (*i.e.* DCM and EDCM distributions) to tackle the burstiness problem. Next, in Section 2.3, we outline the EDCM mixture model, describe the expectation propagation approach, and

derive a complete learning approach. Section 2.4 describes our experimental setup and evaluation of our proposed method. Finally, we conclude the chapter in Section 2.5.

2.2 The Exponential-family Approximation to DCM Distribution

We start with a brief review of the approximation of the Dirichlet Compound Multinomial distribution (EDCM) [26]. We are given a dataset $\mathcal{X}$ with D samples $\mathcal{X} = \{\mathbf{x_i}\}_{i=1}^{D}$, each $\mathbf{x}_i$ is a vector of count data (*e.g.* a document or an image, represented as a vector of word frequencies or visual words, respectively).

2.2.1 Dirichlet Compound Multinomial distribution

A text document of vocabulary size V is commonly modeled with a multinomial distribution with parameters $\boldsymbol{\theta}$:

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \frac{n!}{\prod_{w=1}^{V} x_w!} \prod_{w=1}^{V} \theta_w^{x_w} \tag{12}$$

where $n = \sum_{w=1}^{V} x_w$ is the document length.

However, the multinomial distribution is not appropriate when analyzing the burstiness of words. This is due to the fact that according to the multinomial distribution words follow the i.i.d assumption, but in real data, there is actually an occurrence dependency such that if a word appears once, it is more likely to appear again [44]. In [53], the authors proposed a generative model to deal with this problem by introducing a prior Dirichlet distribution with parameters $\boldsymbol{\alpha}$. They define a new marginal distribution by integrating out $\boldsymbol{\theta}$, obtaining a discrete distribution known as the Dirichlet Compound Multinomial (DCM) distribution or multivariate Polya distribution.

$$\mathcal{DCM}(\mathbf{x} \mid \boldsymbol{\alpha}) = \frac{n!}{\prod_{w=1}^{V} x_w!} \frac{\Gamma(s)}{\Gamma(s+n)} \prod_{w=1}^{V} \frac{\Gamma(x_w + \alpha_w)}{\Gamma(\alpha_w)} \tag{13}$$

where $s = \sum_{w=1}^{V} \alpha_w$ is the sum of the Dirichlet distribution parameters. This model has an intuitive interpretation representing the Dirichlet as a general topic and the multinomial as a document-specific subtopic, making some words more likely in document $\mathbf{x}$ based on word counts.

2.2.2 Approximating the DCM distribution

Text documents representation is very sparse because not every word appears in most of the documents. In [26], the authors noted that using only the non-zero values of $\mathbf{x}$ is computationally efficient. Moreover, the parameter α_w of the DCM distribution is small for most words, $\alpha_w \ll 1$. Thus, replacing $\frac{\Gamma(x_w+\alpha_w)}{\Gamma(\alpha_w)}$ by $\Gamma(x_w)\alpha_w$ and using the fact that $\Gamma(x_w) = (x_w - 1)!$ leads to an approximation of the DCM distribution known as EDCM. We replace α with β in order to follow the same notation as in [26]:

$$\mathcal{EDCM}(\mathbf{x} \mid \beta) = n! \frac{\Gamma(s)}{\Gamma(s+n)} \prod_{w:x_w \geq 1} \frac{\beta_w}{x_w} \tag{14}$$

Additionally, Eq.(14) exhibits a nice interpretation for why the DCM or EDCM distributions are appropriate for the burstiness problem. It is noticeable that the resulting probability of a document depends on the words appearing in it since it is proportional to $\prod_{w:x_w \geq 1} \beta_w / x_w$. In other words, if a word w appears once, it reduces the document's probability by β_w, taking into account both word type and word token. Thus, the mth appearance of word w reduces the document's probability by $(m-1)/m$, and as a result, multiple appearances of the same word leads to a high probability.

2.3 The proposed model

2.3.1 Mixture-based Clustering Model

In this section, we state the settings for a finite EDCM mixture model and develop a framework for learning the mixture using expectation propagation.

Here, we state the settings for a finite EDCM mixture model and develop a mathematical framework for learning the mixture using expectation propagation. Generally, a finite mixture model is represented as the graphical model shown in Figure 1. We assume that we are given D documents drawn from an $\mathcal{EDCM}$ distribution, and each $\mathbf{x}_i$ document is composed of V words. $K \geq 1$ represents the number of mixture components or clusters. Thus, a document is drawn from its respective component j as follows: $\mathbf{x_i} \sim \mathcal{EDCM}(\boldsymbol{\beta}_j)$.

Consequently, a latent variable $\mathcal{Z} = \{\mathbf{z}_i\}_{i=1}^{D}$ is introduced for each $\mathbf{x}_i$ document in order to represent the component assignment. We posit a Multinomial distribution for the component assignment such that $\mathbf{z}_i \sim Mult(1, \boldsymbol{\pi})$ where $\boldsymbol{\pi} = \{\pi_j\}_{j=1}^{K}$ represents the mixing weights, and they are subject to the constraints $0 < \pi_j < 1$ and $\sum_j \pi_j = 1$. In other words, $\mathbf{z}_i$ is a K-dimensional indicator vector containing a value of one when document $\mathbf{x}_i$ belongs to the component j, and zero, otherwise. Note that in this setting the value of $z_{ij} = 1$ acts as the selector of the component that generates $\mathbf{x}_i$ document with parameter $\boldsymbol{\beta}_j$; hence, $p(\mathbf{z}_i \mid \boldsymbol{\pi}) = \pi_j$.

Therefore, following the graphical model in Figure 1, the full posterior can be written as follows:

$$p(\boldsymbol{\pi}, \boldsymbol{\beta} \mid \mathcal{X}) \propto p(\boldsymbol{\pi})p(\boldsymbol{\beta}) \prod_{i}^{D} \sum_{\mathbf{z}_i} p(\mathbf{x}_i \mid \mathbf{z}_i, \boldsymbol{\beta})p(\mathbf{z}_i \mid \boldsymbol{\pi}) \tag{15}$$

$$\propto p(\boldsymbol{\pi})p(\boldsymbol{\beta}) \prod_{i}^{D} p(\mathbf{x_i} \mid \boldsymbol{\beta}, \boldsymbol{\pi})$$

$$\propto p(\boldsymbol{\pi})p(\boldsymbol{\beta}) \prod_{i}^{D} \sum_{j}^{K} \pi_j p(\mathbf{x}_i \mid \boldsymbol{\beta}_j) \tag{16}$$

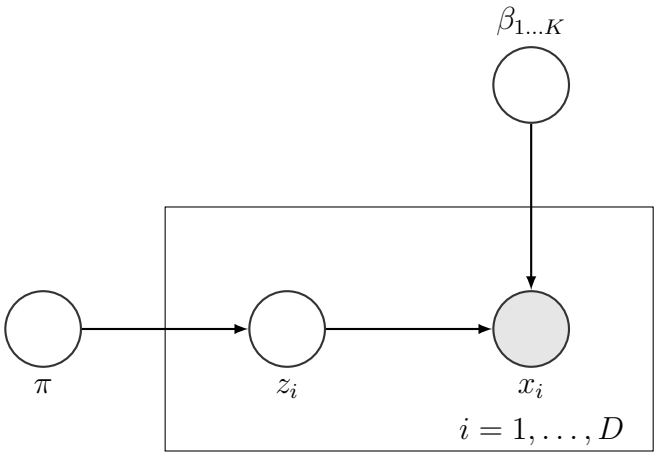

Figure 1: Graphical model representation of the EDCM mixture model. The box is a plate representing documents, white circles represent latent variables and shaded circle represents the observed variables. Arrows represent the conditional dependence between random variables.

2.3.2 Parameters Learning

In this section, we describe the learning approach using EP algorithm. We partition the likelihood in D sites and start by defining an ith approximating site for each of the latent variables ($\boldsymbol{\pi}$ and $\boldsymbol{\beta}$). First, we assign a Dirichlet distribution with parameter $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ as a prior for the mixing weights since it fits properly the constraints imposed by the model and works as a nice prior for the mixing weights $\boldsymbol{\pi}$ that holds conjugacy properties.

$$\tilde{p}_i(\boldsymbol{\pi} \mid \boldsymbol{\alpha}_i) \propto \prod_{j=1}^{K} \pi_j^{\alpha_{ij}-1} \tag{17}$$

For the $\boldsymbol{\beta}$ variable of the EDCM mixture, we adopt a Gaussian distribution, which leads to an intractable distribution since $\tilde{p}(\boldsymbol{\pi})$ is a Dirichlet distribution. However, this setting has been used successfully to approximate Beta and Dirichlet distributions [27,51]. Additionally, a Gaussian distribution not only allows analytically tractable calculations but also captures correlation for the values of $\boldsymbol{\beta}_j$. Hence, we select for the approximating site of $\boldsymbol{\beta}_j$ a Gaussian distribution with mean $\mathbf{m}_{ij}$ and precision matrix $\boldsymbol{\Lambda}_{ij}^{-1}$ for each j

component.

$$\tilde{p}_i(\boldsymbol{\beta}) = \prod_{j}^{K} \mathcal{N}(\boldsymbol{\beta}_j \mid \mathbf{m}_{i,j}, \boldsymbol{\Lambda}_{i,j}^{-1})$$
$$\propto \prod_{j=1}^{K} \exp\left(-\frac{1}{2}(\boldsymbol{\beta}_j - \mathbf{m}_{ij})^{\intercal} \boldsymbol{\Lambda}_{ij} (\boldsymbol{\beta}_j - \mathbf{m}_{ij})\right) \tag{18}$$

The EDCM mixture model posterior $p(\boldsymbol{\pi}, \boldsymbol{\beta})$ can be factorized in D sites, one for each document i with priors $p(\boldsymbol{\pi})$ and $p(\boldsymbol{\beta})$. Additionally, after defining the approximate sites, we compute the approximate posterior $q(\boldsymbol{\pi}, \boldsymbol{\beta})$ by getting the product of D approximate sites:

$$q(\boldsymbol{\pi}, \boldsymbol{\beta} \mid \boldsymbol{\alpha}', \mathbf{m}', \boldsymbol{\Lambda}'^{-1}) \propto \prod_{i}^{D} \tilde{p}_i(\boldsymbol{\pi}, \boldsymbol{\beta} \mid \boldsymbol{\alpha}_i, \mathbf{m}_i, \boldsymbol{\Lambda}_i^{-1}) \tag{19}$$

where $\boldsymbol{\alpha}'$, $\mathbf{m}'$, and $\boldsymbol{\Lambda}'$ are the parameters of the posterior distribution and can be calculated using Eqs. (20), (21), and (22), respectively. We will discuss the initialization scheme used for the approximate sites and inclusion of priors in Section 2.3.3.

$$\alpha'_j = \sum_{i}^{D} \alpha_{i,j} - D \tag{20}$$

$$\boldsymbol{\Lambda}'_j = \sum_{i}^{D} \boldsymbol{\Lambda}_{i,j} \tag{21}$$

$$\mathbf{m}'_j = \boldsymbol{\Lambda}_j'^{-1} \left(\sum_{i}^{D} \boldsymbol{\Lambda}_{i,j} \mathbf{m}_{i,j} \right) \tag{22}$$

In order to create a refinement for the approximate site $p_i(\boldsymbol{\pi}, \boldsymbol{\beta})$, we introduce a cavity distribution $q^{\setminus i}(\boldsymbol{\pi}, \boldsymbol{\beta})$ by deleting the contribution of the ith site. Thus, the cavity distribution has parameters $\boldsymbol{\alpha}^{\setminus i}$, $\boldsymbol{\Lambda}^{\setminus i}$, and $\mathbf{m}^{\setminus i}$ as shown in Eqs. (23), (24) and (25), respectively, and it is calculated as follows: $q(\boldsymbol{\pi}, \boldsymbol{\beta}) / \tilde{p}_i(\boldsymbol{\pi}, \boldsymbol{\beta})$.

$$\alpha_j^{\setminus i} = \alpha'_j - \alpha_{i,j} + 1 \tag{23}$$

$$\boldsymbol{\Lambda}_j^{\setminus i} = \boldsymbol{\Lambda}_j' - \boldsymbol{\Lambda}_{i,j} \tag{24}$$

$$\mathbf{m}_j^{\setminus i} = \boldsymbol{\Lambda}_j^{\setminus i-1}\left(\boldsymbol{\Lambda}_j^{'}\mathbf{m}_j^{'} - \boldsymbol{\Lambda}_{i,j}\mathbf{m}_{i,j}\right) \tag{25}$$

Then, we incorporate the contribution of the ith site to the cavity distribution, resulting in a tilted distribution $q^*(\boldsymbol{\pi}, \boldsymbol{\beta})$ that is an updated posterior. We normalize this new posterior using a normalizing factor Z_i to guarantee that it is a proper distribution (see equation 7).

$$\begin{aligned} q^*(\boldsymbol{\pi}, \boldsymbol{\beta}) &= \frac{1}{Z_i} p(\mathbf{x}_i \mid \boldsymbol{\beta}, \boldsymbol{\pi}) q^{\setminus i}(\boldsymbol{\pi}, \boldsymbol{\beta}) \\ &= \frac{1}{Z_i} p(\mathbf{x}_i \mid \boldsymbol{\beta}, \boldsymbol{\pi}) p(\boldsymbol{\pi}|\boldsymbol{\alpha}^{\setminus i}) \prod_j^K \mathcal{N}(\boldsymbol{\beta}_j \mid \mathbf{m}_j^{\setminus i}, \boldsymbol{\Lambda}_j^{\setminus i-1}) \end{aligned} \tag{26}$$

The normalizing factor can be then calculated by integrating out $\boldsymbol{\pi}$ and $\boldsymbol{\beta}$, obtaining the following expression for the normalization constant $Z_i(\boldsymbol{\alpha}^{\setminus i}, \mathbf{m}_j^{\setminus i}, \boldsymbol{\Lambda}_j^{\setminus i})$:

$$\begin{aligned} Z_i(\boldsymbol{\alpha}^{\setminus i}, \mathbf{m}_j^{\setminus i}, \boldsymbol{\Lambda}_j^{\setminus i}) &= \int p(\mathbf{x}_i \mid \boldsymbol{\beta}, \boldsymbol{\pi}) Dir(\boldsymbol{\pi} \mid \boldsymbol{\alpha}^{\setminus i}) \prod_j^K \mathcal{N}(\boldsymbol{\beta}_j \mid \mathbf{m}_j^{\setminus i}, \boldsymbol{\Lambda}_j^{\setminus i-1}) d\boldsymbol{\pi} d\boldsymbol{\beta} \\ &= \sum_j^K \frac{\alpha_j^{\setminus i}}{\sum_j^K \alpha_j^{\setminus i}} \int \mathcal{EDCM}(\mathbf{x}_i \mid \boldsymbol{\beta}_j) \mathcal{N}(\boldsymbol{\beta}_j \mid \mathbf{m}_j^{\setminus i}, \boldsymbol{\Lambda}_j^{\setminus i-1}) d\boldsymbol{\beta}_j \end{aligned} \tag{27}$$

However, the integration of the normalization factor is not possible since Eq. 27 is intractable, and having an analytically expression is necessary. Thus, we propose to solve this integral via Monte Carlo sampling, as shown in Eq. 28, where we take S samples from $\boldsymbol{\beta}_s \sim \mathcal{N}(\mathbf{m}^{\setminus i}, \boldsymbol{\Lambda}^{\setminus i-1})$. In order to simplify the notation, we remove the dependence of $\boldsymbol{\beta}$ on j.

$$\mathbb{E}_{p(\boldsymbol{\beta})}\left[\mathcal{EDCM}(\mathbf{x}_i \mid \boldsymbol{\beta})\right] = \frac{1}{S}\sum_{s=1}^{S} \mathcal{EDCM}(\mathbf{x}_i \mid \boldsymbol{\beta}_s) \tag{28}$$

Therefore, after rewriting the normalization factor (from Eq. 27), the following expression is obtained:

$$Z_i(\boldsymbol{\alpha}^{\setminus i}, \mathbf{m}_j^{\setminus i}, \boldsymbol{\Lambda}_j^{\setminus i}) = \sum_j^K \frac{\alpha_j^{\setminus i}}{\sum_j^K \alpha_j^{\setminus i}} \mathbb{E}_{p(\boldsymbol{\beta}_j)} \left[\mathcal{EDCM}(\mathbf{x}_i \mid \boldsymbol{\beta}_j)\right] \tag{29}$$

Finally, we minimize the KL divergence between the tilted distribution and the approximate posterior $D_{KL}(q_i^*(\boldsymbol{\pi}, \boldsymbol{\beta}) \;||\; q^{new}(\boldsymbol{\pi}, \boldsymbol{\beta}))$. This minimization is achieved by calculating the partial derivative of $\log Z_i$ with respect to the parameters of the model and matching its respective moments, as shown in the following equations:

$$\nabla_{\alpha_j^{\setminus i}} \log Z_i = \mathbb{E}_{q^*(\pi,\beta)}[\nabla_{\alpha_j^{\setminus i}} \log Dir(\pi \mid \alpha^{\setminus i})] = \Psi(\sum_j^K \alpha_j^{\setminus i}) - \Psi(\alpha_j^{\setminus i}) + \Psi(\alpha_j^{'}) - \Psi(\sum_j^K \alpha_j^{'}) \tag{30}$$

$$\nabla_{m_j^{\setminus i}} \log Z_i = \mathbb{E}_{q^*(\pi,\beta)}[\nabla_{m_j^{\setminus i}} \log \mathcal{N}(\beta_j \mid m_j^{\setminus i}, \Lambda_j^{\setminus i-1})] = \Lambda_j^{\setminus i}(m_j' - m_j^{\setminus i}) \tag{31}$$

$$\nabla_{\Lambda_j^{\setminus i}} \log Z_i = \mathbb{E}_{q^*(\pi,\beta)}[\nabla_{\Lambda_j^{\setminus i}} \log \mathcal{N}(\beta_j \mid m_j^{\setminus i}, \Lambda_j^{\setminus i-1})] = \frac{1}{2}(\Lambda_j^{\setminus i} - \Lambda_j^{'} - m_j^{'} m_j^{'\intercal} + 2m_j^{'} m_j^{\setminus i\intercal} - m_j^{\setminus i} m_j^{\setminus i\intercal}) \tag{32}$$

After matching the sufficient statistics of $\mathbb{E}_{q^*}\left[\nabla_{\alpha_j^{\setminus i}} \log Dir(\boldsymbol{\pi})\right]$, $\mathbb{E}_{q^*}\left[\nabla_{m_j^{\setminus i}} \log \mathcal{N}(\boldsymbol{\beta}_j)\right]$, and $\mathbb{E}_{q^*}\left[\nabla_{\boldsymbol{\Lambda}_j^{\setminus i}} \log \mathcal{N}(\boldsymbol{\beta}_j)\right]$ (Eqs. 30, 31, and 32), we can update the parameters of the approximate posterior $q^{new}(\boldsymbol{\pi}, \boldsymbol{\beta})$ using Eqs. (33), (34), and (35):

$$\Psi(\alpha_j^{'}) - \Psi(\sum_j^K \alpha_j^{'}) = \nabla_{\alpha_j^{\setminus i}} \log Z_i - \Psi(\sum_j^K \alpha_j^{\setminus i}) + \Psi(\alpha_j^{\setminus i}) \tag{33}$$

$$\mathbf{m}_j' = \boldsymbol{\Lambda}_j^{\setminus i-1}(\nabla_{\mathbf{m}_j^{\setminus i}} \log Z_i + \boldsymbol{\Lambda}_j^{\setminus i} \mathbf{m}_j^{\setminus i}) \tag{34}$$

$$\boldsymbol{\Lambda}_j^{'} = -2\nabla_{\boldsymbol{\Lambda}_j^{\setminus i}} \log Z_i + \boldsymbol{\Lambda}_j^{\setminus i} - \mathbf{m}_j^{'} \mathbf{m}_j^{'\intercal} + 2\mathbf{m}_j^{'} \mathbf{m}_j^{\setminus i\intercal} - \mathbf{m}_j^{\setminus i} \mathbf{m}_j^{\setminus i\intercal} \tag{35}$$

The gradient of $\log Z_i$, can be calculated analytically using Eq. (29). The values of $\boldsymbol{\alpha}^{'}$ are calculated using fixed point iteration as describe in [57]. Finally, we reuse the updated approximate posterior and remove the cavity distribution in order to obtain the update for the current approximate site $\tilde{p}_i$ as:

$$\tilde{p}_i = Z_i \frac{q^{new}(\boldsymbol{\pi}, \boldsymbol{\beta})}{q^{\setminus i}(\boldsymbol{\pi}, \boldsymbol{\beta})} \tag{36}$$

where the parameters of the ith site can be updated using the followings equations:

$$\alpha_{i,j} = \alpha_j^{'} - \alpha_j^{\setminus i} + 1 \tag{37}$$

$$\mathbf{m}_{i,j} = \left(\boldsymbol{\Lambda}_j^{'-1} - \boldsymbol{\Lambda}_j^{\setminus i-1}\right)\left(\boldsymbol{\Lambda}_j^{'}\mathbf{m}_j^{'} - \boldsymbol{\Lambda}_j^{\setminus i}\mathbf{m}_j^{\setminus i}\right) \tag{38}$$

$$\boldsymbol{\Lambda}_{i,j} = \boldsymbol{\Lambda}_j^{'} - \boldsymbol{\Lambda}_j^{\setminus i} \tag{39}$$

This procedure is repeated for all the D documents and iterated until a certain level of convergence is reached. The values of the mixing weights can be approximated by calculating its expectation with respect to the approximating posterior.

$$\mathbb{E}_q\left[\pi_j\right] = \frac{\alpha_j^{'}}{\sum_{j=1}^{K} \alpha_j^{'}} \tag{40}$$

2.3.3 A Note on Initialization and Learning Algorithm

We initialize each approximate site such that $\tilde{p}_i(\boldsymbol{\pi}, \boldsymbol{\beta}) \to 1$. In that sense, the approximate posterior is initialized with the values of the prior $q(\boldsymbol{\pi}, \boldsymbol{\beta}) = \tilde{p}_0(\boldsymbol{\pi}, \boldsymbol{\beta})$. For instance, we initialize the mixing weights uniformly, thus we consider a symmetric Dirichlet prior $\tilde{p}_0(\boldsymbol{\pi})$ with parameter value $1/K$. Consequently, for the prior $p(\boldsymbol{\beta})$, we follow an adaptation of the method of moments (MoM) described in [18]. We compute an initial $\boldsymbol{\beta}_j$ and calculate its statistics as follows: 1) we apply K-means clustering[1]; 2) apply MoM for the EDCM

[1]We use the implementation of NLTK with the cosine distance. `https://www.nltk.org/api/nltk.cluster.html`

distribution to each j component found; 3) calculate $\mathbf{m}_{0,j}$ and $\mathbf{\Lambda}_{0,j}$. It is possible to encode any prior information in the mixing weights (*i.e.* the means of the k-means clusters). Nevertheless, for the EDCM parameter $\boldsymbol{\beta}$, we find that the MoM restricts the values of $\boldsymbol{\beta}$ to be small and positive while sampling from a Gaussian distribution. This initialization scheme helps the proposed framework to stabilize while fitting the values of $\beta_{j,w} \ll 1$. Algorithm 1 illustrates the complete algorithm for EDCM Mixture Model.

Algorithm 1: Expectation Propagation (EP) algorithm for learning a EDCM Mixture model

Input : K: number of clusters; $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_D\}$: corpus

Initialize approximate sites $\tilde{p}_i$. This can be achieved by initializing its parameters $\alpha_{i,j}$, $\mathbf{m}_{i,j}$, and $\mathbf{\Lambda}_{i,j}$ for $i = 1, \ldots, D$ and $j = 1, \ldots, K$
Calculate initial values of $\boldsymbol{\alpha}_0$, $\mathbf{m}_0$, and $\mathbf{\Lambda}_0$ as described in the initialization section.
Compute $q(\boldsymbol{\pi}, \boldsymbol{\beta})$ by calculating $\boldsymbol{\alpha}'$, $\mathbf{\Lambda}'$, and $\mathbf{m}'$
while *not convergence* **do**
    for *i in* $\mathcal{X}$ **do**
        Select an approximate site $\tilde{p}_i(\boldsymbol{\pi}, \boldsymbol{\beta})$ to refine
        Compute the cavity distribution $q^{\setminus i}(\boldsymbol{\pi}, \boldsymbol{\beta})$ by removing the contribution of the selected approximate site. This is done by calculating $\boldsymbol{\alpha}^{\setminus i}$, $\mathbf{\Lambda}^{\setminus i}$, and $\mathbf{m}^{\setminus i}$
        Match moments of the tilted distribution $q^*(\boldsymbol{\pi}, \boldsymbol{\beta})$ and approximate posterior $q^{new}(\boldsymbol{\pi}, \boldsymbol{\beta})$ by minimizing $D_{KL}(q^* \parallel q^{new})$.
        Update parameters of $\tilde{p}_i(\boldsymbol{\pi}, \boldsymbol{\beta})$
    end
end
Estimate mixing weights π_j
Combine or eliminate clusters with very small weights ($\pi_j \to 0$)

2.4 Results

We evaluate the validity of the proposed framework in two tasks. First, we perform sentiment analysis in various review datasets. Next, we use the Swedish leaf dataset [74] for object recognition. In both applications, the achieved results outperform the traditional EDCM mixture model with maximum-likelihood learning approach.

2.4.1 Text clustering

Many online users employ online platforms to express opinions or experiences regarding a product or service through reviews. We exploit these data to investigate the validity of our framework on a sentiment analysis task where we know the right number of components (i.e. positive/negative, $K = 2$). For all the experiments, we use a greater number of clusters. We use a number of clusters of $K = 10$ and ignore components with very small values (i.e. $\pi_j \to 0$). We use three benchmark datasets [52, 93]: 1) Amazon Review Polarity; 2) Yelp review Polarity; 3) IMDB Movie Reviews. This section presents the details of our experimentation and its results.

Experimental setup

Before describing the experimental results of our framework, we first outline the key properties of the datasets used, as well as the setup for the experiments carried out. For each j component, at inference time, we set all values to zero except the diagonal ones from the precision matrix $\mathbf{\Lambda}_{*j}^{-1}$ for computational simplicity. Additionally, we take $S = 100$ samples from $\mathcal{N}(\mathbf{m}^{\setminus i}, \mathbf{\Lambda}^{\setminus i-1})$ and force all values to be positive. For every dataset, we analyze the effect of pre-processing. In other words, we examine whether pre-processing helps the mixture to fit the data better. We performed the following pre-processing for all datasets: 1) lowercase all text; 2) remove non-alphabetical characters; 3) remove stop words; 4) lemmatize text.

All datasets are reviews and contain two labels indicating whether the post has a positive or negative sentiment. Specifically, *Amazon Review Polarity* contains $180K$ customer reviews from products on the *Amazon.com* website. The dataset has an average of 75 words per review before pre-processing and 40 words after pre-processing. Our second dataset, *Yelp Review Polarity*, contains $560K$ user reviews from *Yelp* with an average of 133 words before pre-processing and 60 words after pre-processing. The final dataset we consider is the *IMDB movie reviews*. This dataset consists of $50K$ movie reviews with 231 and 108 words before and after pre-processing respectively.

Results

We apply the proposed framework to all the datasets described in the above section. We compare our approach with an EDCM mixture model using maximum-likelihood (ML) for learning its parameters as reported in [26]. Additionally, we evaluate the effect of pre-processing text documents when using the proposed method since in latent models (such as LDA), it has been shown that common pre-processing steps have no impact on the obtained results [69]. Thus, we evaluate our parameter learning method where pre-processing is involved (EP-P) and raw text (EP-NP). We evaluate our results in terms of precision and recall as shown in Table 1.

Table 1: Results on the three text datasets. Comparison using precision and recall for every inference method. ML: maximum-likelihood; EP-P: expectation propagation + pre-processing; EP-NP: expectation propagation + raw text.

		Dataset		
Metrics		Amazon	Yelp	IMDB
Precision	MM	50.83	89.12	64.18
	DCM	55.65	**91.01**	71.14
	ML	80.65	89.25	78.54
	EP-P	84.84	74.26	78.60
	EP-NP	**86.91**	80.50	**86.36**
Recall	MM	51.99	89.20	64.40
	DCM	63.94	91.01	**89.45**
	ML	80.88	89.28	89.33
	EP-P	81.23	**93.83**	78.45
	EP-NP	**84.82**	78.60	85.94

For the case of the *Amazon Review Polarity* dataset our framework completely outperforms the maximum-likelihood estimation by $\sim 6\%$ and $\sim 4\%$ improvement for precision and recall respectively, and thus, achieving 86.91% and 84.82%. Additionally, we notice that pre-processing causes a bad effect on the model instead of helping infer the right cluster assignments. For *Yelp Review Polarity* dataset our approach outperforms the

maximum-likelihood approach in terms of recall, meaning that the EP model is more confident at assigning the right clusters. Finally, for the *IMDB movie review* EP surpasses ML in terms of precision by a large margin $\sim 9\%$.

2.4.2 Object recognition

For object recognition, we use the Swedish leaf dataset [74] that contains 15 different types of leaves. We evaluate with 26 and 39 clusters (i.e. $K = 26, K = 39$). Mixture components π_j with very small values are ignored.

Experimental setup

The framework configuration is similar to the one used in the previous section.

Moreover, the leaf dataset contains 585 images, each corresponding to a specific specie from the following list: *Ulmus carpinifolia, Acer platanoides, Ulmus, Quercus robur, Alnus incana, Tilia, Salix fragilis, Populus tremula, Corylus avellana, Sorbus aucuparia, Prunus padus, Tilia, Populus, Sorbus hybrida, and Fagus silvatica*. Each image size is 128×128. For each image, we extracted 200 discrete features. In order to extract features from the leaves images, we use shape context [3] in which an object is assumed to be essentially captured by a finite set of its N points sampled from the internal or external contours on the object. A shape context is a descriptor for each point, which captures the distribution of the remaining points relative to the current one. As choosing more points will result in an accurate representation of the shape, we sampled 200 points from internal and external boundary of each shape image. Then, following the practice in [82], we considered each context vector as a visual word and created the bag-of-features (BoF).

Results

We compare the mixture of EDCM model with both ML and EP inference methods and report performance in terms of accuracy (see Table 2) using the leaf dataset. The proposed

Table 2: Results for object recognition on the leaf dataset. Comparison using accuracy for every inference method. ML: maximum-likelihood; EP: expectation propagation.

Inference	Accuracy	Recall
ML	94.45	-
EP ($K = 26$)	**98.12**	23.93
EP ($K = 39$)	88.76	**78.63**

model improves the accuracy of the leaf dataset. The EDCM mixture with ML gets an accuracy of 94.45 while results with EP improves accuracy by 3.67%, obtaining 98.12 when using 26 components. On the other hand, we obtain a lower accuracy with a greater number of components $K = 39$. Consequently, with a number of clusters smaller than 26 we get an average accuracy of ~ 78. On the other hand, we notice that EP with $K = 26$ gives a really high precision with low recall while the model with $K = 39$ provides a balance between precision and recall. The selection of one of these models will highly depend on the intended application.

2.4.3 Discussion

In general, the EDCM mixture with EP provides comparable results to ML estimation, and outperforms, in some cases, the previous state of the art results. We also notice that text pre-processing does not have an impact on the obtained clusters. In fact, it can have a bad effect on the inferred clusters. In the sampling schema used to solve the integral in equation 27, we use Monte Carlo samples where S determines the number of samples to be taken. We notice that $S = 100$ provides accurate estimates compared to the DCM distribution. However, in order to speed up inference, other smaller values can be used with the risk of hurting performance. On the other hand, large values could provide better performance exposing a greater computational time. We observe that the initialization scheme used in section 2.3.3 helps the proposed framework achieve not only faster convergence but also improves the performance of the obtained clusters. Finally, different

values of K provide different cluster assignments and analyzing the values of the mixture components helps to not only select the optimal number of components but it can also be used for feature selection tasks.

2.5 Conclusions

In this chapter, we propose the use of Expectation Propagation to learn a finite EDCM mixture model instead of the maximum-likelihood (ML), and as a result, incorporating some advantages that the Bayesian framework provides. EP is used to learn the model parameters and additionally, we notice that the number of clusters can be determined by ignoring or merging components with very small values of the expected mixing weights. Moreover, we propose a simple but optimal initialization scheme in order to meet the restrictions that the approximation of the DCM distribution is subject to. Given that we use the Bayesian framework, some other sources of prior information can be encoded in the model. Finally, we demonstrate the efficacy of our framework by evaluating it in sentiment analysis and shape recognition tasks. Results show the validity of our framework and obtaining comparable and superior results as opposed to using ML estimation in terms of clustering performance.

Chapter 3

Clustering Count Data with Stochastic Expectation Propagation

Clustering count vectors is a challenging task given its sparsity and high-dimensionality. An efficient generative model called EDCM has been recently proposed, as an exponential-family approximation to the Multinomial Scaled Dirichlet distribution, and has shown to offer excellent modeling capabilities in the case of sparse count data and to overcome some limitations of the frameworks based on the Dirichlet distribution. In this work, we develop an approximate Bayesian learning framework for the parameters of a finite mixture of EDCM using the Stochastic Expectation Propagation approach [48]. In this approach, we maintain a global posterior approximation that is being updated in a local way, which reduces the memory consumption, important when making inference in large datasets. Experiments on both synthetic and real count data have been conducted to validate the effectiveness of the proposed algorithm in comparison to other traditional learning approaches. Results show that SEP produces comparable estimates with traditional approaches.

3.1 Introduction

Statistical methods are excellent at modeling semantic content of text documents [46]. More specifically, document clustering is widely used in a variety of applications such as text retrieval or topic modeling, (see e.g. [20]). Words in text documents usually exhibit appearance dependencies, *i.e.*, if word w appears once, it is more probable that the same word w will appear again. This phenomenon is denominated as burstiness, which has shown to be addressed by introducing the prior information into the construction of the statistical model to obtain several computational advantages [54]. Given that the Dirichlet distribution is generally taken as a conjugate prior to the multinomial, the most popular hierarchical approach is the Dirichlet Compound Multinomial (DCM) distribution [53]. While the Multinomial distribution fails to model the words burstiness given its dependency assumption, the DCM distribution not only captures this behavior but also models text data better [53]. Furthermore, taking into account the sparsity and high-dimensionality of text data, [26] proposed the EDCM model, which approximates the DCM as a member of the exponential family. EDCM has shown to be more computationally efficient while maintaining the merits of DCM for modeling word occurrence dependency.

The Dirichlet distribution has its own limitations due to its negative covariance structure and equal confidence [50,86]. Hence, a generalization of it called the Scaled Dirichlet (SD) distribution has shown to be a good alternative as a prior to the multinomial [88]. Indeed, Multinomial scaled Dirichlet (MSD) distribution has shown to have high flexibility in count data modeling with superior performance in several challenging applications [88–90, 92]. Despite its flexibility, MSD distribution shares similar limitations to the one with DCM since its parameter estimation is slow, especially in high-dimensional spaces. Thus, [92] proposed a close exponential-family approximation called EDCM to combine the flexibility and efficiency of MSD with the desirable statistical and computational properties of the exponential family of distributions, including sufficiency. EDCM

has shown to reduce the complexity and computational efforts, especially for sparse and high-dimensional data.

Moreover, finite mixture models have been frequently used as an efficient flexible statistical approach to cluster data into homogeneous groups [56]. In mixture models, three crucial issues need to be addressed, including the choice of the component's densities, the estimation of the mixture parameters, and the selection of the number of clusters that best describes the data. In order to learn the parameters of a mixture model, both frequentist and Bayesian approaches have been used. Bayesian learning commonly involves statistical modeling and inference methods. Since parameter learning is one of the encountered challenges in mixture models, the maximum-likelihood method via the expectation-maximization (EM) algorithm is typically used for learning the parameters of the EDCM mixture model. Even though that the maximum-likelihood method shows fast parameter learning, it carries some disadvantages since it provides point estimates and is highly dependant on parameter initialization [4] while in the Bayesian setting we can compute an approximate posterior and measure uncertainty. In fact, deterministic Bayesian inference techniques (*e.g.* variational inference or expectation propagation) allow good approximations by introducing a prior distribution that is much better in approximating the full posterior.

In this work, we study the application of the Bayesian framework for learning the exponential-family approximation to the Multinomial Scaled Dirichlet (EMSD) mixture model which has been shown to be an appropriate distribution to model the burstiness in high-dimensional feature space. In particular, we propose a learning approach for an EDCM mixture model using Stochastic Expectation Propagation (SEP) [48] for parameter estimation. Indeed, SEP combines both Assumed Density Filtering (ADF) and Expectation Propagation (EP) in order to scale to large datasets while maintaining accurate estimations. Only EP is usually more accurate than methods such as Variational Inference (VI) and Markov Chain Monte Carlo (MCMC) [4, 59], and SEP solves some of the

problems encountered when using EP given that the number of parameters increase according to number of datapoints. Thus, SEP is a deterministic approximate inference method that prevents memory overheads when increasing the number of data points. EP has shown to be an appropriate generalization in the case of Gaussian mixture model [61], hierarchical models such as LDA [59] or even infinite mixture models [28]. Furthermore, SEP has been used with Deep Gaussian process [21], showing the benefits of scalable Bayesian inference and outperforming traditional Gaussian process. The contributions of this chapter are summarized as follows: 1) we show that SEP can provide effective parameter estimates when dealing with large datasets; 2) we derive foundations to learn an EDCM mixture model using SEP; 3) we exhaustively evaluate the proposed approach on synthetic and real count data and compare the performance with other models and learning approaches.

The rest of this chapter is organized as follows. First, Section 3.2 revisits the approximation to the Multinomial Scaled Dirichlet (EMSD) distribution used to tackle the burstiness phenomenon efficiently for high-dimensional count data. In Section 3.3, we outline the EDCM mixture model, describe the SEP learning approach, and derive the complete learning algorithm. Section 3.4 is devoted to the experimental results on both synthetic and real high-dimensional count data. Finally, conclusions are given in Section 3.5.

3.2 The Exponential-family Approximation to MSD Distribution

We start with a brief review of the Multinomial Scaled Dirichlet distribution (MSD) recently introduced by [88]. We are given a dataset $\mathcal{X}$ with D samples $\mathcal{X} = \{\mathbf{x_i}\}_{i=1}^{D}$, each $\mathbf{x}_i$ is a vector of count data (*e.g.* a text document or an image, represented as a frequencies vector of words or visual words, respectively). We assume that each data set has a vocabulary of size V.

The Multinomial distribution with positive parameters $p = (p_1, \ldots, p_V)$ is commonly used to model features involving counts:

$$\mathcal{M}(\boldsymbol{x} \mid \boldsymbol{p}) = \frac{n!}{\prod_{w=1}^{V} x_w!} \prod_{w=1}^{V} p_w^{x_w} \tag{41}$$

where $n = \sum_{w=1}^{V} x_w$ is the document length.

However, the Multinomial distribution is not appropriate when analyzing the burstiness of words [53]. This is due to the fact that according to the Multinomial distribution, words follow the i.i.d assumption, but in real data, there is actually an occurrence dependency such that if a word appears once, it is more likely to appear again [44].

The hierarchical approach of DCM considers the count vector to be generated by a multinomial distribution whose parameters are generated by the Dirichlet distribution. That is, in a specific document, for example, the Multinomial is linked to particular subtopics, and thus, it makes the emission of some words more likely than others. This gives it the ability to handle burstiness, even for rare words. The limitations of the Dirichlet motivated the scholars to use different interesting alternative priors for the multinomial including the generalized Dirichlet [13], and the Beta-Liouville [15]. Recently, [88] proposed a more flexible generative model to deal with burstiness phenomenon, called the Multinomial scaled Dirichlet (MSD), which is the composition of the Multinomial and Scaled Dirichlet in the same way that the DCM is the composition of the Multinomial and the Dirichlet. In this model, the prior information is introduced using the scaled Dirichlet distribution, which is a generalization of Dirichlet distribution that is obtained after some perturbation and powering operations to a Dirichlet random composition, operations that define a vector-space structure in the simplex [65]. The scaled Dirichlet with a scale $\boldsymbol{\rho}$ and shape $\boldsymbol{\nu}$ parameter is defined as:

$$\mathcal{SD}(\boldsymbol{p} \mid \boldsymbol{\rho}, \boldsymbol{\nu}) = \frac{\Gamma(s)}{\prod_{w=1}^{V} \Gamma(\rho_w)} \frac{\prod_{w=1}^{V} \nu_w^{\rho_w} p_w^{\rho_w - 1}}{\left(\sum_{w=1}^{V} \nu_w p_w\right)^s} \tag{42}$$

where $s = \sum_{w=1}^{V} \rho_w$ is the sum of the scale parameter.

Thus, the MSD is the marginal distribution defined by integrating out the probability parameter $\boldsymbol{p}$ (i.e. $\int p(\boldsymbol{x} \mid \boldsymbol{p}) p(\boldsymbol{p} \mid \boldsymbol{\rho}, \boldsymbol{\nu})$), obtaining a discrete distribution known as the Multinomial Scaled Dirichlet (MSD) distribution [88], which is given by:

$$\mathcal{MSD}(\mathbf{x} \mid \boldsymbol{\rho}, \boldsymbol{\nu}) = \frac{n!}{\prod_{w=1}^{V} x_w!} \frac{\Gamma(s)}{\Gamma(s+n) \prod_{w=1}^{V} \nu_w^{x_w}} \prod_{w=1}^{V} \frac{\Gamma(x_w + \rho_w)}{\Gamma(\rho_w)} \tag{43}$$

Notice that the authors in [88] use an approximation to $\left(\sum_{w=1}^{V} \nu_w p_w\right)^{\sum_{w=1}^{V} x_w} \approx \prod_{w=1}^{V} \nu_w^{x_w}$. Observe that when $\boldsymbol{\nu} = 1$, we obtain the Dirichlet Compound Multinomial (DCM) distribution [53]. Similar to DCM, this model, MSD, has an intuitive interpretation representing the Scaled Dirichlet as a general topic and the Multinomial as a document-specific subtopic, making some words more likely in a document $\mathbf{x}$ based on word counts.

The text documents representation is very sparse as many words in the vocabulary do not appear in most of the documents. Thus, in [92], the authors note that using only the non-zero values of $\mathbf{x}$ is computationally efficient since $x_w! = 1$, $\nu_w^{x_w} = 1$ and $\Gamma(x_w + \rho_w)/\Gamma(\rho_w) = 1$ when $x_w = 0$. Moreover, since in high dimensional data the parameters are very small, [26], the following fact for small values of ρ when $x \geq 1$ was used in [92]:

$$\lim_{\rho \to 0} \frac{\Gamma(x+\rho)}{\Gamma(\rho)} - \Gamma(x)\rho = 0 \tag{44}$$

Thus, being able to approximate $\Gamma(x_w + \rho_w)/\Gamma(\rho_w) = \Gamma(x_w)\rho_w$ and using the fact that $\Gamma(x_w) = (x_w - 1)!$ leads to an approximation of the MSD distribution known as the Exponential-family approximation to the MSD distribution (EMSD), given by:

$$\mathcal{EMSD}(\mathbf{x} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{n!}{\prod_{w:x_w \geq 1}^{V} x_w} \frac{\Gamma(s)}{\Gamma(s+n)} \prod_{w:x_w \geq 1}^{V} \frac{\alpha_w}{\beta_w^{x_w}} \tag{45}$$

The parameters of the EDCM distribution are denoted with $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to distinguish them from the MSD parameters.

3.3 EMSD Mixture Model

This section gives groundwork of the main components that our work is built on and introduces the notation used throughout the present chapter.

3.3.1 Clustering Model

We assume that we are given D documents drawn from a finite number of EDCM distributions, and each $\mathbf{x}_i$ document is composed of V words. $K \geq 1$ represents the number of mixture components. Thus, a document is drawn from its respective component j as follows: $\mathbf{x_i} \sim \mathcal{EMSD}(\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j)$.

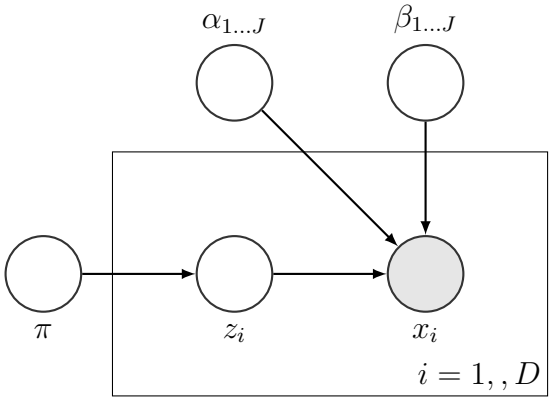

Figure 2: EMSD mixture model.

In a mixture model, a latent variable $\mathcal{Z} = \{\mathbf{z}_i\}_{i=1}^{D}$ is introduced for each $\mathbf{x}_i$ document in order to represent the component assignment. We posit a Multinomial distribution for the component assignment such that $\mathbf{z}_i \sim Mult(1, \boldsymbol{\pi})$ where $\boldsymbol{\pi} = \{\pi_j\}_{j=1}^{K}$ represents the mixing weights, and they are subject to the constraints $0 < \pi_j < 1$ and $\sum_j \pi_j = 1$ (Figure 2 illustrates the graphical model for the mixture model). In other words, $\mathbf{z}_i$ is a K-dimensional indicator vector containing a value of one when document $\mathbf{x}_i$ belongs to the component j, and zero otherwise. Note that in this setting the value of $z_{ij} = 1$ acts as the selector of the component that generates $\mathbf{x}_i$ document with parameters $\boldsymbol{\alpha}_j$ and $\boldsymbol{\beta}_j$;

hence, $p(\mathbf{z}_i \mid \boldsymbol{\pi}) = \pi_j$. Thus, the full posterior is in equation 46.

$$p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta} \mid \mathcal{X}) \propto p(\boldsymbol{\pi})p(\boldsymbol{\alpha})p(\boldsymbol{\beta}) \prod_{i}^{D} \sum_{j}^{K} \pi_j p(\mathbf{x}_i \mid \boldsymbol{\alpha}_j, \boldsymbol{\beta}_j) \tag{46}$$

3.3.2 Parameter Learning

We use SEP in order to learn the parameters of the mixture model. We start by partitioning the likelihood in D sites and define a global approximating site for each of the latent variables ($\boldsymbol{\pi}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$). Theoretically, any distribution belonging to the exponential family can be used for the sites. We use a Gaussian distribution for the parameters of the EDCM distribution in order to facilitate calculations [51]. For the mixture weights, we use a Dirichlet distribution since it belongs to the $K-1$ simplex and fits the constraints imposed by the mixing weights. Eqs. 47, 48 and 49 illustrate the choices for the approximate sites.

$$\tilde{p}(\boldsymbol{\pi}) \propto \prod_{j} \pi_j^{a_j} \tag{47}$$

$$\tilde{p}(\boldsymbol{\alpha}) = \prod_{j}^{K} \mathcal{N}(\boldsymbol{\alpha}_j \mid \boldsymbol{m}_j, p_j^{-1}) \tag{48}$$

$$\tilde{p}(\boldsymbol{\beta}) = \prod_{j}^{K} \mathcal{N}(\boldsymbol{\beta}_j \mid \boldsymbol{n}_j, q_j^{-1}) \tag{49}$$

Once have defined the global approximate site, we compute the approximate posterior $q(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ by introducing the priors and the average effect of the global site:

$$
\begin{aligned}
q(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto & p(\boldsymbol{\pi}, \boldsymbol{a}^0) \tilde{p}(\boldsymbol{\pi} \mid \boldsymbol{a})^D \\
& \prod_j^K p\left(\boldsymbol{\alpha}_j \mid \boldsymbol{m}_j^0, (p_j^0)^{-1}\right) \tilde{p}\left(\boldsymbol{\alpha}_j \mid \boldsymbol{m}_j, (p_j)^{-1}\right)^D \\
& p\left(\boldsymbol{\beta}_j \mid \boldsymbol{n}_j^0, (q_j^0)^{-1}\right) \tilde{p}\left(\boldsymbol{\beta}_j \mid \boldsymbol{n}_j, q_j^{-1}\right)^D
\end{aligned}
$$

The approximate posterior distribution have the parameters illustrated in Eqs. 50, 51, 52, 53 and 54.

$$
\boldsymbol{a}' = 1 + \boldsymbol{a}^0 + D\boldsymbol{a} \tag{50}
$$

$$
(p_j^{'})^{-1} = (p_j^0 + Dp_j)^{-1} \tag{51}
$$

$$
\boldsymbol{m}_j^{'} = (p_j^{'})^{-1}(p_j^0 \boldsymbol{m}_j^0 + Dp_j \boldsymbol{m}_j) \tag{52}
$$

$$
(q_j^{'})^{-1} = (q_j^0 + Dq_j)^{-1} \tag{53}
$$

$$
\boldsymbol{n}_j^{'} = (q_j^{'})^{-1}(q_j^0 \boldsymbol{n}_j^0 + Dq_j \boldsymbol{n}_j) \tag{54}
$$

Consequently, we introduce a cavity distribution by removing the contribution of one of the copies of the global site. The cavity distribution has parameters $\boldsymbol{a}^{\setminus 1}$, $\left(p_j^{\setminus 1}\right)^{-1}$, $\boldsymbol{m}_j^{\setminus 1}$, $\left(q_j^{\setminus 1}\right)^{-1}$, and $\boldsymbol{n}_j^{\setminus 1}$ illustrated in Eqs. 55, 56, 57, 58 and 59 that are calculated as follows: $q(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) / \tilde{p}_i(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$

$$
\boldsymbol{a}^{\setminus 1} = \boldsymbol{a}^{'} - \boldsymbol{a}; \tag{55}
$$

$$\left(p_j^{\setminus 1}\right)^{-1} = \left(p_j^{'} - p_j\right)^{-1} \tag{56}$$

$$\boldsymbol{m}_j^{\setminus 1} = \left(p_j^{\setminus 1}\right)^{-1}\left(p_j^{'}\boldsymbol{m}_j^{'} - p_j\boldsymbol{m}_j\right) \tag{57}$$

$$\left(q_j^{\setminus 1}\right)^{-1} = \left(q_j^{'} - q_j\right)^{-1} \tag{58}$$

$$\boldsymbol{n}_j^{\setminus 1} = \left(q_j^{\setminus 1}\right)^{-1}\left(q_j^{'}\boldsymbol{n}_j^{'} - q_j\boldsymbol{n}_j\right) \tag{59}$$

We use the cavity distribution and incorporate the ith site, resulting in the tilted distribution $\hat{p} = \frac{1}{Z_i} p_i q^{\setminus 1}$. We use this distribution to compute the KL divergence with the approximate distribution, which is equivalent to matching the moments. However, in this case, matching the moments leads to another analytically intractable integral (i.e. $Z_i = \sum_j^K \frac{a_j^{\setminus 1}}{\sum_k^K a_k^{\setminus 1}} \mathbb{E}_{p(\alpha_j,\beta_j)}\left[p(x_i \mid \alpha_j, \beta_j)\right]$). Thus, we compute this integral via Monte Carlo sampling. After matching the moments, we obtain the parameters for an updated approximate posterior (Eq. 60, 61, 63, 64 and 62).

$$\Psi(a_j^{'}) - \Psi(\sum_j^K a_j^{'}) = \Psi(a_j^{\setminus 1}) - \Psi(\sum_j^K a_j^{\setminus 1}) + \nabla_{a_j^{\setminus 1}} \log Z_i \tag{60}$$

$$\boldsymbol{m}_j^{'} = \boldsymbol{m}_j^{\setminus 1} + \left(p_j^{\setminus 1}\right)^{-1} \nabla_{\boldsymbol{m}_j^{\setminus 1}} \log Z_i \tag{61}$$

$$\boldsymbol{n}_j^{'} = \boldsymbol{n}_j^{\setminus 1} + \left(q_j^{\setminus 1}\right)^{-1} \nabla_{\boldsymbol{n}_j^{\setminus 1}} \log Z_i \tag{62}$$

$$p_j^{'} = \left(p_j^{\setminus 1}\right)^{-1}\left(2\nabla_{\left(p_j^{\setminus 1}\right)^{-1}} \log Z_i + p_j^{\setminus 1}\right)\left(p_j^{\setminus 1}\right)^{-1} - \left(\boldsymbol{m}_j^{'} - \boldsymbol{m}_j^{\setminus 1}\right)\left(\boldsymbol{m}_j^{'} - \boldsymbol{m}_j^{\setminus 1}\right)^{\intercal} \tag{63}$$

$$q_j^{'} = \left(q_j^{\backslash 1}\right)^{-1} \left(2\nabla_{\left(q_j^{\backslash 1}\right)^{-1}} \log Z_i + q_j^{\backslash 1}\right) \left(q_j^{\backslash 1}\right)^{-1} - \left(\boldsymbol{n}_j^{'} - \boldsymbol{n}_j^{\backslash 1}\right) \left(\boldsymbol{n}_j^{'} - \boldsymbol{n}_j^{\backslash 1}\right)^{\intercal} \tag{64}$$

The values of $\boldsymbol{a}^{'}$ are calculated using fixed point iteration as describe in [57]. Using this updated approximate posterior, we remove the cavity distribution in order to obtain an approximation to the ith site (Eq. 65 to Eq. 69).

$$\boldsymbol{a} = \boldsymbol{a}' - \boldsymbol{a}^{\backslash 1} \tag{65}$$

$$(p_j)^{-1} = (p_j^{'} - p_j^{\backslash 1})^{-1} \tag{66}$$

$$\boldsymbol{m}_j = (p_j)^{-1} \left(p_j^{'}\boldsymbol{m}_j^{'} - p_j^{\backslash 1}\boldsymbol{m}_j^{\backslash 1}\right) \tag{67}$$

$$(q_j)^{-1} = (q_j^{'} - q_j^{\backslash 1})^{-1} \tag{68}$$

$$\boldsymbol{n}_j = (q_j)^{-1} \left(q_j^{'}\boldsymbol{n}_j^{'} - q_j^{\backslash 1}\boldsymbol{n}_j^{\backslash 1}\right) \tag{69}$$

Finally, we use damping to partially update the global approximate site. First, we update the parameters of the global site as follows $\Theta^{new} = (1-\eta)\Theta^{old} + \eta\Theta_i$ where Θ^{old} are the current parameters of the global site, and Θ_i are the parameters for the approximation of a single likelihood. Then, we introduce the global approximate site in the approximate distribution. The learning approach is described in the algorithm 2.

3.4 Experimental Results

In this section, we describe the experiments carried out to test the validity of the proposed method on both synthetic and real count data.

Algorithm 2: Stochastic Expectation Propagation (SEP) algorithm for learning a EMSD Mixture model

Input : K: number of clusters; $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_D\}$: corpus; $p_0(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$: prior knowledge

Initialize the approximate site $\tilde{p}(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$.
If priors are not provided, initialize them to 1 (i.e. $p_0(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$=1)
Compute the approximate distribution $q(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ by calculating the average effect $\tilde{p}(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})^D$ of the likelihood and introducing the priors p_0
while *not convergence* **do**
    for x_i *in* $\mathcal{X}$ **do**
        Compute the cavity distribution $q^{\setminus 1}(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ by removing the contribution of one of the copies of the approximate site.
        Match moments of the tilted distribution $\hat{p}(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ and approximate posterior $q^{new}(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ by minimizing $D_{KL}(\hat{p} \parallel q^{new})$.
        Compute the parameters of a revised approximate site after matching the moments.
        Make a partial update to the approximate site and include the approximate site in the approximate distribution.
    end
end
Estimate mixing weights π_j

3.4.1 Synthetic dataset

We create a synthetic dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{D}$ by using the probabilistic mixture model with $D = 210$ data points. We use $K = 3$ components each is an EDCM distribution where the mixing weights are uniformly sampled. For simplicity, we set a fixed value of 1 for the scale parameter of the Scaled Dirichlet. Since the shape parameter is commonly $\alpha_w \ll 1$ [26], we sample from a Beta distribution.

We initialize the priors of the model with covariance matrix $5\boldsymbol{I}$ and $3\boldsymbol{I}$ for the scale and shape parameter. Random values are used for the prior means and mixing weights parameter. We set a step size of $\eta = 0.1$ and approximate the posterior using SEP. Table 3 show the obtained estimates. The mixing weights can be estimated using the expected value of π_j; for instance, $\mathbb{E}[\pi_j] = a_j^{'} / \sum_{j=1}^{K} a_j^{'}$.

The used parameters as well as the estimated values are shown in Table 3. We notice that estimates are very close to the target values. Since we need to store only the local

Table 3: Original parameters and estimated parameters for the mixture of EMSD using the proposed approach.

j	π	α	β
		Real	
1	0.333	$[0.610, 0.318, 0.646]$	$\mathbf{1}$
2	0.333	$[0.556, 0.188, 0.848]$	$\mathbf{1}$
3	0.334	$[0.129, 0.891, 0.507]$	$\mathbf{1}$
		Estimation	
1	0.335	$[0.663, 0.305, 0.676]$	$[1.082, 1.055, 1.062]$
2	0.332	$[0.573, 0.098, 0.720]$	$[0.963, 1.027, 0.996]$
3	0.333	$[0.193, 0.858, 0.527]$	$[1.087, 0.976, 1.002]$

and global parameters, we emphasize the fact that SEP reduces memory consumption allowing us to scale EP.

3.4.2 Sentiment Analysis

We analyze the problem of sentiment analysis in the setting when online users employ online platforms to express opinions or experiences regarding a product or service through reviews. We exploit these data to investigate the validity of our framework where we know the right number of components (i.e. positive/negative, $K = 2$). We use three benchmark datasets [52,93]: 1) Amazon Review Polarity; 2) Yelp review Polarity; 3) IMDB Movie Reviews. This section presents the details of our experimentation and its results.

Before describing the experimental results, we first outline the key properties of the datasets and the performed setup. We pre-process the dataset as follows: 1) lowercase all text; 2) remove non-alphabetical characters; 3) lemmatize text. All datasets are reviews and contain two labels indicating whether the post has a positive or negative sentiment.

Amazon Review Polarity contains $180k$ customer reviews that span a period of 18 years, for products on the *Amazon.com* website. The dataset has an average of 75 words per review with a vocabulary size of over $55k$ unique words.

Yelp Review Polarity contains $560k$ user reviews from *Yelp* with an average of 133 words with $> 85k$ unique words. The Yelp dataset contains a polarity label by considering stars 1 and 2 negative, and 3 and 4 positive reviews about local businesses.

IMDB movie reviews this dataset consists of $50K$ movie reviews with an average 231 words per review and a vocabulary size of over $76k$ unique words. Ratings on IMDB are given as star values $\in [1, 10]$ which were linearly mapped to $[0, 1]$ to use as document labels; negative and positive, respectively.

We compare the clustering performance of EDCM mixture model using the proposed SEP to different modeles with the same approach and different learning techniques such as Expectation Propagation (EP), and maximum-likelihood (ML) for parameter estimation. More precisely, we compared to the following models that use maximum-likelihood for estimating its parameters. Firstly, we have a mixture of Multinomials (MM) [16]. Even though the MM is appropriate for modeling common words, not words burstiness problem, we add it to the comparison to evaluate its predictive power. Next, we make a comparison with different models that capture the words bustiness problem such as Dirichlet Compound Multinomial (DCM) [53], Exponential-family Approximation to DCM (EDCM) [26], the Multinomial Scaled Dirichlet (MSD) [88], and the Exponential-family Approximation to MSD (EMSD) [92]. Furthermore, we compare to the performance of EDCM mixture model in case of considering EP for parameter estimation as we have recently proposed in [80]. We evaluate the performance of the considered models according to precision and recall as illustrated in Table 4.

In general, most models are superior than a Multinomial mixture model (except for Yelp dataset). We notice that SEP gives comparable results to the EDCM model in terms of precision and recall. Additionally, we evaluate an EDCM mixture that uses EP for parameter learning where we can assume that SEP is computing similar approximations to EP with the advantage that there is no need to store the parameters for each of the approximate sites. One of the main advantages is that we only store the local and global

Table 4: Results on the three text datasets. Comparison using precision and recall. ML: maximum-likelihood; EP: expectation propagation; SEP: sthocastic expectation propagation.

Metrics		Dataset		
		Amazon	Yelp	IMDB
Precision	ML-MM	50.83	89.12	64.18
	ML-DCM	55.65	91.01	71.14
	ML-EDCM	80.65	89.25	78.54
	EP-EDCM	86.91	80.50	86.36
	ML-MSD	82.21	86.96	84.00
	ML-EMSD	83.31	87.23	85.00
	SEP-EMSD (ours)	86.35	82.83	86.83
Recall	ML-MM	51.99	89.20	64.40
	ML-DCM	63.94	91.01	89.45
	ML-EDCM	80.88	89.28	89.33
	EP-EDCM	84.82	93.83	85.94
	ML-MSD	82.21	87.09	84.00
	ML-EMSD	83.57	87.28	86.00
	SEP-EMSD (ours)	83.91	90.02	87.64

parameters, reducing memory usage. More specifically, for the Amazon dataset, EP and SEP are superior in terms of precision and recall compared with most models that use maximum-likelihood estimation. Our intuition is that the length of documents plays a critical role in parameter estimation. That is, in the Amazon dataset, for example, we obtain better precision and recall using a Bayesian approach given that the document length is relatively shorter than in the other two datasets.

3.5 Conclusions

In this chapter, we propose a Stochastic Expectation Propagation (SEP) algorithm to learn a finite EDCM mixture model. We derive the mathematical framework using SEP, and since performing moment matching leads to an intractable integral, we use sampling in order to compute its moments. Then, we evaluate the proposed approach on both synthetic and real data and notice that SEP-EMSD provides comparable results to traditional approaches and in some cases being superior. Although we evaluated the proposed learning method with text data, we can use any type of count data such as a clustering of visual words for images or videos. It is noticeable that SEP does not need a site per data point and similar to variational inference maintains a global posterior approximation that is updated locally and reduces memory consumption.

Chapter 4

Improving classification using topic correlation and Expectation Propagation

Probabilistic topic models are broadly used to infer meaningful patterns of words over a mixture of latent topics that are commonly used for statistical analyses or as a proxy for supervised tasks. However, models such as Latent Dirichlet Allocation (LDA) assume independence between topic proportions due to the nature of the Dirichlet distribution; this effect is captured with other distributions such as the logistic normal distribution, resulting in a complex model. In this chapter, we develop a probabilistic topic model using the generalized Dirichlet distribution (LGDA) in order to capture topic correlation while maintaining conjugacy. We make use of Expectation Propagation to approximate the posterior, resulting in a model that achieves more accurate inferences compared to variational inference. We evaluate the convergence of EP compared with the classical LDA by comparing the approximation to the marginal distribution. We show the obtained topics by LGDA and evaluate its predictive performance in two text classification tasks, outperforming the vanilla LDA.

4.1 Introduction

Topic models are among the best-known models to automatically organize documents. Especially, probabilistic topic models [5] have received great attention from the research community. They use statistical methods for uncovering topics from a collection of documents and are commonly used for annotating or organizing documents. Latent Dirichlet Allocation (LDA) [12] was proposed as an improvement of probabilistic Latent Semantic Analysis [38,39] and has become the most popular topic model since its introduction. Many variations have been introduced leading to applications [20] in a variety of domains. For instance, they are used in academics for bibliometrics [32], labeling groups of publications [78], entity disambiguation [71], and the author-topic model [67] that captures information not only about documents but also authors. LDA has also been used successfully in applications for computer vision [7,29,47,72] commonly using a representation of visual words. Other applications can be found in areas such as healthcare [49], social sciences [66], and psychology [73].

These applications have been possible due to the flexibility of the LDA model. LDA can be extended with other more complex models and adapted to a specific problem. For instance, DiscLDA [45] is an extension of LDA for dimensionality reduction and classification that uses a linear transformations. On the other hand, other models deal with the exchangeability assumption made by LDA for word order [24,35] and dynamic topic models for document order [9]. LDA assumes that the number of topics is known beforehand. However, in real applications, this is not always the case. The number of topics can be learned using a non-parametric approach of Hierarchical Dirichlet Process [81]. And going even further, hierarchies of topics [6] can be modeled using the Nested Chinese restaurant process. Features as these manifest the importance of topic models since they can potentially improve the experience and performance of information retrieval tasks. The extensions of LDA introduced so far aim at learning unsupervised representations

only. The supervised LDA [55] uses a response variable to tackle prediction tasks. Finally, other extensions allow to model correlation between topics; we will introduce these models further since are related to this work.

It is noticeable that the applicability of topic models are endless and due to digitalization, there is an exponential growth of information available online. Thus, organizing and annotating those documents can be overwhelming and obtaining better topic models can substantially ease these tasks. For doing so, a lot of emphasis has been put in approximate inference since these models need to compute a posterior distribution which is intractable. Commonly sampling methods or deterministic approaches are used to deal with this intractable integral. For instance, Markov Chain Monte Carlo (MCMC), a sampling method, is usually implemented using a Gibbs sampling algorithm [34, 76]. Similarly, there are deterministic approaches such as Expectation Propagation (EP) and [60] and Variational Inference (VI) [8]. VI has been an active area of research having variations that are much faster and scale to great amounts of data by using stochastic optimization [36,37] or Autoencoding variational Bayes [42,75] that uses neural networks for approximating the posterior distribution.

In this work, we introduce a variation of LDA that models topic correlations leveraging the advantages of EP for approximating the posterior distribution. Topic correlations are important when, for example, a document about sports has content about soccer and athletics but lacks information about basketball. This correlation cannot be captured by LDA for the intrinsic nature of the Dirichlet distribution. However, the Generalized Dirichlet (GD) distribution is a generalization of the Dirichlet distribution that solves the limitations of its negative covariance matrix. It has been used successfully with count data [13], and apart from solving the restrictions of the Dirichlet distribution, maintains conjugacy in the LDA model. EP factorizes the joint distribution for later combining each factor with an approximation, and as a result, obtaining an overall approximation of the

posterior distribution. This is appealing for models such as LDA since data partition allows EP to be distributed and scale to large datasets. In addition, EP has shown to obtain a better approximation than VI [59], which are biased.

The rest of this chapter is organized as follows. First, Section 4.2 revisits the core methods upon our work is built on and related work in Section 4.3. Next, in Section 4.4, we outline the LGDA model, describe the expectation propagation approach, and derive a complete learning approach. Section 4.5 describes our experimental setup and evaluation of our proposed method. Finally, we conclude the chapter in Section 4.6.

4.2 Background

This section gives groundwork of the main components upon our work is built on and introduces the notation used throughout this work.

4.2.1 Latent Dirichlet Allocation

LDA [12] is the most popular probabilistic topic model and since its introduction, it has become the most conventional and known unsupervised topic model for the discovery of latent topics. It can be described as a generative model, meaning that uses a probabilistic approach allowing to generate documents.

Following this generative process, each topic $\boldsymbol{\beta}_k$ is a distribution over a vocabulary V and a document has a mixture of topics $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)$, where K is the number of topics, which has to be known beforehand. All documents in the corpus share the topics $\boldsymbol{\beta}$, but each document can express a topic in a different proportion $\boldsymbol{\theta}_d$. The generative process continues by drawing a word $w_{d,n}$ from topic $\beta_{z_{d,n}}$, where $z_{d,n}$ is the topic assignment for the word $w_{d,n}$. The topic assignment $z_{d,n}$ is drawn from a distribution over the document proportion $\boldsymbol{\theta}_d$.

Commonly, the document proportion is modeled with a Dirichlet distribution, and the topics and words with a Multinomial distribution. However, the evidence $p(\boldsymbol{w})$ of this

model is intractable due to the coupling of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ [25]. Thus, the posterior is frequently approximated with VI using the mean-field variational family, and by integrating out the latent variables, LDA is capable to infer the topic structure of a set of documents.

4.2.2 Generalized Dirichlet distribution

A Dirichlet distribution can only capture negative correlations due to its negative covariance matrix. Additionally, when it is used as a prior, poses only one degree of freedom which hinders the ability to introduce variance information to each component of the random vector. Therefore the GD distribution [23, 85] was introduced to alleviate these problems. It has positive parameters $\boldsymbol{\alpha} = \alpha_1, \ldots, \alpha_K$ and $\boldsymbol{\kappa} = \kappa_1, \ldots, \kappa_K$, and a random vector $\boldsymbol{\theta} = \theta_1, \ldots, \theta_K$, where $\sum_k^K \theta_k \leq 1$ and $0 < \theta_k < 1$ for $k = 1, \ldots, K$. GD's PDF is illustrated in equation 70.

$$p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) = \prod_k^K \frac{\Gamma(\alpha_k + \kappa_k)}{\Gamma(\alpha_k)\Gamma(\kappa_k)} \theta_k^{\alpha_k - 1} (1 - \sum_{j=1}^{k} \theta_j)^{\gamma_k}, \tag{70}$$

where $\gamma_k = \kappa_k - \alpha_{k+1} - \kappa_{k+1}$ for $k = 1, \ldots, K-1$ and $\gamma_K = \kappa_K - 1$; $\Gamma(\cdot)$ is the Gamma function. The mean and variance are shown in equation 71 and 72 respectively.

$$\mu_k = \frac{\alpha_k}{\alpha_k + \kappa_k} \prod_{j=1}^{k-1} \frac{\kappa_j}{\alpha_j + \kappa_j} \tag{71}$$

$$Var(\theta_k) = \mu_k \left(\frac{\alpha_k + 1}{\alpha_k + \kappa_k + 1} \prod_{j=1}^{k-1} \frac{\kappa_j + 1}{\alpha_j + \kappa_j + 1} - \mu_k \right) \tag{72}$$

Additionally, equation 73 illustrates the covariance matrix, which has a more general structure. For instance, the Dirichlet distribution is just an special case of the GD distribution when $\kappa_k = \alpha_{k+1} + \kappa_{k+1}$.

$$Cov(\theta_m, \theta_n) = \mu_n \left(\frac{\alpha_m}{\alpha_m + \kappa_m + 1} \prod_{j=1}^{m-1} \frac{\kappa_j + 1}{\alpha_j + \kappa_j + 1} - \mu_m \right) \tag{73}$$

It is noteworthy that the GD distribution has K degrees of freedom which makes it more flexible and suitable for modeling correlated topics.

4.3 Related Work

The work in [59] proposes an inference alternative using Expectation Propagation (EP) for LDA model that does not bound the marginal probability as in [12] and leads to higher accuracy. However, in general, the LDA model is incapable of capturing topic correlation due to the limitation of the Dirichlet distribution for the document-topic probability. The Correlated Topic Model (CTM) [10] is proposed in order to capture a correlation of the topic proportions using a logistic normal distribution which results in a complicated model since the conjugacy with the Multinomial distribution is lost. Thus, [22] showed that the CTM can be modeled using a Generalized Dirichlet distribution (denominated GD-LDA or LGDA), maintaining conjugacy and leading to faster inference. Finally, the work of [1] and [40] propose inference alternatives to the LGDA model using collapsed variational bayes inference and variational bayes inference, respectively.

4.4 Latent Generalized Dirichlet Allocation

This section provides an overview of the LGDA model and an approach to perform inference and estimation using expectation propagation.

4.4.1 Model

LGDA is a generative probabilistic model for count data. The generative process is similar to the vanilla LDA [12] with the difference that document-topic proportions $\boldsymbol{\theta}_d$ are drawn from a GD distribution.

1. Choose $\boldsymbol{\theta} \sim GenDir(\boldsymbol{\alpha}, \boldsymbol{\kappa})$

2. For each of the N words $\boldsymbol{w}_n$:

   (a) Choose a topic $\boldsymbol{z}_n \sim Multinomial(\boldsymbol{\theta})$

   (b) Choose a word $\boldsymbol{w}_n$ from $p(\boldsymbol{w}_n \mid \boldsymbol{z}_n, \boldsymbol{\beta})$

The probabilistic graphical model of LDA is depicted in figure 3. The model has the corpus level hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\kappa}$ for the prior GD distribution and β for the topics. Words are observed and represented by the shaded node $\boldsymbol{w}$.

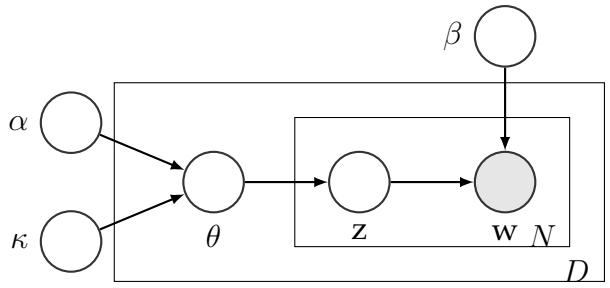

Figure 3: Probabilistic graphical model of LGDA. The shaded circle represent the observed words w while the blank circles represent the topics β, the topic proportion θ, and the topic assignments z.

Given the hyperparameters, the joint distribution for a document of the model is given in equation 74.

$$p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}) = p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) \prod_{n=1}^{N} p(\boldsymbol{z}_n \mid \boldsymbol{\theta}) p(\boldsymbol{w}_n \mid \boldsymbol{z}_n, \boldsymbol{\beta}) \tag{74}$$

We can impose that each word among the documents belongs to a fixed vocabulary of size V. Then, because we assume there are K fixed topics in the corpus, and we are using a GD distribution prior, the word-topic probability matrix β is $K+1 \times V$. Additionally, since we are dealing with probabilities, the topic proportions have to sum up to one $\sum_{k=1}^{K+1} \theta_k = 1$. It is evident that $\boldsymbol{\theta}$ is a different sample for each document, and as a result, each document exhibits a different topic proportion.

The topic assignment dictates which component to select from the topic mixture such that $p(z_n|\boldsymbol{\theta}) = \theta_{z_n}$. Similarly, a word topic probability is selected from β in a manner that

$p(w_n \mid z_n, \boldsymbol{\beta}) = \beta_{z_n,w_n}$. Thus, we rewrite the joint distribution as a sum over the topic assignments z_n, obtaining equation 75.

$$p(\boldsymbol{\theta}, \boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}) = p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) \prod_{n=1}^{N} \sum_{k=1}^{K+1} \theta_k \beta_{k,w_n} \tag{75}$$

Each document has length N yet we can use a fixed vocabulary to represent the words over the collection of documents, and because of the ex-changeability assumption [12], the order of words is not relevant. Thus, the joint for a fixed vocabulary is represented in equation 76.

$$p(\boldsymbol{\theta}, \boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}) = p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) \prod_{w=1}^{V} \left(\sum_{k}^{K+1} \theta_k \beta_{k,w} \right)^{n_w}, \tag{76}$$

where n_w is the number of times that word w appears in the document.

Finally, the marginal probability of a document is obtained by integrating out the mixing topics $\boldsymbol{\theta}$ such that $p(\boldsymbol{w}) = \int p(\boldsymbol{\theta}, \boldsymbol{w}) d\boldsymbol{\theta}$. Now, it is more evident the coupling between $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, which makes the posterior intractable [25]. Thus, in this work, we will make use of EP to approximate the posterior distribution. For instance, the probability of a collection of documents $\boldsymbol{C}$ is shown in equation 77.

$$p(\boldsymbol{C} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}) = \prod_{d=1}^{D} \int p(\boldsymbol{\theta}_d \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) \prod_{w=1}^{V} \left(\sum_{k=1}^{K+1} \theta_{d,k} \beta_{k,w} \right)^{n_{d,w}} d\boldsymbol{\theta}_d \tag{77}$$

4.4.2 Inference

As it is common in any Bayesian setting, the posterior distribution is defined by the hidden variables given the observed words $p(\boldsymbol{\theta}, \mid \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}) \propto p(\boldsymbol{\theta}, \boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta})$. Hence, LGDA's evidence is intractable. Thus, we generate an approximation to $p(\boldsymbol{w})$ using EP since it has been shown that generates more accurate approximations [59, 60]; unlike VI that tends to create biased approximations. Then, EP can provide an estimate for both the posterior and evidence, and sites can be defined as show in equation 78.

$$t_w(\boldsymbol{\theta}) = \sum_{k=1}^{K+1} \theta_k \beta_{k,w} \tag{78}$$

So, the posterior distribution can be factorized as shown in equation 79, where we use a GD distribution as prior.

$$p(\boldsymbol{\theta}, \mid \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}) \propto p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) \prod_{w=1}^{V} t_w(\boldsymbol{\theta})^{n_w} \tag{79}$$

Similar to [59], the approximate sites have a product form (Eq. 80). The parameter ϕ is a matrix $V \times K+1$ and s_w is a normalization constant for the site w.

$$\tilde{t}_w(\boldsymbol{\theta}) = s_w \prod_{k=1}^{K+1} \theta_k^{\phi_{w,k}} \tag{80}$$

By making use of the approximate sites and the GD prior, an approximate posterior distribution can be calculated. Notice that because of conjugacy, we obtain an approximate GD distribution (Eq. 81)

$$q(\boldsymbol{\theta} \mid \boldsymbol{\alpha}', \boldsymbol{\kappa}') \propto p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) \prod_{w=1}^{V} \tilde{t}_w(\boldsymbol{\theta})^{n_w}, \tag{81}$$

where $\gamma'_k = \kappa'_k - \alpha'_{k+1} - \kappa'_{k+1}$ for $k = 1, \ldots, K-1$ and $\gamma'_K = \kappa'_K + \sum_{w=1}^{V} \phi_{w,K+1} n_w - 1$, and its parameters are shown in equations 82 and 83, respectively.

$$\alpha'_k = \alpha_k + \sum_{w=1}^{V} \phi_{w,k} n_w \text{ for } k = 1, \ldots, K \tag{82}$$

$$\kappa'_k = \kappa_k + \sum_{j=k+1}^{K+1} \sum_{w=1}^{V} \phi_{w,j} n_w \text{ for } k = 1, \ldots, K \tag{83}$$

In order to update the approximate site $\tilde{t}_w(\boldsymbol{\theta})$, a cavity distribution is introduced by removing it from the approximate posterior $q^{\setminus w}(\boldsymbol{\theta}) = q(\boldsymbol{\theta}) / \tilde{t}_w(\boldsymbol{\theta})$. We obtain a cavity distribution that is another GD distribution with parameters $\boldsymbol{\alpha}^{\setminus w}$ and $\boldsymbol{\kappa}^{\setminus w}$ shown in

equation 84 and 85, where $\gamma_k^{\setminus w} = \kappa_k^{\setminus w} - \alpha_{k+1}^{\setminus w} - \kappa_{k+1}^{\setminus w}$ for $k = 1, \ldots, K-1$ and $\gamma_K^{\setminus w} = \kappa_K^{\setminus w} + \sum_{w=1}^{V} \phi_{w,K+1} n_w - \phi_{w,K+1} - 1$.

$$\alpha_k^{\setminus w} = \alpha'_k - \phi_{w,k} \text{ for } k = 1, \ldots, K \tag{84}$$

$$\kappa_k^{\setminus w} = \kappa'_k - \sum_{j=k+1}^{K+1} \phi_{w,j} \text{ for } k = 1, \ldots, K \tag{85}$$

Next, the tilted posterior distribution can be obtained by using the site $t_w(\boldsymbol{\theta})$ and the cavity distribution such that

$$q_w^*(\boldsymbol{\theta}) = \frac{1}{z_w} t_w(\boldsymbol{\theta}) q^{\setminus w}(\boldsymbol{\theta}), \tag{86}$$

where the normalization constant $z_w(\boldsymbol{\alpha}^{\setminus w}, \boldsymbol{\kappa}^{\setminus w})$ is shown in equation 87.

$$z_w = \beta_{K+1,w} + \sum_{k=1}^{K} (\beta_{k,w} - \beta_{K+1,w}) \frac{\alpha_k^{\setminus w}}{\alpha_k^{\setminus w} + \kappa_k^{\setminus w}} \prod_{j=1}^{k-1} \frac{\kappa_j^{\setminus w}}{\alpha_j^{\setminus w} + \kappa_j^{\setminus w}} \tag{87}$$

Once found the tilted distribution, we proceed to match the moments with the approximate distribution in order to approximate the current site t_w with the approximate site $\tilde{t}_w$. Since moment matching is equivalent to minimizing the KL divergence, we obtain an optimal distribution $q^{new}(\boldsymbol{\theta})$ with parameters $\boldsymbol{\alpha}^{new}$ and $\boldsymbol{\kappa}^{new}$ that can be obtained from the system of equations shown in equations 88 and 89. The values of the parameters can be obtained with fixed-point iteration method.

$$\Psi(\alpha_k^{new}) - \Psi(\alpha_k^{new} + \kappa_k^{new}) = \frac{1}{z_w} \frac{\partial z_w}{\partial \alpha_k^{\setminus w}} + \Psi(\alpha_k^{\setminus w}) - \Psi(\alpha_k^{\setminus w} + \kappa_k^{\setminus w}) \tag{88}$$

$$\Psi(\kappa_k^{new}) - \Psi(\alpha_k^{new} + \kappa_k^{new}) = \frac{1}{z_w} \frac{\partial z_w}{\partial \kappa_k^{\setminus w}} + \Psi(\kappa_k^{\setminus w}) - \Psi(\alpha_k^{\setminus w} + \kappa_k^{\setminus w}) \tag{89}$$

After matching the moments, the approximate site can be updated using the tilted distribution. In order to accomplish faster convergence and obtain a better representation

of the global approximation, we use damping [30] with a step size μ. Notice when $\mu = 1$, no damping is applied. Hence, the factor updates are expressed in equations 90 and 91.

$$s'_w = z_w \prod_{k=1}^{K} \frac{\Gamma(\alpha_k^{new} + \kappa_k^{new})}{\Gamma(\alpha_k^{new})\Gamma(\kappa_k^{new})} \frac{\Gamma(\alpha_k^{\setminus w})\Gamma(\kappa_k^{\setminus w})}{\Gamma(\alpha_k^{\setminus w} + \kappa_k^{\setminus w})} \tag{90}$$

$$\begin{aligned} \phi'_{w,k} &= \mu(\alpha_k^{new} - \alpha_k^{\setminus w}) + (1-\mu)\phi_{w,k} \\ \phi'_{w,K+1} &= \frac{\mu}{2}\left(\kappa_K^{new} - \kappa_K^{\setminus w} + \phi_{w,K+1} - \sum_w \phi_{w,K+1} n_w\right) + (1-\mu)\phi_{w,K+1} \end{aligned} \tag{91}$$

Finally, we incorporate the contribution of the optimized site in the global approximate distribution $q^*(\boldsymbol{\theta}_d)$ by employing the cavity distribution and the optimal site; the updates are shown in equation 92.

$$\begin{aligned} \alpha_k'^{new} &= \alpha'_k + n_w(\phi'_{w,k} - \phi_{w,k}) \\ \kappa_k'^{new} &= \kappa'_k + n_w(\sum_{j=k+1}^{K+1} \phi'_{w,j} - \phi_{w,j}) \end{aligned} \tag{92}$$

After convergence, we can compute $p(\boldsymbol{w})$ as follows:

$$z = \prod_{k=1}^{K} \frac{\Gamma(\alpha_k + \kappa_k)}{\Gamma(\alpha_k)\Gamma(\kappa_k)} \frac{\Gamma(\alpha'_k)\Gamma(\kappa'_k)}{\Gamma(\alpha'_k + \kappa'_k)} \times \prod_{w=1}^{V} s_w^{n_w} \tag{93}$$

The full learning algorithm for inference is depicted in Algorithm 3.

Algorithm 3: LGDA inference algorithm with EP. We use an step size of n_{dw}.

Initialize approximate factors $\tilde{t}_w = 1$, where $\phi_{w,k} = 0$ and $s_w = 1$. This is the same as initializing approximate parameters with priors $\alpha'_k = \alpha_k$ and $\kappa'_k = \kappa_k$;
for *doc in Corpus* **do**
  Compute $q(\boldsymbol{\theta})$ by calculating α' and κ';
  while *not convergence* **do**
    for *word in doc* **do**
      Delete: compute cavity distribution $q^{\setminus w}(\boldsymbol{\theta} \mid \boldsymbol{\alpha}^{\setminus w}, \boldsymbol{\kappa}^{\setminus w})$.;
      if $\boldsymbol{\alpha}^{\setminus w} < 0$ *or* $\boldsymbol{\kappa}^{\setminus w} < 0$ **then**
        Ignore *word* in this iteration and undo changes.;
      end
      Match moments: match the moments of $q^*(\boldsymbol{\theta})$ and $q^{new}(\boldsymbol{\theta})$ by minimizing $D_{KL}(q^*(\boldsymbol{\theta}) \parallel q^{new}(\boldsymbol{\theta}))$;
      Update: get parameters of $\tilde{t}_w$ by calculating $\phi'_{w,k}$ and s'_w;
      Incorporate: introduce the optimized site into the global approximation $q^(\boldsymbol{\theta} \mid \boldsymbol{\alpha}', \boldsymbol{\kappa}')$;
      if $\boldsymbol{\alpha}' < 0$ *or* $\boldsymbol{\kappa}' < 0$ **then**
        Ignore *word* in this iteration and undo changes.;
      end
    end
  end
end

4.4.3 Parameter Estimation

Finally, we obtain estimates of the model parameters by maximizing the ELBO with respect to α, κ, and β. Thus, we can write the ELBO as shown in equation 94.

$$\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}) = \sum_{d=1}^{D} \mathbb{E}_q \left[\log p(\boldsymbol{\theta}_d)\right] + \sum_{d=1}^{D} \mathbb{E}_q \left[\sum_{w=1}^{V} n_{d,w} \log(\sum_{k=1}^{K+1} \theta_{d,k}\beta_{k,w})\right] + C \tag{94}$$

Maximizing this expression with respect to α_k and κ_k lead us to the following system of equations (eq. 95, which has no closed-form and can be approximated using Newton's method [57].

$$D\left[\Psi(\alpha_k + \kappa_k) - \Psi(\alpha_k)\right] = \sum_{d}^{D} \left[-\Psi(\alpha'_{d,k}) + \Psi(\alpha'_{d,k} + \kappa'_{d,k})\right]$$
$$D\left[\Psi(\alpha_k + \kappa_k) - \Psi(\kappa_k)\right] = \sum_{d}^{D} \left[-\Psi(\kappa'_{d,k}) + \Psi(\alpha'_{d,k} + \kappa'_{d,k})\right] \tag{95}$$

Next, we find the optimal topics by maximizing the ELBO w.r.t. $\beta_{k,w}$ (see eq 96) where we find an expectation that can be approximated using second-order Taylor expansion about $\mathbb{E}\left[\boldsymbol{\theta}_d\right]$.

$$\beta_{k,w} \propto \sum_{d}^{D} n_{d,w} \mathbb{E}_q \left[\frac{\theta_{d,k}\beta_{k,w}}{\sum_{k=1}^{K+1} \theta_{d,k}\beta_{k,w}} \right] \tag{96}$$

4.5 Results

In this section, we test convergence by comparing the lower bounds and evaluate the LGDA model on a text classification task in order to evaluate the predictive performance due that correlation can lead to better predictive distributions.

Dataset We use the Reuters-21578[1] corpus which is a collection of labeled newswire articles. The dataset consists of $21,578$ documents, including documents without topics and typographical errors. Thus, we use the top-6 categories following the experiment performed by [1], resulting in approximately $9,000$ documents. Table 5 summarizes the selected categories and number of documents per class. We preprocess the selected corpus by lowercasing words and removing punctuation. Next, words in third person are changed to first person and tenses are changed to present by using a standard lemmatizer. Stop words and words with less than three characters are filtered. Finally, we use a stemmer to reduce all the remaining words to its root form and tokenize to form the vocabulary.

[1] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

Table 5: Classes and number of documents extracted from Reuters dataset

category	num. docs
acq	2369
crude	578
earn	3964
grain	582
interest	478
money-fx	717

Models We compare the performance of LGDA-Expectation Propagation with LDA since it is the most commonly used topic model and has not only similar conjugacy properties but also a similar generative process. We use an implementation of LDA with variational Bayes inference[2].

Experiment description As noticed by [1], LGDA has a similar predictive power as LDA yet LGDA is better at discriminating related categories due that topics are correlated. Thus, we use train/test splits as specified in [1] and build two classifiers, a supervised LASSO regression with a Multinomial and Bernoulli distribution for multiclass and binary classification. We use the full dataset for the multiclass classifier which has a vocabulary size of $V = 10,123$ words, and similarly for the binary classifier, we use two related categories (i.e. *interest* and *money-fx*) resulting in a vocabulary size of $V = 4,233$ words. We use the number of topics K reported in [1].

Topic Interpretability We train LGDA-EP and LDA and evaluate the lower bounds using the full dataset with $K = 15$ and $K = 30$ topics as shown in figure 4. For EP, we initialize the approximate factors $\tilde{t}_w = 1$, and for LDA-VI, we initialize the variational parameters randomly. We can notice that LGDA-EP not only converged considerably faster but also reaches a better solution by looking at the approximate evidence.

We next look at the learned topics. Table 6 displays the 4 most used topics for LDA-EP, as given by the average of the topic proportions $\boldsymbol{\theta}_d$. LDA provide interpretable topics.

[2]We use an implementation of LDA where no smoothing is applied [11].

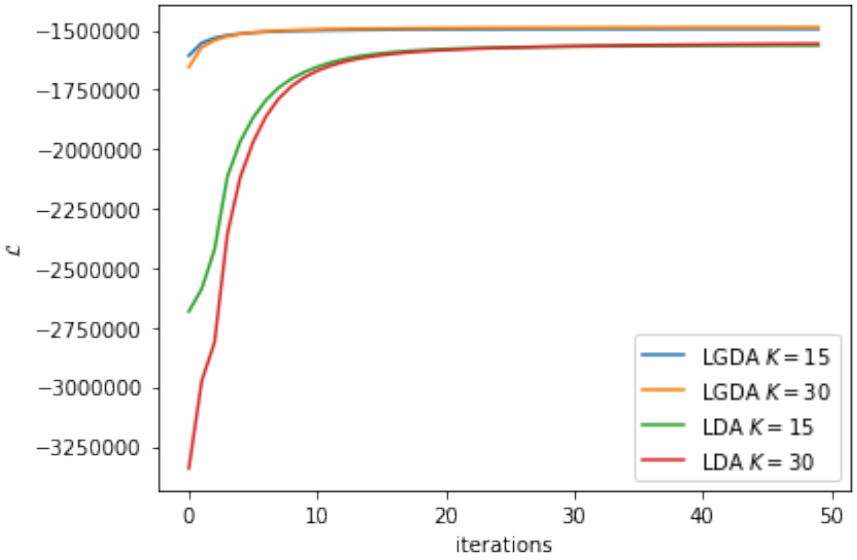

Figure 4: Comparison of LGDA-EP and LDA in terms of evidence lower bound for $K = 15$ and $K = 30$ topics.

Table 6: Top five words on the full dataset with vocabulary size $10,123$ and $K = 15$ topics.

LGDA-EP Topics			
bank	dlrs	stock	say
market	billion	record	share
say	loss	april	company
billion	profit	dividend	dlrs
money	year	prior	offer

Topic Classification We evaluate the predictive power of LDA-EP and compare the obtained results with LDA using variational Bayes inference (LGDA-VI) [1] and LDA [11]. We evaluate the models' performance in terms of accuracy. First, we build a binary classifier in order to evaluate the ability of LDA to discriminate similar categories. We select the optimal number of topics as proposed by [1]. Table 7 illustrates the results of binary classification for the categories *money-fx* and *interest*. As expected LDA is slightly better at discriminating similar categories obtaining 71% of accuracy.

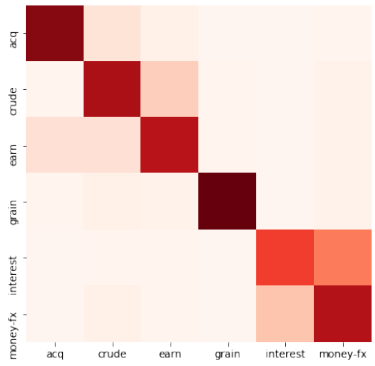

Figure 5: LDA $K = 15$ topics

Consequently, we build a classifier using the full-dataset, and as expected LDA-EP[3] provides similar or better predictive performance than the vanilla LDA as shown in Table 7. Figure 5 and 6 illustrate the confusion matrix for both LDA and LDA with $K = 15$ topics. It is noticeable that LDA is better not only at discriminating distinct categories but also similar categories which accounts for the accuracy's jump.

Table 7: Results for binary classification with K=15 and multi-class classification with K=15 and K=30. Comparison using accuracy. VI: variational inference model; EP: expectation propagation.

Models	Accuracy		
	money-fx vs. interest	*all classes*	
	K=15	K=15	K=30
LDA	69%	81%	78.8%
LGDA-VI [1]	70%	64.9%	64.8%
LGDA-EP	**71%**	**84%**	**78.9%**

4.6 Conclusions

In this chapter, we propose the use of Expectation Propagation (EP) for the Latent Generalized Dirichlet allocation model to learn a mixture of latent topics over documents and

[3]Results with LDA-VI differ due to the pre-processing or hyperparameter configuration.

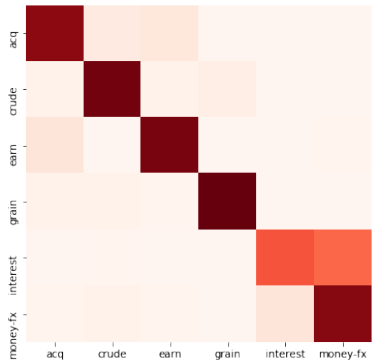

Figure 6: LGDA $K = 15$ topics

a vocabulary while maintaining topic correlation. We make use of EP in order to have accurate approximations since as opposed to variational inference, EP doesn't need to be bounded to create an approximation to the posterior. We additionally develop a method for parameter estimation. We evaluate topic interpretability by looking at the resulting topics and the predictive power of LDA-EP showing the efficacy of the proposed method and showing superior results to the traditional LDA.

Chapter 5

Conclusions and Future Directions

This thesis focuses in learning efficiently mixture models employing message passing when dealing with count data, more specifically we use Expectation Propagation (EP) and Stochastic Expectation Propagation (SEP) to learn the parameters of the model and a latent variable model. We use the Exponential approximation to the Dirichlet Compound Multinomial (EDCM) distribution and Exponential approximation to the Multinomial Scaled Dirichlet (EMSD) distribution to model word appearance. Additionally, we use the Generalized Dirichlet (GD) distribution to model correlation between topics.

We show how to use effectively EP to learn a finite EDCM mixture model that shows comparable results with other inference methods. Consequently, we learn a finite EMSD mixture using SEP that performs comparably to EP but requires fewer parameters to be saved, and thus, being faster and reducing memory consumption. We, finally, use EP for the Latent Generalized Dirichlet allocation model to learn a mixture of latent topics over documents while maintaining topic correlation and show that the learned topics can be used as feature inputs for downstream machine learning tasks.

The proposed models can be extended to feature selection by weighting discrete features, similar to [19] and using model selection methods (e.g. Bayesian Information Criterion [70]) to choose the appropriate model. Additionally, there is the problem of knowing the number of topics beforehand, selecting the right number of clusters can be challenging

depending on the application. A non-parametric Bayesian mixture model could alleviate this complication. Thus, applying an infinite mixture model would not allow to detect the appropriate number of clusters only but also find relevant features (e.g. [14]). These models could be extended to a supervised settings.

EP and SEP depend on moment matching which in some cases is intractable. In this work, we attempted different approaches to match the moments such as Laplace Method, Black Box variational inference and sampling methods. We found sampling to be the most stable but future directions could be devoted to compute the moments effectively (e.g. [21, 84]). Next, SEP is a new inference method that saves memory consumption and performs similar to EP. Here, we use SEP for estimating parameters but would be interesting to see the performance of SEP when combined with latent variable models and as well how well SEP performs when doing mini-batching.

Acronyms

ADF Assumned Density Filtering. 4, 5

DCM Dirichlet Compound Multinomial. 10, 11, 13, 24, 25, 39

EDCM Exponential-family Approximation to DCM. 7–11, 13–16, 19, 20, 22–29, 31–33, 37, 39, 41

EM Expectation Maximization. 10

EMSD Exponential-family Approximation to MSD. 7, 8, 39

EP Expectation Propagation. 4–9, 25, 42, 44, 45, 49, 53, 55, 58

IS Importance Sampling. 3

LDA Latent Dirichlet Allocation. 10, 11, 22, 44–48, 55–58

LGDA Latent Generalized Dirichlet Allocation. 7, 42, 45, 47, 49, 54, 55

MC Monte Carlo. 2

MCMC Markov Chain Monte Carlo. 3, 4, 28, 44

MSD Multinomial Scaled Dirichlet. 39

SEP Stochastic Expectation Propagation. 5–8, 28

VI Variational Inference. 4, 28, 44

Bibliography

[1] Ali Shojaee Bakhtiari and Nizar Bouguila. A variational bayes model for count data learning and classification. *Engineering Applications of Artificial Intelligence*, 35:176–186, 2014.

[2] Rohan A Baxter and Jonathan J Oliver. Finding overlapping components with mml. *Statistics and Computing*, 10(1):5–16, 2000.

[3] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4):509–522, 2002.

[4] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[5] David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April 2012.

[6] David M Blei, Thomas L Griffiths, and Michael I Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7, 2010.

[7] David M Blei and Michael I Jordan. Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134. ACM, 2003.

[8] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[9] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[10] David M Blei, John D Lafferty, et al. A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35, 2007.

[11] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. In *Advances in neural information processing systems*, pages 601–608, 2002.

[12] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[13] Nizar Bouguila. Clustering of count data using generalized dirichlet multinomial distributions. *IEEE Transactions on Knowledge and Data Engineering*, 20(4):462–474, 2008.

[14] Nizar Bouguila. A model-based approach for discrete data clustering and feature weighting using MAP and stochastic complexity. *IEEE Trans. Knowl. Data Eng.*, 21(12):1649–1664, 2009.

[15] Nizar Bouguila. Count data modeling and classification using finite mixtures of distributions. *IEEE Transactions on Neural Networks*, 22(2):186–198, 2010.

[16] Nizar Bouguila and Djemel Ziou. Unsupervised learning of a finite discrete mixture model based on the multinomial dirichlet distribution: Application to texture modeling. In Ana L. N. Fred, editor, *Pattern Recognition in Information Systems, Proceedings of the 4th International Workshop on Pattern Recognition in Information Systems, PRIS 2004, In conjunction with ICEIS 2004, Porto, Portugal, April 2004*, pages 118–127. INSTICC Press, 2004.

[17] Nizar Bouguila and Djemel Ziou. Unsupervised selection of a finite dirichlet mixture model: an mml-based approach. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):993–1009, 2006.

[18] Nizar Bouguila and Djemel Ziou. Unsupervised learning of a finite discrete mixture: Applications to texture modeling and image databases summarization. *Journal of Visual Communication and Image Representation*, 18(4):295–309, 2007.

[19] Nizar Bouguila and Djemel Ziou. A countably infinite mixture model for clustering and feature selection. *Knowl. Inf. Syst.*, 33(2):351–370, 2012.

[20] Jordan Boyd-Graber, Yuening Hu, David Mimno, et al. Applications of topic models. *Foundations and Trends® in Information Retrieval*, 11(2-3):143–296, 2017.

[21] Thang D Bui, José Miguel Hernández-Lobato, Yingzhen Li, Daniel Hernández-Lobato, and Richard E Turner. Training deep gaussian processes using stochastic expectation propagation and probabilistic backpropagation. *arXiv preprint arXiv:1511.03405*, 2015.

[22] Karla L Caballero, Joel Barajas, and Ram Akella. The generalized dirichlet distribution in enhanced topic detection. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 773–782. ACM, 2012.

[23] Robert J Connor and James E Mosimann. Concepts of independence for proportions with a generalization of the dirichlet distribution. *Journal of the American Statistical Association*, 64(325):194–206, 1969.

[24] Gabriela Csurka and Florent Perronnin. Fisher vectors: Beyond bag-of-visual-words image representations. In *International Conference on Computer Vision, Imaging and Computer Graphics*, pages 28–42. Springer, 2010.

[25] James M Dickey. Multiple hypergeometric functions: Probabilistic interpretations and statistical uses. *Journal of the American Statistical Association*, 78(383):628–637, 1983.

[26] Charles Elkan. Clustering documents with an exponential-family approximation of the dirichlet compound multinomial distribution. In *Proceedings of the 23rd international conference on Machine learning*, pages 289–296. ACM, 2006.

[27] Wentao Fan and Nizar Bouguila. Non-gaussian data clustering via expectation propagation learning of finite dirichlet mixture models and applications. *Neural processing letters*, 39(2):115–135, 2014.

[28] Wentao Fan and Nizar Bouguila. Expectation propagation learning of a dirichlet process mixture of beta-liouville distributions for proportional data clustering. *Engineering Applications of Artificial Intelligence*, 43:1–14, 2015.

[29] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531. IEEE, 2005.

[30] Andrew Gelman, Aki Vehtari, Pasi Jylänki, Christian Robert, Nicolas Chopin, and John P Cunningham. Expectation propagation as a way of life. *arXiv preprint arXiv:1412.4869*, 157, 2014.

[31] Andrew Gelman, Aki Vehtari, Pasi Jylänki, Tuomas Sivula, Dustin Tran, Swupnil Sahai, Paul Blomstedt, John P Cunningham, David Schiminovich, and Christian Robert. Expectation propagation as a way of life: A framework for bayesian inference on partitioned data. *arXiv preprint arXiv:1412.4869*, 2017.

[32] Sean Gerrish and David M Blei. A language-based approach to measuring scholarly impact. In *ICML*, volume 10, pages 375–382. Citeseer, 2010.

[33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[34] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[35] Thomas L Griffiths, Mark Steyvers, David M Blei, and Joshua B Tenenbaum. Integrating topics and syntax. In *Advances in neural information processing systems*, pages 537–544, 2005.

[36] Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864, 2010.

[37] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[38] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[39] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196, 2001.

[40] Koffi Eddy Ihou and Nizar Bouguila. Variational-based latent generalized dirichlet allocation model in the collapsed space and applications. *Neurocomputing*, 332:372–395, 2019.

[41] Joshua Johnston and Greg Hamerly. Improving simpoint accuracy for small simulation budgets with edcm clustering. *Worksh. on Statistical and Machine learning approaches to ARchitectures and compilaTion (SMART08)*, 2008.

[42] Weonyoung Joo, Wonsung Lee, Sungrae Park, , and Il-Chul Moon. Dirichlet variational autoencoder, 2019.

[43] Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[44] Slava M Katz. Distribution of content words and phrases in text and language modelling. *Natural language engineering*, 2(1):15–59, 1996.

[45] Simon Lacoste-Julien, Fei Sha, and Michael I Jordan. Disclda: Discriminative learning for dimensionality reduction and classification. In *Advances in neural information processing systems*, pages 897–904, 2009.

[46] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[47] Li-Jia Li, Chong Wang, Yongwhan Lim, David M Blei, and Li Fei-Fei. Building and using a semantivisual image hierarchy. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3336–3343. IEEE, 2010.

[48] Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. Stochastic expectation propagation. In *Advances in neural information processing systems*, pages 2323–2331, 2015.

[49] Yue Li and Manolis Kellis. A latent topic model for mining heterogenous non-randomly missing electronic health records data. *arXiv preprint arXiv:1811.00464*, 2018.

[50] Robert H Lochner. A generalized dirichlet distribution in bayesian life testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 37(1):103–113, 1975.

[51] Zhanyu Ma and Arne Leijon. Expectation propagation for estimating the parameters of the beta distribution. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2082–2085. IEEE, 2010.

[52] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

[53] Rasmus E Madsen, David Kauchak, and Charles Elkan. Modeling word burstiness using the dirichlet distribution. In *Proceedings of the 22nd international conference on Machine learning*, pages 545–552. ACM, 2005.

[54] Dimitris Margaritis and Sebastian Thrun. A bayesian multiresolution independence test for continuous variables. *arXiv preprint arXiv:1301.2292*, 2013.

[55] Jon D Mcauliffe and David M Blei. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128, 2008.

[56] Geoffrey J McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

[57] Thomas Minka. Estimating a dirichlet distribution, 2000.

[58] Thomas Minka. Power ep. Technical report, Technical report, Microsoft Research, Cambridge, 2004.

[59] Thomas Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 352–359. Morgan Kaufmann Publishers Inc., 2002.

[60] Thomas P Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

[61] Thomas Peter Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

[62] Fatma Najar, Nuha Zamzami, and Nizar Bouguila. Fake news detection using bayesian inference. In *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*, pages 389–394. IEEE, 2019.

[63] Radford M Neal. Probabilistic inference using markov chain monte carlo methods. 1993.

[64] Manfred Opper and Ole Winther. A bayesian approach to on-line learning. *On-line learning in neural networks*, pages 363–378, 1998.

[65] Vera Pawlowsky-Glahn and Antonella Buccianti. *Compositional data analysis*. Wiley Online Library, 2011.

[66] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[67] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

[68] Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226, 2015.

[69] Alexandra Schofield, Måns Magnusson, and D Mimno. Understanding text preprocessing for latent dirichlet allocation. In *Proceedings of the 15th conference of the European chapter of the Association for Computational Linguistics*, volume 2, pages 432–436, 2017.

[70] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[71] Liangcai Shu, Bo Long, and Weiyi Meng. A latent topic model for complete entity resolution. In *Proceedings of the 2009 IEEE International Conference on Data Engineering*, pages 880–891. IEEE Computer Society, 2009.

[72] Josef Sivic, Bryan C Russell, Andrew Zisserman, William T Freeman, and Alexei A Efros. Unsupervised discovery of visual object class hierarchies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[73] Richard Socher, Samuel Gershman, Per Sederberg, Kenneth Norman, Adler J Perotte, and David M Blei. A bayesian analysis of dynamics in free recall. In *Advances in neural information processing systems*, pages 1714–1722, 2009.

[74] Oskar Söderkvist. Computer vision classification of leaves from swedish trees, 2001.

[75] Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

[76] Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[77] Xavier Sumba and Nizar Bouguila. Improving classification using topic correlation and expectation propagation. In *Lecture Notes in Artificial Intelligence*. Canada AI, Accepted, 2020.

[78] Xavier Sumba, Freddy Sumba, Andres Tello, Fernando Baculima, Mauricio Espinoza, and Víctor Saquicela. Detecting similar areas of knowledge using semantic and data mining technologies. *Electronic Notes in Theoretical Computer Science*, 329:149–167, 2016.

[79] Xavier Sumba, Nuha Zamzami, and Nizar Bouguila. Clustering count data with stochastic expectation propagation. In *Pattern Recognition Journal*. Submitted, 2020.

[80] Xavier Sumba, Nuha Zamzami, and Nizar Bouguila. Improving the edcm mixture model with expectation propagation. In *2020 Association for the Advancement of Artificial Intelligence AAAI*. FLAIRS 33, Accepted, 2020.

[81] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[82] Osman Tursun and Sinan Kalkan. METU dataset: A big dataset for benchmarking trademark retrieval. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 514–517. IEEE, 2015.

[83] Martin J Wainwright and Michael I Jordan. Introduction to variational methods for graphical models. *Foundations and Trends in Machine Learning*, 1:1–103, 2008.

[84] Zheng Wang and Shandian Zhe. Conditional expectation propagation. *arXiv preprint arXiv:1910.12360*, 2019.

[85] Tzu-Tsung Wong. Generalized dirichlet distribution in bayesian analysis. *Applied Mathematics and Computation*, 97(2-3):165–181, 1998.

[86] Tzu-Tsung Wong. Alternative prior assumptions for improving the performance of naïve bayesian classifiers. *Data Mining and Knowledge Discovery*, 18(2):183–213, 2009.

[87] Nuha Zamzami and Nizar Bouguila. MML-based approach for determining the number of topics in EDCM mixture models. In *Canadian Conference on Artificial Intelligence*, pages 211–217. Springer, 2018.

[88] Nuha Zamzami and Nizar Bouguila. Text modeling using multinomial scaled dirichlet distributions. In *Mouhoub M., Sadaoui S., Ait Mohamed O., Ali M. (eds) Recent Trends and Future Technology in Applied Intelligence. IEA/AIE 2018. Lecture Notes in Computer Science, vol 10868*, pages 69–80. Springer, 2018.

[89] Nuha Zamzami and Nizar Bouguila. An accurate evaluation of msd log-likelihood and its application in human action recognition. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5. IEEE, 2019.

[90] Nuha Zamzami and Nizar Bouguila. Hybrid generative discriminative approaches based on multinomial scaled dirichlet mixture models. *Applied Intelligence*, 49(11):3783–3800, 2019.

[91] Nuha Zamzami and Nizar Bouguila. Model selection and application to high-dimensional count data clustering: via finite EDCM mixture models. *Applied Intelligence*, 49(4):1467–1488, 2019.

[92] Nuha Zamzami and Nizar Bouguila. A novel scaled dirichlet-based statistical framework for count data modeling: Unsupervised learning and exponential approximation. *Pattern Recognition*, 2019.

[93] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.