IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 15, 2020, accepted June 22, 2020, date of publication July 10, 2020, date of current version July 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3008534

MAP Estimation With Bernoulli Randomness, and Its Application to Text Analysis and Recommender Systems

XUAN BUI[1,2], HIEU VU[3], OANH NGUYEN[2], AND KHOAT THAN[2,3]

[1]Faculty of Computer Science and Engineering, Thuyloi University, Hanoi 100000, Vietnam
[2]School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi 100000, Vietnam
[3]VinAI Research, Hanoi 100000, Vietnam

Corresponding author: Xuan Bui (xuanbtt@tlu.edu.vn)

This work was supported by the Vingroup Innovation Foundation (VINIF) under Project VINIF.2019.DA18.

ABSTRACT MAP estimation plays an important role in many probabilistic models. However, in many cases, the MAP problem is non-convex and intractable. In this work, we propose a novel algorithm, called BOPE, which uses Bernoulli randomness for Online Maximum a Posteriori Estimation. We show that BOPE has a fast convergence rate. In particular, BOPE implicitly employs a prior which plays as regularization. Such a prior is different from the one of the MAP problem and will be vanishing as BOPE does more iterations. This property of BOPE is significant and enables to reduce severe overfitting for probabilistic models in ill-posed cases, including short text, sparse data, and noisy data. We validate the practical efficiency of BOPE in two contexts: text analysis and recommender systems. Both contexts show the superior of BOPE over the baselines.

INDEX TERMS Bernoulli randomness, online MAP estimation, probabilistic models.

I. INTRODUCTION

Maximum a Posteriori (MAP) estimation is a popular approach to inference in probabilistic models [1], [2]. It plays an essential role in various practical scenarios where there exist hidden variables or uncertainty. Some applications include image processing [3], [4], text analysis [5]–[7], recommender system [8], protein design and protein side-chain prediction problems [9], [10]. Adding the prior probability information reduces the overdependence on the observed data for parameter estimation, MAP estimation be seen as a regularization of Maximum Likelihood Estimation (MLE), MAP can deal well with low training data. In MAP estimation, our task is to find

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\Omega} P(\boldsymbol{x}|D) \tag{1}$$

where D denotes the observed data, $\boldsymbol{x}$ denotes a hidden/unobserved variable, and Ω denotes the domain of $\boldsymbol{x}$. Using Bayes' theorem, we have

$$P(\boldsymbol{x}|D) = \frac{P(D|\boldsymbol{x})P(\boldsymbol{x})}{P(D)} \propto P(D|\boldsymbol{x})P(\boldsymbol{x}) \tag{2}$$

where $P(D|\boldsymbol{x})$ denotes the likelihood of D, $P(\boldsymbol{x})$ denotes the prior of $\boldsymbol{x}$, and $P(D)$ denotes the marginal probability of D. Using (2), we can rewrite (1) as

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\Omega}[f(\boldsymbol{x}) = \log P(D|\boldsymbol{x}) + \log P(\boldsymbol{x})] \tag{3}$$

In general, the MAP problem is well-known to be NP-hard [11], [12], and is at least as computationally difficult as MLE [13]. Hence, there is a large number of studies about approximate methods [4], [9], [10], [14]–[20]. It is interesting that most existing studies focus on discrete MAP, i.e., when domain Ω is discrete.

In this work, we focus on the MAP problem which is continuous and non-convex, i.e., when the function $-f(\boldsymbol{x}) = -\log P(D|\boldsymbol{x}) - \log P(\boldsymbol{x})$ is non-convex over the continuous domain Ω. In contrast with convex cases which are tractable [4], the non-convex MAP problem is NP-hard and hence intractable [11]. A popular choice is to employ some recent advances in non-convex optimization such as Concave-Convex procedure (CCCP) [21], Stochastic Majorization-Minimization (SMM) [22], Frank-Wolfe (FW) [23], Online Frank-Wolfe (OFW) [24], [25], Natasha2 [26], NEON2 [27] to solve the MAP estimation as a non-convex optimization problem. However, the non-convex

The associate editor coordinating the review of this manuscript and approving it for publication was Tomasz Trzcinski.

optimization is NP-hard [25], [28], [29] and those methods for general non-convex problems may not provide good solutions since they ignore the special structure of the MAP problem in probabilistic models. Many approaches to estimating a full posterior distribution have been studied, e.g., Variational Bayes (VB) [30], collapsed Variational Bayes (CVB) [31], Collapsed Gibbs Sampling (CGS) [32], Particle Mirror Decent (PMD) [33], Hessian Approximated Markov Chain Monte Carlo (HAMCMC) [34]. However, those approaches do not solve directly the MAP problem, thus, provide suboptimal solutions, and have a slow convergence rate.

Our contributions are as follows:

- We introduce a novel efficient algorithm, namely Bernoulli randomness in Online maximum a Posteriori Estimation (BOPE), for solving the non-convex MAP problem via using Bernoulli sampling and stochastic bounds. BOPE is stochastic in nature and converges to a stationary point of the MAP problem at a rate of $\mathcal{O}(1/T)$ which is the state-of-the-art convergence rate, where T denotes the number of iterations. BOPE can be readily employed in a wide range of contexts.
- In particular, we prove that the Bernoulli randomness in BOPE plays the regularization role. BOPE implicitly employs a prior which will be stochastically vanishing w.r.t T and is entirely different from the prior of the MAP problem. This regularization role will be crucial to prevent overfitting in probabilistic models when working with the challenges of extremely sparse data and noisy data [7], [35], [36]. Existing inference methods do not have this property.
- We investigate the practical effectiveness of BOPE in two applications: text analysis and recommender system. Extensive experiments show the superior of BOPE to the state-of-the-art inference methods. In particular, for short text, BOPE often performs significantly better than the other baselines, owing to its regularization ability.

Organization: This paper is organized as follows: In Section II, we present the background of MAP problem. We present BOPE to solve effectively the MAP problem via Bernoulli randomness and two stochastic bounds in Section III. Next, in Section IV, we apply BOPE to text analysis. In Section V, we show the application of BOPE in recommender systems. Finally, we make the conclusion in Section VI.

II. RELATED WORK

In Bayesian inference, MAP estimation provides the mode of the posterior distribution of interest. The MAP estimation can be used to obtain a point estimate of an unobserved quantity on the basis of empirical data. Problem (3) will be non-convex in cases that the likelihood or prior is not concave.[1]

Since our focus in this paper is the non-convex MAP problem (3), we will survey some closely related literature and recent developments. From (3), we can consider MAP estimation as an optimization problem. In some cases, the MAP problem (3) is a convex problem [4], then it can be solved well. In addition, in probabilistic models, we usually study the MAP problem in high dimensions. Therefore, the difficulty of the MAP problem depends on the objective function $f(\boldsymbol{x}) = \log P(D|\boldsymbol{x}) + \log P(\boldsymbol{x})$. If the densities of distribution over $\boldsymbol{x}$ and D can be described by some analytic function, then the MAP estimation problem turns out to be the maximization of the objective function $f(\boldsymbol{x}) = g_1(\boldsymbol{x}) + g_2(\boldsymbol{x})$ where $g_1(\boldsymbol{x}) = \log P(D|\boldsymbol{x})$ and $g_2(\boldsymbol{x}) = \log P(\boldsymbol{x})$. Thus, we can formulate the problem (3) as a non-convex constrained optimization problem as follows

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\Omega}[f(\boldsymbol{x}) = g_1(\boldsymbol{x}) + g_2(\boldsymbol{x})] \tag{4}$$

Solving (4) is harder when the objective function $f(\boldsymbol{x})$ is non-convex because non-convex optimization problems usually admit a multimodal structure. Then, common optimization methods such as gradient descent or Newton method may be trapped in poor local optima [37], [38]. In this work, we are interested in solving the MAP problem (4) in cases that $f(\boldsymbol{x})$ is smooth and non-convex. Under that condition, problem (4) is NP-hard in general [11], [39], [40].

There are some inference methods such as VB, CVB, CVB0, CGS, CCCP, SMM, PMD, HAMCMC, and OPE [39] for solving the MAP problem in probabilistic graphical models. They can be considered as the state-of-the-art inference methods, yet there still are many drawbacks when going into details. To our knowledge, we have not seen any theoretical analysis about how fast VB, CVB, CVB0, and CGS do inference. While CCCP and SMM are guaranteed to converge to a stationary point of the inference problem, the convergence rate of both methods is unknown for non-convex MAP problems. With theoretical proofs, PMD and HAMCMC, which are both based on sampling to estimate a posterior distribution, converge at a rate of $\mathcal{O}(T^{-1/2})$ and $\mathcal{O}(T^{-1/3})$, respectively. We find out that those rates of convergence are relatively slow compared with OPE. OPE is an online version of the OFW and has a convergence rate of $\mathcal{O}(1/T)$, where T denotes the number of iterations. Details of OPE is presented in Algorithm 1.

Each iteration of OPE requires us to solve a linear program that is significantly easier than a non-linear problem. Instead of directly solving the MAP problem (4) with the true objective function $f(\boldsymbol{x})$, OPE constructs a sequence of stochastic functions $F_t(\boldsymbol{x})$ that approximates to $f(\boldsymbol{x})$ by alternatively picking uniformly randomly an f_t from $\{g_1(\boldsymbol{x}),\ g_2(\boldsymbol{x})\}$ at each iteration t. It is guaranteed that F_t converges to f as $t \to \infty$ and then MAP problem (4) becomes easily to solve. Although OPE is easy to implement and has a fast convergence, it remains some limitations. When inferring the hidden variable $\boldsymbol{x}$, we use either likelihood or knowledge we have known before (prior). This behavior is very natural.

[1] For example, the following distributions will result in non-convex problems: Beta, Gamma, Weibull, Log logistic, Logit normal, Levy, Generalized pareto, Erlang, F, Kent, Dagum, Gompertz.

Algorithm 1 OPE: A General Framework for MAP Estimation

```
Output: $\boldsymbol{x}^*$ that maximizes the objective function $f(\boldsymbol{x}) = g_1(\boldsymbol{x}) + g_2(\boldsymbol{x})$ over the compact domain $\Omega$.
Initialize $\boldsymbol{x}_1$ arbitrary in $\Omega$
for $t = 1, 2, \ldots, \infty$ do
    Pick $f_t$ uniformly from $\{g_1(\boldsymbol{x}), g_2(\boldsymbol{x})\}$
    $F_t := \frac{2}{t}\sum_{h=1}^{t} f_h$
    $\boldsymbol{a}_t := \arg\max_{\boldsymbol{x}\in\Omega}\langle F_t'(\boldsymbol{x}_t), \boldsymbol{x}\rangle$
    $\boldsymbol{x}_{t+1} := \boldsymbol{x}_t + \frac{\boldsymbol{a}_t - \boldsymbol{x}_t}{t}$
end for
```

Algorithm 2 BOPE for Solving the MAP Problem

```
Input: Bernoulli parameter $p \in (0, 1)$
Output: $\boldsymbol{x}^*$ that maximizes $f(\boldsymbol{x}) = \log P(D|\boldsymbol{x}) + \log P(\boldsymbol{x})$ over the compact domain $\Omega$.
Initialize $\boldsymbol{x}_1$ arbitrary in $\Omega$
$G_1(\boldsymbol{x}) := \frac{1}{p}\log P(D|\boldsymbol{x})$; $G_2(\boldsymbol{x}) := \frac{1}{1-p}\log P(\boldsymbol{x})$
$f_1^l := G_1(\boldsymbol{x}), f_1^u := G_2(\boldsymbol{x})$
for $t = 1, 2, \ldots, T$ do
    Pick $f_{t+1}^l$ randomly from $\{G_1(\boldsymbol{x}), G_2(\boldsymbol{x})\}$ according to
    the Bernoulli distribution with parameter $p$, where
    $P(f_{t+1}^l = G_1(\boldsymbol{x})) = p$; $P(f_{t+1}^l = G_2(\boldsymbol{x})) = 1 - p$
    $L_t := \frac{1}{t}\sum_{h=1}^{t} f_h^l$
    $\boldsymbol{a}_t^l := \arg\max_{\boldsymbol{x}\in\Omega} < L_t'(\boldsymbol{x}_t), \boldsymbol{x} >$
    $\boldsymbol{x}_{t+1}^l := \boldsymbol{x}_t + \frac{\boldsymbol{a}_t^l - \boldsymbol{x}_t}{t}$
    Pick $f_{t+1}^u$ randomly from $\{G_1(\boldsymbol{x}), G_2(\boldsymbol{x})\}$ according to
    the Bernoulli distribution with parameter $p$, where
    $P(f_{t+1}^u = G_1(\boldsymbol{x})) = p$; $P(f_{t+1}^u = G_2(\boldsymbol{x})) = 1 - p$
    $U_t := \frac{1}{t}\sum_{h=1}^{t} f_h^u$
    $\boldsymbol{a}_t^u := \arg\max_{\boldsymbol{x}\in\Omega} < U_t'(\boldsymbol{x}_t), \boldsymbol{x} >$
    $\boldsymbol{x}_{t+1}^u := \boldsymbol{x}_t + \frac{\boldsymbol{a}_t^u - \boldsymbol{x}_t}{t}$
    $\boldsymbol{x}_{t+1} := \arg\max_{\boldsymbol{x}\in\{\boldsymbol{x}_{t+1}^l,\ \boldsymbol{x}_{t+1}^u\}} f(\boldsymbol{x})$
end for
```

We find out that OPE builds an approximation $F_t(\boldsymbol{x})$ by choosing either likelihood or prior with uniform distribution. In fact, when humans deal with a new sample, one can rely on likelihood if we have observed enough evidence, or rely on prior knowledge if we have been lack of evidence. Based on this natural and simple idea and exploiting the approximation technique of OPE, we propose the BOPE algorithm that still preserves all theoretical guarantees of convergence but more general and flexible by using Bernoulli distribution and two stochastic bounds.

III. BERNOULLI RANDOMNESS IN MAP ESTIMATION

In this section, we introduce a provably fast algorithm, namely BOPE for solving the MAP problem (3) whose objective function $f(\boldsymbol{x})$ is non-convex and smooth on the compact domain Ω. The idea of BOPE is quite simple. The details of BOPE is presented in Algorithm 2.

Denote

$$g_1(\boldsymbol{x}) = \log P(D|\boldsymbol{x}), \quad g_2(\boldsymbol{x}) = \log P(\boldsymbol{x})$$

Assume that $g_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$ are continuously differentiable over the compact domain Ω. We use Bernoulli distribution with parameter $p \in (0, 1)$ to replace for uniform distribution in OPE, and we construct two stochastic sequences that converge to objective function $f(\boldsymbol{x})$: the lower sequence $\{L_t\}$ begun with $g_1(\boldsymbol{x})$, the upper sequence $\{U_t\}$ begun with $g_2(\boldsymbol{x})$.

Given Bernoulli parameter $p \in (0, 1)$, we denote

$$G_1(\boldsymbol{x}) := \frac{g_1(\boldsymbol{x})}{p}, \quad G_2(\boldsymbol{x}) := \frac{g_2(\boldsymbol{x})}{1-p}$$

We initialize $f_1^l := G_1(\boldsymbol{x})$. For each iteration t, $(t = 2, 3, \ldots)$, we pick f_t^l randomly from $\{G_1(\boldsymbol{x}),\ G_2(\boldsymbol{x})\}$ according to the Bernoulli distribution with parameter $p \in (0, 1)$, where

$$P(f_t^l = G_1(\boldsymbol{x})) = p, \quad P(f_t^l = G_2(\boldsymbol{x})) = 1 - p$$

We set

$$L_t := \frac{1}{t}\sum_{h=1}^{t} f_h^l$$

and solve the linear program over Ω:

$$\boldsymbol{a}_t^l := \arg\max_{\boldsymbol{x}\in\Omega} \langle L_t'(\boldsymbol{x}_t), \boldsymbol{x}\rangle$$

then update

$$\boldsymbol{x}_{t+1}^l := \boldsymbol{x}_t + \frac{\boldsymbol{a}_t^l - \boldsymbol{x}_t}{t}$$

Next, we construct the sequence $\{U_t\}$ similarly to the sequence $\{L_t\}$. We initialize $f_1^u := G_2(\boldsymbol{x})$. For each iteration t $(t = 2, 3, \ldots)$, we pick f_t^u randomly from $\{G_1(\boldsymbol{x}),\ G_2(\boldsymbol{x})\}$ according to the Bernoulli distribution with parameter $p \in (0, 1)$, where

$$P(f_t^u = G_1(\boldsymbol{x})) = p\ , \quad P(f_t^u = G_2(\boldsymbol{x})) = 1 - p$$

Then, we obtain $U_t := \frac{1}{t}\sum_{h=1}^{t} f_h^u$ and solve the linear program over Ω:

$$\boldsymbol{a}_t^u := \arg\max_{\boldsymbol{x}\in\Omega} \langle U_t'(\boldsymbol{x}_t), \boldsymbol{x}\rangle$$

and update

$$\boldsymbol{x}_{t+1}^u := \boldsymbol{x}_t + \frac{\boldsymbol{a}_t^u - \boldsymbol{x}_t}{t}$$

It is easy to verify that L_t and U_t are average of all sample functions drawn until the current step. So, they are both guaranteed to converge to $f(\boldsymbol{x})$ as $t \to \infty$, which will be shown in the proof of Theorem 1 in Appendix A. We also see that the Bernoulli parameter p controls how much likelihood and prior contribute to L_t and U_t. At each iteration, using both two stochastic sequences $\{L_t\}$ and $\{U_t\}$ gives us more information about $f(\boldsymbol{x})$, so that we can get chances to faster reach a maximum of $f(\boldsymbol{x})$. We obtain $\{\boldsymbol{x}_t^l\}$ from the sequence $\{L_t\}$, and $\{\boldsymbol{x}_t^u\}$ from $\{U_t\}$. At each iteration, based on the greedy approach, we always compare two values of $f(\boldsymbol{x}_t^u)$ and $f(\boldsymbol{x}_t^l)$, then take the point that has the highest value of objective $f(\boldsymbol{x})$ as

$$\boldsymbol{x}_t := \arg\max_{\boldsymbol{x}\in\{\boldsymbol{x}_t^u, \boldsymbol{x}_t^l\}} f(\boldsymbol{x})$$

BOPE uses Bernoulli distribution which is more general than uniform distribution and creates three sequences $\{\boldsymbol{x}_t^u\}$, $\{\boldsymbol{x}_t^l\}$ and $\{\boldsymbol{x}_t\}$ depending on each other. The following theorem shows some properties of BOPE. Its proof appears in Appendix A.

Theorem 1 (Convergence): Assume that $g_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$ are continuously differentiable over the compact domain Ω. Given the Bernoulli parameter $p \in (0, 1)$, with probability one, the sequence $\{\boldsymbol{x}_t\}$ obtained by Algorithm 2, converges to a local maximal/stationary point $\boldsymbol{x}^*$ of $f(\boldsymbol{x})$ at rate of $\mathcal{O}(1/T)$ where T denotes the number of iterations.

Comparing with other methods such as CCCP and SMM, Algorithm 2 has many preferable properties. First, the convergence rate of CCCP and SMM is unknown for non-convex problems. Second, while each iteration of SMM requires us to solve a convex problem, that of CCCP has to solve a (non-linear) equation system which is expensive and non-trivial in many cases. Note that each iteration of Algorithm 2 requires us to solve the linear programs which are significantly easier than non-linear problems. Therefore, Algorithm 2 promises to be much more efficient than CCCP and SMM. We have shown that BOPE and OPE both converge at a rate of $\mathcal{O}(1/T)$ while PMC converges at a rate of $\mathcal{O}(T^{-1/2})$ [33] and HAMCMC converges at a rate of $\mathcal{O}(T^{-1/3})$ [34] where T is the number of iterations. We emphasize that by using Bernoulli randomness, BOPE is more general and flexible than OPE, when varying the value of Bernoulli parameter p, we control the contribution of each information element to the learning process, so we can obtain variants of BOPE.

Overfitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. When the model does really well on the training data but really bad on real data, thus, the model cannot be generalized. In statistics and machine learning, overfitting occurs when a statistical model describes random error or noise instead of the underlying relationship. Overfitting is a general problem that plagues all machine learning methods. Regularization [41] is a well-known technique to solve ill-posed problems and to prevent overfitting of a learning system. Another property of BOPE is that there is an implicit regularization when solving the MAP problem. This is an outstanding advantage of BOPE compared with previous methods. The idea is to add a regularization term $R(\boldsymbol{x})$ to a loss function loss$(D, \boldsymbol{x})$:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}}[loss(\boldsymbol{D}, \boldsymbol{x}) + \lambda R(\boldsymbol{x})]$$

where λ is a parameter which controls the strength of regularization. In MAP (3), the prior term $-\log P(\boldsymbol{x})$ naturally plays as regularization:

$$\begin{aligned}\boldsymbol{x}^* &= \arg\max_{\boldsymbol{x}\in\Omega}[\log P(D|\boldsymbol{x}) + \log P(\boldsymbol{x})]\\ &= \arg\min_{\boldsymbol{x}\in\Omega}[-\log P(D|\boldsymbol{x}) - \log P(\boldsymbol{x})]\end{aligned}$$

Surprisingly, there is another regularization term when BOPE solves problem (3).

TABLE 1. **Theoretical comparison of inference methods. *T* denotes the number of iterations, and '–' denotes *'unknown'*.**

Methods	Convergence rate	Randomness	Regularization
VB	–	–	–
CVB	–	–	–
CVB0	–	–	–
CGS	–	Uniform	–
CCCP	–	–	–
SMM	–	Uniform	–
PMD	$\mathcal{O}(T^{-1/2})$	Uniform	–
HAMCMC	$\mathcal{O}(T^{-1/3})$	Uniform	–
OPE	$\mathcal{O}(1/T)$	Uniform	–
BOPE (this work)	$\mathcal{O}(1/T)$	Bernoulli	Yes

Theorem 2 (Regularization): Consider the BOPE algorithm for maximizing $f(\boldsymbol{x}) = g_1(\boldsymbol{x}) + g_2(\boldsymbol{x})$, given parameter $p \in (0, 1)$. At each iteration t, BOPE tries to maximize

$$f(\boldsymbol{x}) + R_t(g_1, g_2, p)$$

where $R_t(g_1, g_2, p) = h(t, p)\left(\frac{1}{p}g_1(\boldsymbol{x}) - \frac{1}{1-p}g_2(\boldsymbol{x})\right)$ satisfies $h(t, p) \to 0$ as $t \to \infty$.

The proof of this theorem appears in Appendix B. This theorem essentially says that the regularization term is stochastically composed from the objective function and is vanishing as more iterations are done. Furthermore, parameter p implicitly controls the strength of regularization. Smaller p basically implies slower vanishing of the regularization term.

Table 1 summarizes some properties of different inference methods for probabilistic models. Comparing with other approaches, BOPE has many preferable properties. Among those, implicit regularization is a big advantage of BOPE.

In the next sections, we present the efficiency of BOPE via its application for text analysis and recommender systems.

IV. CASE STUDY 1: APPLICATION TO TEXT ANALYSIS

In this section, for evaluating the efficiency of BOPE on the experimental aspect, we adopt BOPE to solve the MAP problem in topic models, which are powerful tools for text mining [32], [42], [42]–[50]. We will see that the regularization ability in BOPE plays a significant role to prevent overfitting when working with short text, which is an example of sparse data.

The following notations will be used throughout this section.

$\mathcal{V}$: A vocabulary of V terms, often written as $\{1, 2, \ldots, V\}$

$\boldsymbol{d}$: A document represented as a count vector,

$\boldsymbol{d} = (d_1, \ldots, d_V)$, where d_j is the frequency of term j.

n_d: The length of $\boldsymbol{d}$, $n_d = \sum_j d_j$.

$\boldsymbol{\beta}_k$: A topic which is a distribution over the vocabulary $\mathcal{V}$,

$$\boldsymbol{\beta}_k = (\beta_{k1}, \ldots, \beta_{kV})^T, \beta_{kv} \geq 0, \sum_{v=1}^{V}\beta_{kv} = 1.$$

Δ_K: The unit simplex

$$\Delta_K = \{\boldsymbol{x} \in \mathbb{R}^K : \boldsymbol{x} \geq 0, \sum_{k=1}^{K} x_k = 1\}$$

$\overline{\Delta}_K$: The interior of unit simplex Δ_K.

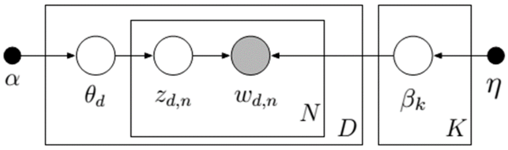

FIGURE 1. Graphical model representation of LDA.

A. THE MAP PROBLEM IN TOPIC MODELS

Topic modeling is a potential approach to help organizing, searching and understanding vast amounts of information. Topic modeling provides a framework to model high-dimensional sparse data. Latent Dirichlet allocation (LDA) [30] is a generative model for modeling texts and discrete data. LDA has been successfully applied in a wide range of areas including text modeling [42], [43], bioinformatics [44], [45], history [32], [46], [47], politics [42], [48], and psychology [49].

LDA often assumes that a corpus is composed from K topics, $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$, each of which is a sample from a V-dimensional Dirichlet distribution, $Dirichlet(\eta)$. Figure 1 represents the graphical model of LDA.

Each document $\boldsymbol{d}$ is a mixture of those topics and is assumed to arise from the following generative process:

1) Draw $\boldsymbol{\theta}_d|\alpha \sim \text{Dirichlet}(\alpha)$
2) For the n^{th} word of $\boldsymbol{d}$:
   - draw topic index $z_{dn}|\boldsymbol{\theta}_d \sim \text{Multinomial}(\boldsymbol{\theta}_d)$
   - draw word $w_{dn}|z_{dn}, \boldsymbol{\beta} \sim \text{Multinomial}(\boldsymbol{\beta}_{z_{dn}})$

Each topic mixture $\boldsymbol{\theta}_d = (\theta_1, \ldots, \theta_K)$ represents the contributions of topics to document $\boldsymbol{d}$ while β_{kj} shows the contribution of term j to topic k. Note that $\boldsymbol{\theta}_d \in \Delta_K, \ \boldsymbol{\beta_k} \in \Delta_V, \ \forall k$. Both $\boldsymbol{\theta}_d$ and z_d are unobserved variables and are local for each document. LDA further assumes that $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ are samples of Dirichlet distributions, more specifically, $\boldsymbol{\theta}_d \sim Dirichlet(\alpha)$ and $\boldsymbol{\beta}_k \sim Dirichlet(\eta)$ where α and η are hyper-parameters.

Originally LDA is applied to model the corpus of text documents in which each document is assumed as a random mixture of topics and a topic is a distribution over words. The learning problem is finding the topic distribution of each document and the distribution of words in topics. When learning these parameters, we have to deal with an inference step which is to find the topic distribution of a document with the known distributions of words in topics. One of the core issues in topic models is posterior inference. It often refers to the problem of estimating the posterior distribution of latent variables for individual documents $\boldsymbol{d}$ such as topic proportion $\boldsymbol{\theta}$. This problem is considered by many researchers in recent years and various algorithms such as VB, CVB, CVB0, CGS, and OPE.

The problem of posterior inference for each document $\boldsymbol{d}$, given a model $\{\boldsymbol{\beta}, \alpha\}$, is to estimate the full joint distribution $P(z_d, \boldsymbol{\theta}_d, \boldsymbol{d}|\boldsymbol{\beta}, \alpha)$. Direct estimation of this distribution is a NP-hard in the worst case [11]. Hence existing inference approaches use different schemes. Some methods such as VB, CVB, CVB0 try to estimate the distribution by maximizing a lower bound of the likelihood $P(\boldsymbol{d}|\boldsymbol{\beta}, \alpha)$, whereas CGS tries to estimate $P(z|\boldsymbol{d}, \boldsymbol{\beta}, \alpha)$. They are being popularly used in topic modeling, but we have not seen any theoretical analysis about how fast they do inference for individual documents. Other good candidates for posterior inference includes CCCP, SMM, OFW, and Threshold Linear Inverse (TLI) [51]. One might employ CCCP and SMM to do inference in topic models. Those two algorithms are guaranteed to converge to a stationary point of the inference problem. However, the rates of convergence of CCCP and SMM are not clearly analyzed in non-convex circumstances such as inferences in topic models.

Unlike the above methods, we approach in a different way, that is MAP. We consider the MAP estimation of topic mixture for a given document $\boldsymbol{d}$:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\Delta_K} P(\boldsymbol{\theta}, \boldsymbol{d}|\boldsymbol{\beta}, \alpha) = \arg\max_{\boldsymbol{\theta}\in\Delta_K} P(\boldsymbol{d}|\boldsymbol{\theta}, \boldsymbol{\beta})P(\boldsymbol{\theta}|\alpha) \quad (5)$$

According to [39], the MAP problem (5) is equivalent to the following:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\Delta_K} [\sum_j d_j \log \sum_{k=1}^K \theta_k \beta_{kj} + (\alpha - 1) \sum_{k=1}^K \log \theta_k] \quad (6)$$

It is shown that this problem is NP-hard in the worst case when $\alpha < 1$ by the authors of [11]. In the case of $\alpha \geq 1$, one can easily show that the problem (6) is concave optimization, therefore it can be solved in polynomial time. Unfortunately, in practice LDA, the parameter α is often small, says $\alpha < 1$, which causes (6) to be non-concave optimization. In this paper, we consider problem (6) in case hyper-parameter $\alpha < 1$. We see that problem (6) has a form as problem (4) when we denote

$$g_1(\boldsymbol{\theta}) := \sum_j d_j \log \sum_{k=1}^K \theta_k \beta_{kj},$$

$$g_2(\boldsymbol{\theta}) := (1 - \alpha) \sum_{k=1}^K \log \theta_k$$

We also see that $g_1(\boldsymbol{\theta}) < 0$, $g_2(\boldsymbol{\theta}) > 0$ and

$$g_1(\boldsymbol{\theta}) < f(\boldsymbol{\theta}) = g_1(\boldsymbol{\theta}) + g_2(\boldsymbol{\theta}) < g_2(\boldsymbol{\theta})$$

then we can design upper and lower bounds of the objective function $f(\boldsymbol{\theta})$ and we can apply BOPE algorithm for solving well the problem (6). We further show in this section the simplicity of using BOPE for designing fast learning algorithm for LDA. More specifically, based on Online-OPE [39], we design Online-BOPE which learns LDA from large corpora in an online fashion. Details of Online-BOPE is presented in Algorithm 3.

This algorithm employs BOPE to do MAP inference for individual documents, and the online scheme [37], [52] to infer global variables (topics). Hence, the stochastic nature appears in both local and global inference phases. Note that the MAP inference of local variables by BOPE has a theoretical guarantee on fast convergence rate. Such a property might help Online-BOPE, the new large-scale learning algorithm,

Algorithm 3 Online-BOPE for Learning LDA From Massive Data

```
Input: training data $\mathcal{C}$ with $D$ documents $K, \alpha, \eta, \tau > 0, \kappa \in (0.5, 1]$
Output: $\boldsymbol{\lambda}$
Initialize $\boldsymbol{\lambda}^0$ randomly
for $t = 1, \ldots, \infty$ do
    Sample a set $\mathcal{C}_t$ consisting of $S$ documents.
    Use BOPE algorithm to do posterior inference for each document $\boldsymbol{d} \in \mathcal{C}_t$, given the global variable $\boldsymbol{\beta}^{t-1} \propto \boldsymbol{\lambda}^{t-1}$ in the last step, to get topic mixture $\boldsymbol{\theta}_d$. Then compute $\boldsymbol{\phi}_d$ as:
        $\phi_{djk} \propto \theta_{dk}\beta_{kj}$
    For each $k \in \{1, 2, \ldots, K\}$, form an intermediate global variable $\hat{\lambda}_k$ for $\mathcal{C}_t$ by
        $\hat{\lambda}_{kj} = \eta + \frac{D}{S}\sum_{\boldsymbol{d} \in \mathcal{C}_t} d_j\phi_{djk}$
    Update the global variable by, where $\rho_t = (t + \tau)^{-\kappa}$,
        $\boldsymbol{\lambda}^t := (1 - \rho_t)\boldsymbol{\lambda}^{t-1} + \rho_t\hat{\boldsymbol{\lambda}}$
end for
```

be more attractive than existing ones. We do experiments with Online-BOPE on five datasets including long texts and short texts and we compared its results with other inference methods such as VB, CVB0, CGS, and OPE.

B. EMPIRICAL EVALUATION

This section is devoted to investigating practical behaviors of BOPE, and how useful it is when BOPE is employed to design a new algorithm for learning topic models at large scales. To this end, we take the following methods, and performance measures into investigation.

Inference Methods: VB, CVB0, CGS, OPE, and BOPE which is our new inference algorithm. CVB0 and CGS have been observing to work best by several previous studies [32], [53], [54]. Therefore, they can be considered as the state-of-the-art inference methods.

In this section, we carry out extensive experiments to investigate the effectiveness of Online-BOPE when comparing with variety of stochastic learning methods: Online-VB [52], Online-CVB0 [55], Online-CGS [32], and Online-OPE [39]. Online-CGS is a hybrid algorithm, for which CGS is used to estimate the distribution of local variables (z) in a document, and VB is used to estimate the distribution of global variables ($\boldsymbol{\lambda}$). Online-CVB0 is an online version of the batch algorithm in [53], where inference for a document is done by CVB0. Online-VB is a stochastic algorithm for which inference for a document is done by VB.

Data for Experiments: Probabilistic topic models have been proven to be effective tools for uncovering the hidden topics of textual corpora. While topic models such as LDA have broad success on news articles and academic papers [30]; they often suffer from bad performance on short texts. Unlike long texts (e.g. carefully edited articles, academic papers), short texts such as mobile short message, instant message, news title, online chat record, blog comment, news comment, etc are often characterized by a very short length, a large vocabulary, a massive size, and noises. The short texts consist of from a dozen words to a few sentences and they do not provide enough word co-occurrence or shared context for a good similarity measure. Short and noisy data poses severe challenges for modeling, and thus traditional methods for learning topic models do not work well or face severe overfitting [7]. Thus, in order to compare BOPE to other inference methods in learning LDA, we use two types of datasets which are long texts and short texts in our experiments. The detailed description for each dataset is shown in Table 2.

TABLE 2. Description of five datasets for our experiments.

Datasets	Corpus size	Average length per doc	Vocabulary size
New York Times	300,000	325.13	102,661
PubMed	330,000	65.12	141,044
Yahoo Questions	517,770	4.73	24,420
Twitter tweets	1,457,687	10.14	89,474
NYT-Titles	1,664,127	5.15	55,488

- Long texts: We use two large datasets of long texts for evaluation: PubMed dataset consists of 330,000 articles from the PubMed central and New York Times dataset consists of 300,000 news.[2]
- Short texts: We use three large datasets of short texts for evaluation: Yahoo Questions crawled from answers.yahoo.com, each document is a question; Tweets from Twitter (twitter.com), each document is the text content of a tweet; NYT-Titles from The New York Times (www.nytimes.com), each document is the title of an article [56]. These datasets are preprocessed by tokenizing, stemming, removing stop-words, removing low-frequency words (appear in less than 3 documents) and removing extremely short documents (less than 3 words).

The shortness of texts poses various difficulties [51], [56], [57] because of its natural characters such as sparseness, large-scale, immediacy, non-standardization [7]. It is difficult for traditional methods to deal with short texts mainly because too limited words in the short text cannot represent the feature space and the relationship between words and documents. Therefore, the usage of both long and short texts in our investigation would show more insights into the performance of different methods. For each corpus, we set aside randomly $1,000$ documents for testing and used the remaining for learning.

Parameter Settings:

- Model parameters: We set the number of topics $K = 100$, the hyper-parameters $\alpha = \frac{1}{K}$ and the topic Dirichlet

[2] The datasets were taken from http://archive.ics.uci.edu/ml/datasets

parameter $\eta = \frac{1}{K}$. These parameters are commonly used in topic models. Such a choice of (α, η) has been observed to work well in many previous studies [52], [55], [58].

- Inference parameters: We choose Bernoulli parameter $p \in \{0.1, 0.2, \ldots, 0.8, 0.9\}$. At most 50 iterations were allowed for BOPE, OPE and VB to do inference. We terminated VB if the relative improvement of the lower bound on likelihood is not better than 10^{-4}. 50 samples were used in CGS for which the first 25 were discarded and the remaining were used to approximate the posterior distribution. 50 iterations were used to do inference in CVB0, in which the first 25 iterations were burned in. Those number of samples/iterations are often enough to get a good inference solution, according to [32], [55].
- Learning parameters: We set the mini-batch size $S = |C_t| = 5,000$, $\kappa = 0.9$, $\tau = 1$. This choice of learning parameters has been found to result in competitive performance of Online-VB and Online-CVB0 [52], [55]. Therefore, it was used in our investigation to avoid possible bias. We used default values for some other parameters in Online-CVB0.

Performance Measures: We have used two measures: Log Predictive Probability (LPP) [32] and Normalised Pointwise Mutual Information (NPMI) [59]. Predictive probability measures the predictiveness and generalization of a model to new data, while NPMI evaluates semantics quality of an individual topic. Details of LPP and NPMI are presented in Appendix D and Appendix E.

Because we use BOPE as an inference method in learning LDA, then we do comparing it with other inference methods via applying to design large-scale learning methods. To avoid randomness, the learning methods for each dataset are run five times and reported their average results.

1) THE EFFECT OF BERNOULLI PARAMETER *p*

In this experiment, we investigate that how important the value of Bernoulli parameter p is in BOPE. Because $p \in (0, 1)$, so we choose p respectively in $\{0.1, 0.2, \ldots, 0.9\}$, then run Online-BOPE in five datasets. We find out that p affects very much in the performance in terms of both measures and on both short texts and long texts. Firstly, we report the performance of Online-BOPE on New York Times and PubMed datasets in Figure 2.

Overall, the value of Bernoulli parameter p hightly effects the performance in terms of both measures, especially on NPMI. More specific, the difference between the highest value and the smallest value in NPMI with same batches of data varying from 0.5 to 1. In addtion, it is showed that the value of Bernoulli parameter p affects to Online-BOPE on PubMed more than on New York Times in LPP, which can be explained by the effect of document length in each dataset. Average document length in PubMed is shorter than that of New York Times, thus it tends to reduce contribution of the likelihood part in the MAP problem.

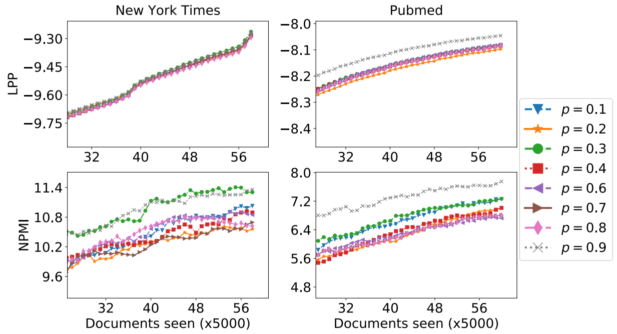

FIGURE 2. Results of Online-BOPE with different value of Bernoulli parameter *p* on long text datasets with LPP and NPMI measures. Higher is better.

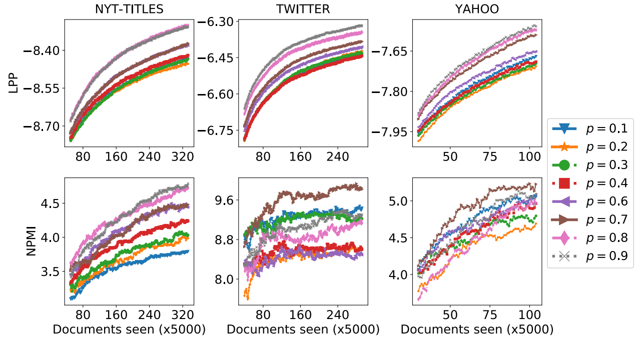

FIGURE 3. Results of Online-BOPE with different value of Bernoulli parameter *p* with LPP and NPMI measures on short-text datasets. Higher is better.

As previous studies have shown [7], [39], LDA does not work well with short texts, even occurs over-fitting. So, we want to point out that our algorithm can help LDA working well on short texts such as NYT-Titles, Twitter tweets and Yahoo question datasets.

Via Figure 3, we show that Bernoulli parameter p has a great impact on the effectiveness of Online-BOPE, especially on short texts. Experimenting on short texts such as Nytimes titles, Twitter tweets and Yahoo question, BOPE gives higher results when p tends to 1. This is a suggestion for selecting parameter p in the BOPE.

2) PERFORMANCE OF THE LEARNING METHODS

LDA can be fit to large datasets of text by using stochastic optimization [43], [52]. However, it fails in the face of large vocabularies and short texts. We evaluate the efficiency of BOPE for solving the MAP problem in topic models via results of Online-BOPE for learning LDA on LPP and NPMI measures and comparing with other learning algorithms such as Online-VB, Online-CVB0, Online-CGS, and Online-OPE. All of the experiments have done on both types of datasets: short texts and long texts.

Long texts: In this experiment, we carry out Online-BOPE in comparing with Online-VB, Online-CVB0, Online-CGS, and Online-OPE on long texts (New York Times and PubMed) with the results shown in Figure 4.

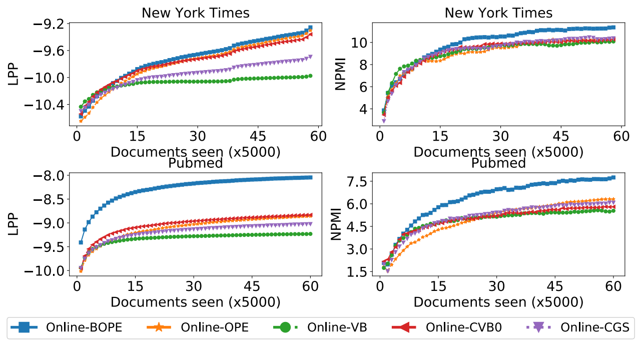

FIGURE 4. Results of the stochastic learning methods on New York Times and PubMed. Higher is better. We find out that Online-BOPE gives the best performance.

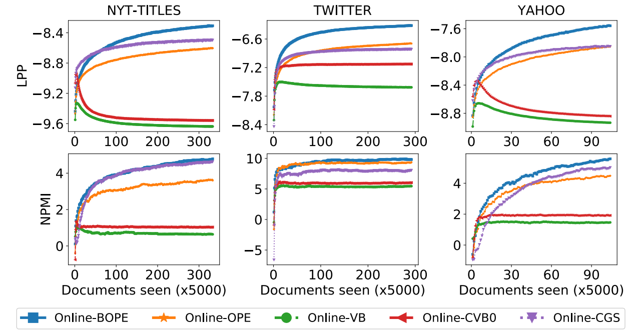

FIGURE 5. Results of the learning methods on short texts: NYT-Titles, Twitter, and Yahoo. We find out that Online-BOPE has the best performance.

We observe that value of LPP and NPMI of all learning methods increase according to the number of documents learned. The explanation for this trend is that the LDA model is often consistent with long texts. However, there is still a huge difference between Online-BOPE and other learning methods, especially on PubMed. This explains that BOPE is especially suitable for documents made the likelihood and prior are not too different, then Bernoulli parameter p controls the ratio of likelihood and prior in the objective function of MAP problem.

Short texts: We investigate the effectiveness of Online-BOPE when working on short texts such as Twitter, NYT-Titles, Yahoo question (see Figure 5). We show that BOPE helping Online-BOPE better than competing methods when working on short texts in some aspects: the predictiveness, generalization and preventing overfitting.

We observe the overfitting of Online-VB and Online-CVB0 in Figure 5. LPP and NPMI measures of Online-VB and Online-CVB0 decrease according to the number of documents learned while LPP and NPMI of Online-CGS, Online-OPE and Online-BOPE sill increase according to the number of documents learned. Thus, it means that the general ability of the model has reduced when using Online-VB and Online-CVB on three short text datasets, especially NYT-Titles and Yahoo datasets which are very short.

Next, we continue to do experiments on short texts and record the experimental results of learning methods after five epochs. For each datasets, we have done Online-BOPE with Bernoulli parameter $p \in \{0.1, 0.2, \ldots, 0.9\}$ then recorded the best results (Online-BOPE-max) and the worst results (Online-BOPE-min) and compared with Online-VB, Online-CVB0, and Online-CGS (see Figure 6).

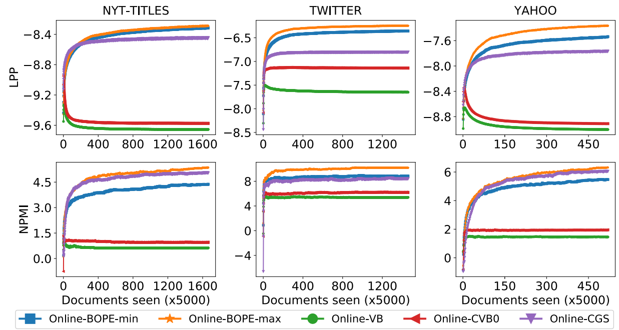

FIGURE 6. Results of the learning methods on short texts: NYT-Titles, Twitter, and Yahoo after five epochs. We find out that Online-BOPE gives the best performance.

We find out that the quality of Online-BOPE is still good after five epochs. However, the over-fitting of Online-VB and Online-CVB0 is more and more, especially on NYT-Titles and Yahoo datasets. It is clear that over-fitting the LDA model depends on the length of documents and Online-VB and Online-CVB0 do not work well on short texts. It can be seen that the length of each document of NYT-Titles and Yahoo is shorter than Twitter. Thus, the quality of the models learned from NYT-Titles or Yahoo by Online-CVB0 and Online-VB decline significantly during the learning process.

V. CASE STUDY 2: APPLICATION TO RECOMMENDER SYSTEMS

In this section, we investigate the application of BOPE for solving the MAP problem in Collaborative Topic Model for Poisson distributed ratings (CTMP) model [8] which is used for recommendation systems. We do not try to compete with state-of-the-art models for recommendation systems, but instead show that the use of BOPE would be more beneficial than existing inference methods.

A. COLLABORATIVE TOPIC MODEL FOR POISSON DISTRIBUTED RATINGS

We will use the following notations in this section:

- U, J: number of users and items respectively.
- $w_j = \{c_j^v\}_{v=1}^V$: bag-of-word representation for item j where c_j^v denotes the frequency of word v in the content/description of item j.
- V: vocabulary size of item content.
- $D = \{r_{uj}, w_j\}_{u=1,j=1}^{U,J}$: dataset of implicit ratings r_{uj} and item content (w_j). Ratings are represented by a matrix $R = \{r_{uj}\}_{U \times J}$, indicating the rating that user u had given to item j. Each rating r_{uj} can take value 1 (indicating that user u “liked” the item j) or 0 (indicating that user u “disliked” or simply did not know about item j).

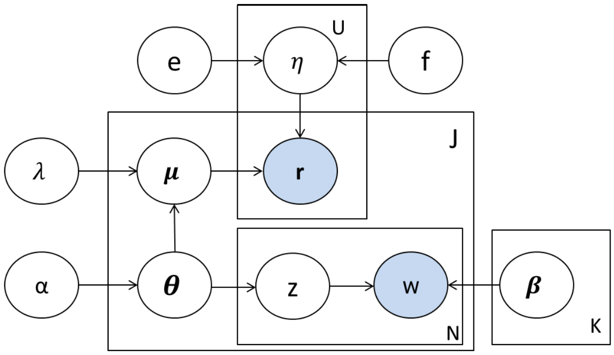

FIGURE 7. Collaborative topic model for poisson distributed ratings (CTMP) model.

- K: the number of topics.
- $\boldsymbol{\beta} = \{\beta_{kv}\}_{K\times V}$: topic representation. Each topic k is a distribution on the vocabulary, and it is represented by vector $\boldsymbol{\beta}_k = \{\beta_{kv}\}_{V\times 1}$ $(\sum_{v=1}^{V} \beta_{kv} = 1, \beta_{kv} \geq 0)$
- $\boldsymbol{\theta}_{1:j}$: item content representation at topic-level. The vector $\boldsymbol{\theta}_j = \{\theta_{jk}\}_{K\times 1}$ is a distribution on topics $(\sum_{k=1}^{K} \theta_{jk} = 1, \theta_{jk} \geq 0)$, and is another representation of item content in the topic space.

Recommender systems often use feedbacks from users to make recommendations of items. The feedbacks may be explicitly or implicitly provided by users. Some systems also use the textual content, such as a product's description or news' content, to further understand the user's preference and then make an accurate recommendation. Collaborative filtering based systems mainly use users' feedbacks alone, whereas a hybrid system can use both feedbacks and contents. CTMP is a hybrid and interpretable probabilistic content-based collaborative filtering model for recommender systems. The model enables both content representation by admixture topic modeling, and computational efficiency from Poisson factorization living together under one tightly coupled probabilistic model, thus addressing the limitation of previous models such as CTR [60] and CTPF [61].

The graphical model of CTMP is represented in Figure 7. The full generative process of CTMP is as follows

1) For each user u, draw η_u where $\eta_{uk} \sim \text{Gamma}(e, f)$
2) For each item j:
    a) Draw topic proportion $\boldsymbol{\theta}_j \sim \text{Dirichlet}(\alpha)$
    b) For the n-th word of item j:
        i) Draw topic index $z_{jn} \sim \text{Categorical}(\boldsymbol{\theta}_j)$
        ii) Draw word $w_{jn} \sim \text{Categorical}(\beta_{z_{jn}})$
    c) Draw latent factor $\boldsymbol{\mu}_j \sim \mathcal{N}(\boldsymbol{\theta}_j, \lambda^{-1}\mathbb{I}_K)$
3) For each user-item pair (u, j), draw $r_{uj} \sim \text{Poisson}(\eta_u^T \mu_j)$

Learning CTMP: Exact computation of the full posterior of latent variables

$$P(\boldsymbol{\theta}, \boldsymbol{\mu}, \eta | D, \alpha, \beta, \lambda, e, f) = \frac{P(\boldsymbol{\theta}, \boldsymbol{\mu}, \eta, D | \alpha, \beta, \lambda, e, f)}{P(D | \alpha, \beta, \lambda, e, f)} \quad (7)$$

is intractable, thus exact inference is not possible. There are two main approaches to the problem: point estimation by MAP estimation, or full Bayesian learning using approximate methods such as MCMC sampling and variational methods [1]. In learning CTMP, we have to learn $\boldsymbol{\theta}_j$. According to [8], we have to learn the point estimation of local topic proportion $\boldsymbol{\theta}_j$ that maximizes

Algorithm 4 Learning CTMP by Coordinate Ascent

```
Input: Observed data $w$, $r$, Bernoulli parameter $p \in (0, 1)$
    and hyper-parameters $\alpha, \lambda, e, f$.
Output: Estimates $\boldsymbol{\theta}$, $\boldsymbol{\mu}$, $\phi_{uj}$, $shp_{uk}$, $rte_{uk}$ and $\boldsymbol{\beta}$.
Initialize $\boldsymbol{\theta}, \boldsymbol{\beta}$ by their respective estimates from LDA
repeat
    for $j = 1 : J$ do
        Update $\boldsymbol{\theta}_j$ by BOPE algorithm
        Update $\boldsymbol{\mu}_j$ as in [8]
    end for
    for $u = 1 : U,\ k = 1 : K$ do
        Update variational parameters as Table 2 in [8]
        $\phi_{ujk} \propto \exp[\log \mu_{jk} + \psi(shp_{uk}) - \log(rte_{uk})]\ \forall j$ if $r_{uj} > 0$
        $shp_{uk} \leftarrow e + \sum_j r_{uj}\phi_{uj}$
        $rte_{uk} \leftarrow f + \sum_j r_{uj}$
        $\beta_{kv} \propto \sum_j c_j^v \theta_{jk}$ , $\forall k, v$
    end for
until convergence
```

$$g(\boldsymbol{\theta}_j) = (\alpha - 1)\sum_k \log \theta_{jk} + \sum_v c_j^v \log \left(\sum_k \theta_{jk}\beta_{kv}\right) - \frac{\lambda}{2}\|\boldsymbol{\theta}_j - \boldsymbol{\mu}_j\|_2^2 \quad (8)$$

We find out that objective function $g(\boldsymbol{\theta}_j)$ is non-convex when $\alpha < 1$. The authors of [8] used OPE [39] algorithm to find the optimum $\boldsymbol{\theta}_j$. At each iteration, OPE tries to direct the solution of the optimization problem to the closed neighbors of the vertices in the convex hull of input domain. OPE provides considerable advantage to computation, with fast convergence rate $\mathcal{O}(1/T)$ and proven quality bound. We denote

$$g_1 = (\alpha - 1)\sum_k \log \theta_{jk} + \sum_v c_j^v \log \left(\sum_k \theta_{jk}\beta_{kv}\right)$$

$$g_2 = -\frac{\lambda}{2}\|\theta_j - \mu_j\|_2^2$$

then objective function $g(\theta_j)$ in (8) be rewritten as $g = g_1 + g_2$. As mentioned before, we find out that BOPE has many advantages overcome to OPE. Thus, we can apply BOPE for learning $\boldsymbol{\theta}_j$ in CTMP. Details for learning CTMP is presented in Algorithm 4.

B. EXPERIMENTAL EVALUATION

We know that predictive performance of a recommender system is measured on the ability to recommend in-matrix items and out-matrix items (also called cold-items). In-items are those containing information from user ratings; cold-items, on the other hand, do not have such information. The tasks of

TABLE 3. Statistics of the experimented datasets. Sparsity indicates proportion of the entries that do not have any positive ratings in each rating matrix ***R***.

Datasets	#Users	#Items	Cold-item	#Ratings
CiteULike	5,551	16,890	3,396	204,986
MovieLens 1M	6,040	3,952	394	1,000,209

recommending items which are all in-items are referred to as in-matrix prediction, and the tasks of recommending both in- and cold-items are called out-of-matrix prediction.

1) EVALUATION MEASURES

Both in-matrix and out-of-matrix prediction are evaluated by precision and recall for all users in test set, measured from top$-M$ recommendation. Top$-M$ recommendation contains items whose predicted ratings are among the M highest. For convenience, precision- and recall-at-M are abbreviated as prec@M and rec@M respectively. By definition

$$prec@M = \frac{1}{U}\sum_{u}\frac{M_u^c}{M}$$

$$rec@M = \frac{1}{U}\sum_{u}\frac{M_u^c}{M_u}$$

where M_u^c is the number of correct items that appear in the top$-M$ recommendation for user u, and M_u is the number of items that user u had rated positive. We do 5$-$fold cross-validation and report the average precision and recall over all users.

2) SETTING OF HYPER-PARAMETERS

We set the Gamma prior parameters $e = f = 0.3$.

3) DATA FOR EXPERIMENTS

In order to investigate application of BOPE in CTMP model, we do many experiments on two datasets from the respective service providers as follows

- CiteULike[3] dataset: a service for managing scientific references. Ratings indicate if an article is in the user libraries.
- MovieLens 1M[4] dataset: User-movie rating dataset. We transformed explicit data into implicit data by let all 4$-$ and 5$-$star ratings be in ‘‘user like this’’ group (i.e. $r_{uj} = 1$)

More details of the processed datasets are in Table 3.

4) EXPERIMENTAL RESULTS

We find out that the Dirichlet prior parameter α, offset precision λ and the number of topics K are the parameters of CTMP which have effects on CTMP. Thus, we consider the effectiveness of BOPE in CTMP via investigating the effects of Dirichlet prior parameter α, offset precision λ, and the number of topics K to CTMP. In this section, we evaluate the BOPE in comparison with OPE when using to learn the parameter $\boldsymbol{\theta}_j$ in the CTMP, and we denote the CTMP model using BOPE as CTMP-BOPE and denote CTMP model using OPE as CTMP-OPE. Details of experimental scenarios are shown in Table 4.

TABLE 4. Some experimental scenarios. Note that CTMP depends on the Dirichlet prior parameter α, offset precision λ and the number of topics K.

Parameters fixed	Parameters changed
$\lambda = 1,000;\ K = 100$	$p = 0.9;\ \alpha \in \{1; 0.1; 0.01; 0.001; 0.0001\}$
$\alpha = 0.01;\ K = 100$	$p = 0.9;\ \lambda \in \{1; 10; 100; 1,000; 10,000\}$
$\alpha = 0.01;\ \lambda = 1,000$	$p = 0.9;\ K \in \{50; 100; 150; 200; 250\}$
$\lambda = 1,000;\ K = 100$	$p = 0.7;\ \alpha \in \{1; 0.1; 0.01; 0.001; 0.0001\}$
$\alpha = 1;\ K = 100$	$p = 0.7;\ \lambda \in \{1; 10; 100; 1,000; 10,000\}$
$\alpha = 1;\ \lambda = 1,000$	$p = 0.7;\ K \in \{50; 100; 150; 200; 250\}$

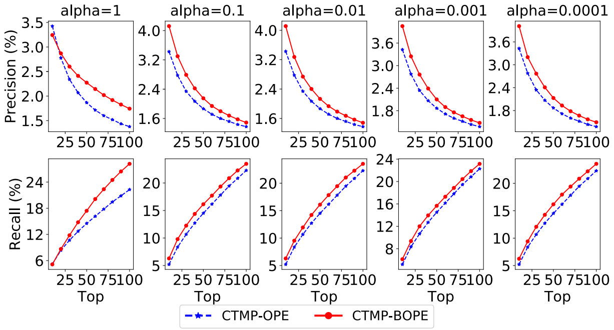

FIGURE 8. Influence of Dirichlet prior parameter α to CTMP model when using OPE and BOPE as inference methods on CiteULike dataset. We fix offset precision $\lambda = 1,000$, the number of topics $K = 100$ and Bernoulli parameter $p = 0.9$. Higher is better.

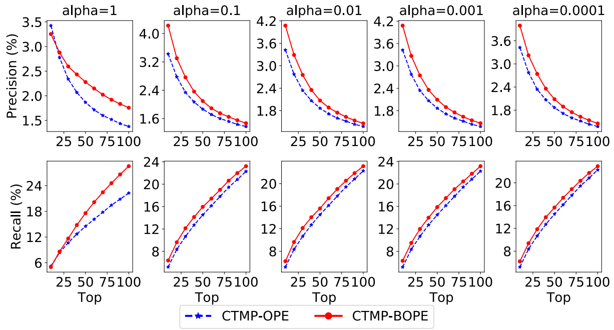

FIGURE 9. Influence of Dirichlet prior parameter α to CTMP model when using OPE and BOPE as inference methods on CiteULike. We fix offset precision $\lambda = 1,000$, the number of topics $K = 100$ and Bernoulli parameter $p = 0.7$. Higher is better.

Firstly, we fix offset precision $\lambda = 1,000$, the number of topics $K = 100$ then change Dirichlet prior parameter $\alpha \in \{1, 0.1, 0.01, 0.001, 0.0001\}$. Experimental results are presented from Figure 8 to 11. Parameter α helps to control the sparsity of topic mixture $\boldsymbol{\theta}$ for each document. CTMP is stable when varying $\alpha \in \{0.1; 0.01; 0.001; 0.0001\}$ on two datasets. However, we find out that with Dirichlet prior parameter $\alpha = 1$, offset precision $\lambda = 1,000$ and the number

[3] This dataset was taken from http://www.citeulike.org/faq/data.adp

[4] This dataset was taken from https://grouplens.org/datasets/movielens/1m/

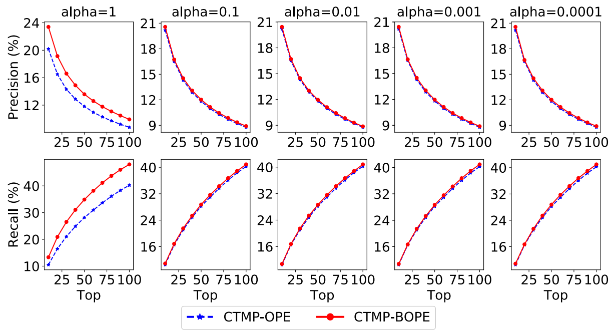

FIGURE 10. Influence of Dirichlet prior parameter α to CTMP model when using OPE and BOPE as inference methods on MovieLens 1M. We fix offset precision $\lambda = 1,000$, the number of topics $K = 100$ and Bernoulli parameter $p = 0.9$. Higher is better.

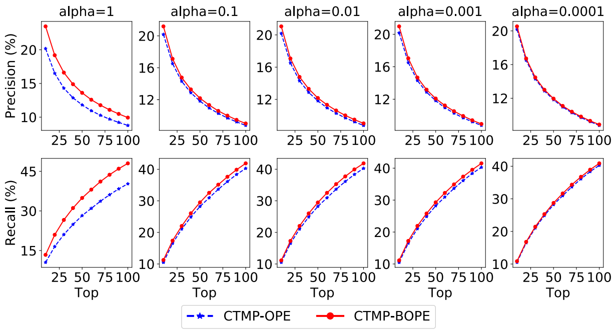

FIGURE 11. Influence of Dirichlet prior parameter α to CTMP model when using OPE and BOPE as inference methods on MovieLens 1M. We fix offset precision $\lambda = 1,000$, the number of topics $K = 100$ and Bernoulli parameter $p = 0.7$. Higher is better.

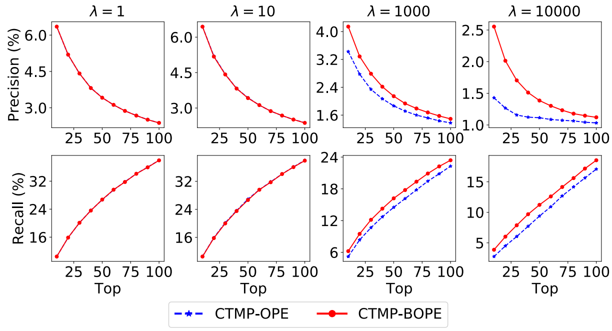

FIGURE 12. Influence of offset precision λ to CTMP model when using OPE and BOPE as inference methods on CiteULike. We fix Dirichlet prior parameter $\alpha = 0.01$, the number of topics $K = 100$, and Bernoulli parameter $p = 0.9$. Higher is better.

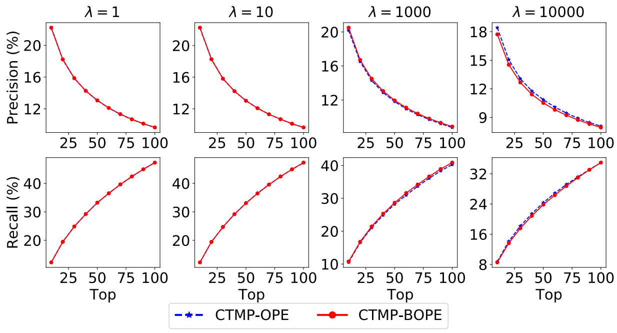

FIGURE 13. Influence of offset precision λ to CTMP model when using OPE and BOPE as inference methods on MovieLens 1M. We fix Dirichlet prior parameter $\alpha = 0.01$, the number of topics $K = 100$ and Bernoulli parameter $p = 0.9$. Higher is better.

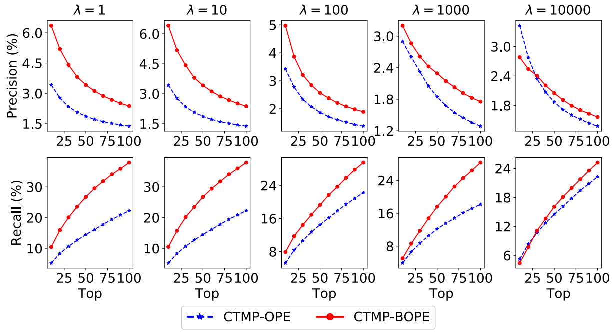

FIGURE 14. Influence of offset precision λ to CTMP model when using OPE and BOPE as inference methods on CiteULike. We fix Dirichlet prior parameter $\alpha = 1$, the number of topics $K = 100$ and Bernoulli parameter $p = 0.7$. Higher is better.

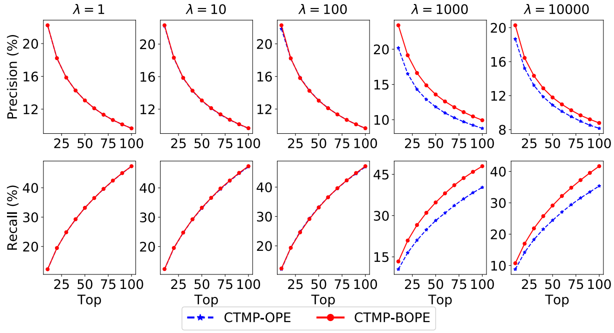

FIGURE 15. Influence of offset precision λ to CTMP model when using OPE and BOPE as inference methods on MovieLens 1M. We fix Dirichlet prior parameter $\alpha = 1$, the number of topics $K = 100$ and Bernoulli parameter $p = 0.7$. Higher is better.

of topics $K = 100$, CTMP-BOPE is better than CTMP-OPE on both precision and recall measures and on two datasets. This is proof of the effectiveness of BOPE in recommendation system applications.

Secondly, we fix Dirichlet prior parameter $\alpha = 0.01$, the number of topics $K = 100$ and choose Bernoulli parameter $p = 0.9$, then change offset precision $\lambda \in \{1; 10; 1, 000; 10, 000\}$. These experimental results are presented in Figure 12 and Figure 13.

We fix Dirichlet prior parameter $\alpha = 1$, the number of topics $K = 100$ and choose Bernoulli parameter $p = 0.7$, then change offset precision $\lambda \in \{1; 10; 100; 1, 000; 10, 000\}$. These experimental results are presented in Figure 14 and Figure 15.

Note that λ is a parameter for the fluctuation of μ around $\boldsymbol{\theta}$. Via Figure 14 and Figure 15, we see that CTMP-BOPE gives better results than CTMP-OPE with setting the Dirichlet prior parameter $\alpha = 1$ and number of topics $K = 100$.

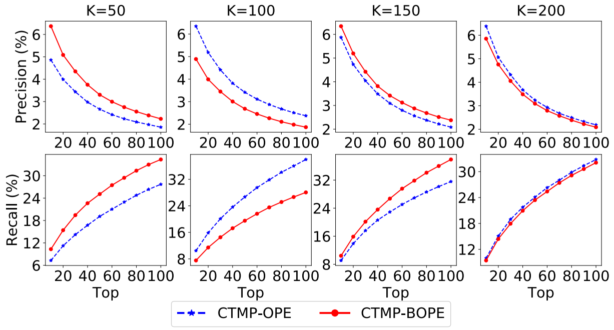

FIGURE 16. Influence of number of topics K to CTMP model when using OPE and BOPE as inference methods on CiteULike dataset. We fix Dirichlet prior parameter $\alpha = 0.01$, the number of topics $K = 100$ and Bernoulli parameter $p = 0.9$. Higher is better.

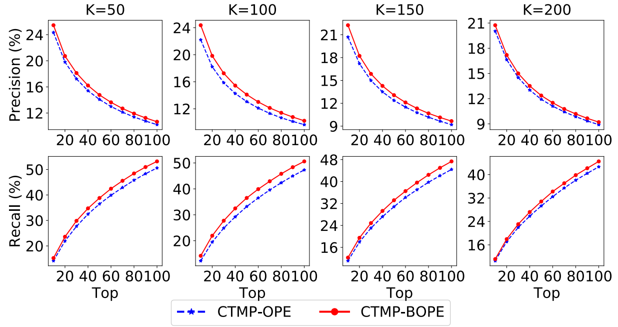

FIGURE 17. Influence of number of topics K to CTMP model when using OPE and BOPE as inference methods on MovieLens 1M. We fix Dirichlet prior parameter $\alpha = 0.01$, the number of topics $K = 100$ and Bernoulli parameter $p = 0.9$. Higher is better.

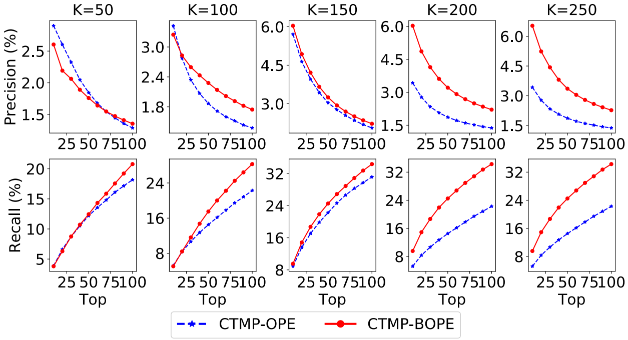

FIGURE 18. Influence of number of topics K to CTMP model when using OPE and BOPE as inference methods on CiteULike dataset. We fix Dirichlet prior parameter $\alpha = 1$, the number of topics $K = 100$ and Bernoulli parameter $p = 0.7$. Higher is better.

Thirdly, to investigate the influence of number of topics K in CTMP, we fix the Dirichlet prior parameter $\alpha = 0.01$, offset precision $\lambda = 1,000$ and choose Bernoulli parameter $p = 0.9$, then change the number of topics $K \in \{50; 100; 150; 200\}$. These experimental results are presented in Figure 16 and Figure 17.

P We fix the Dirichlet prior parameter $\alpha = 1$, offset precision $\lambda = 1,000$ and choose Bernoulli parameter $p = 0.7$, then change the number of topics $K \in \{50; 100; 150; 200; 250\}$. These experimental results are presented in Figure 18 and 19.

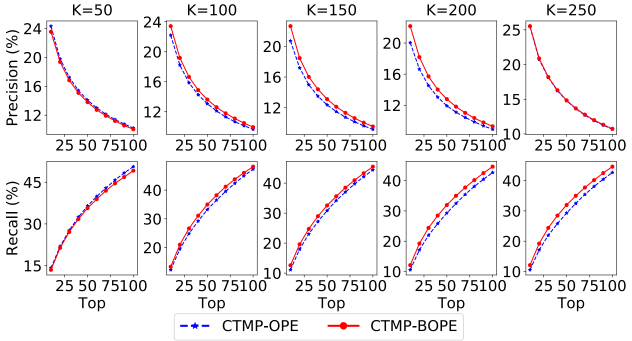

FIGURE 19. Influence of number of topics K to CTMP model when using OPE and BOPE as inference methods on MovieLens 1M. We fix Dirichlet prior parameter $\alpha = 1$, the number of topics $K = 100$ and Bernoulli parameter $p = 0.7$. Higher is better.

The influence of K is more slightly obvious than α and λ in CTMP. The number of hidden topics K expresses the complex of model, and it depends on datasets. Via Figures 16, 17, 18 and 19, we see that CTMP-BOPE is better than CTMP-OPE especially when number of hidden topics $K = 200$ or $K = 250$ and on CiteULike dataset.

We find out that the CTMP-BOPE generally performs better than CTMP-OPE on both precision and recall measures. Thus, this is an intuitive proof of the attractive properties of using Bernoulli distribution and two stochastic bounds over previous methods.

VI. CONCLUSION

In this paper, we have discussed how the MAP problem in probability models can be solved efficiently via using BOPE which is a new stochastic optimization algorithm using Bernoulli randomness. In theory, BOPE has a fast convergence rate and an implicit regularization role which are the most important characters among existing state-of-the-art inference methods. In practice, we have demonstrated that BOPE is successful when applied to text analysis and recommender systems. We emphasize that the parameter $p \in (0, 1)$ in BOPE is a flexible way to deal with different datasets, especially short texts and as well as prevent overfitting. In conclusion, based on theoretical analysis and extensive experiments, we confirm that BOPE is a good candidate for solving the non-convex MAP problem.

APPENDIX A
PROOF OF THEOREM 1

The objective function $f(\boldsymbol{x})$ is non-convex. Different from convex optimization, the criterion used for the convergence analysis is important in non-convex optimization. For unconstrained optimization problems, the gradient norm $\|\nabla f(\boldsymbol{x})\|$ is typically used to measure convergence, because $\|\nabla f(\boldsymbol{x})\| \to 0$ captures convergence to a stationary point. However, this criterion can not be used for constrained optimization problems. Instead, we use the "Frank-Wolfe gap" criterion in [25] for proof of Theorem 1.

We denote

$$G_1(\boldsymbol{x}) := \frac{g_1(\boldsymbol{x})}{p}, \quad G_2(\boldsymbol{x}) := \frac{g_2(\boldsymbol{x})}{1-p}$$

Then, we see that

$$f(\boldsymbol{x}) = g_1(\boldsymbol{x}) + g_2(\boldsymbol{x}) = pG_1(\boldsymbol{x}) + (1-p)G_2(\boldsymbol{x})$$

The first, we consider the sequence $\{U_t\}$.

We set $f_1^u := G_2(\boldsymbol{x})$. For each iteration t ($t = 2, 3, \dots$), we pick f_t^u randomly from $\{G_1(\boldsymbol{x}), G_2(\boldsymbol{x})\}$ according to the Bernoulli distribution with parameter $p \in (0, 1)$, where

$$\{P(f_t^u = G_1(\boldsymbol{x})) = p; P(f_t^u = G_2(\boldsymbol{x})) = 1 - p\}$$

and we have $U_t := \frac{1}{t}\sum_{h=1}^{t} f_h^u$.

Let a_t and $b_t = t - a_t$ be the number of times that we have already picked $G_1(\boldsymbol{x})$ and $G_2(\boldsymbol{x})$ respectively after t iterations. Thus, we have

$$U_t = \frac{1}{t}(a_t G_1 + (t - a_t)G_2) \tag{9}$$

Denote $S_t = a_t - tp$, we obtain

$$U_t - f = \frac{S_t}{t}(G_1 - G_2) \tag{10}$$

$$U_t' - f' = \frac{S_t}{t}(G_1' - G_2') \tag{11}$$

We see that $\frac{S_t}{t} \to 0$ as $t \to \infty$ with probability 1. Combining this with (10), we conclude that the sequence $U_t \to f$ with probability 1, Also due to (11), the derivative sequence $U_t' \to f'$ as $t \to +\infty$. The convergence holds for any $\boldsymbol{x} \in \overline{\Omega}$.

Consider

$$\begin{aligned}\langle U_t'(\boldsymbol{x}_t), \frac{\boldsymbol{a}_t^u - \boldsymbol{x}_t}{t}\rangle &= \langle U_t'(\boldsymbol{x}_t) - f'(\boldsymbol{x}_t), \frac{\boldsymbol{a}_t^u - \boldsymbol{x}_t}{t}\rangle \\ &\quad + \langle f'(\boldsymbol{x}_t), \frac{\boldsymbol{a}_t^u - \boldsymbol{x}_t}{t}\rangle \\ &= \frac{S_t}{t^2}\langle G_1'(\boldsymbol{x}_t) - G_2'(\boldsymbol{x}_t), \boldsymbol{a}_t^u - \boldsymbol{x}_t\rangle \\ &\quad + \langle f'(\boldsymbol{x}_t), \frac{\boldsymbol{a}_t^u - \boldsymbol{x}_t}{t}\rangle\end{aligned}$$

Note that g_1 and g_2 are Lipschitz continuous on Ω. Hence there exists a constant L such that

$$\langle f'(z), y - z\rangle \leq f(y) - f(z) + L\|y - z\|^2, \quad \forall\, y, z \in \Omega$$

We have

$$\begin{aligned}\langle f'(\boldsymbol{x}_t), \frac{\boldsymbol{a}_t^u - \boldsymbol{x}_t}{t}\rangle &= \langle f'(\boldsymbol{x}_t), \boldsymbol{x}_{t+1}^u - \boldsymbol{x}_t\rangle \\ &\leq f(\boldsymbol{x}_{t+1}^u) - f(\boldsymbol{x}_t) + L\|\boldsymbol{x}_{t+1}^u - \boldsymbol{x}_t\|^2 \\ &= f(\boldsymbol{x}_{t+1}^u) - f(\boldsymbol{x}_t) + L\|\frac{\boldsymbol{a}_t^u - \boldsymbol{x}_t}{t}\|^2\end{aligned}$$

We have $\boldsymbol{x}_{t+1} := \arg\max_{\boldsymbol{x} \in \{\boldsymbol{x}_{t+1}^u, \boldsymbol{x}_{t+1}^l\}} f(\boldsymbol{x})$

So that

$$f(\boldsymbol{x}_{t+1}^u) \leq f(\boldsymbol{x}_{t+1})$$

Since $\boldsymbol{a}_t^u$ and $\boldsymbol{x}_t$ belong to Ω, the quantity $|\langle G_1'(\boldsymbol{x}_t) - G_2'(\boldsymbol{x}_t), \boldsymbol{a}_t^u - \boldsymbol{x}_t\rangle|$ and $\|\boldsymbol{a}_t^u - \boldsymbol{x}_t\|^2$ are bounded above for any iteration t. Therefore, there exits a constant $c_1 > 0$ such that

$$\langle U_t'(\boldsymbol{x}_t), \frac{\boldsymbol{a}_t^u - \boldsymbol{x}_t}{t}\rangle \leq c_1\frac{|S_t|}{t^2} + f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}_t) + \frac{c_1 L}{t^2} \tag{12}$$

Summing both sides of (12) for all iterations $t \geq 1$, we have

$$\begin{aligned}&\sum_{t=1}^{+\infty}\frac{1}{t}\langle U_t'(\boldsymbol{x}_t), \boldsymbol{a}_t^u - \boldsymbol{x}_t\rangle \\ &\qquad \leq \sum_{t=1}^{+\infty} c_1\frac{|S_t|}{t^2} + f(\boldsymbol{x}_{+\infty}) - f(\boldsymbol{x}_1) + \sum_{t=1}^{+\infty}\frac{c_1 L}{t^2}\end{aligned} \tag{13}$$

Because $f(\boldsymbol{x})$ is bounded then $f(\boldsymbol{x}_{+\infty})$ is bounded. Note that $S_t = \mathcal{O}(\sqrt{t\log t})$ [62], and hence $\sum_{t=1}^{+\infty} c_1\frac{|S_t|}{t^2}$ converges in probability 1 and $\sum_{t=1}^{+\infty}\frac{L}{t^2}$ also is bounded. Hence, the right-hand side of (13) is finite. In addition, $\langle U_t'(\boldsymbol{x}_t), \boldsymbol{a}_t^u\rangle > \langle U_t'(\boldsymbol{x}_t), \boldsymbol{x}_t\rangle$ for any $t > 0$ because of $\boldsymbol{a}_t^u = \arg\max_{\boldsymbol{x}\in\Omega}\langle U_t'(\boldsymbol{x}_t), \boldsymbol{x}\rangle$. Therefore, we obtain the following

$$0 \leq \sum_{t=1}^{+\infty}\frac{1}{t}\langle U_t'(\boldsymbol{x}_t), \boldsymbol{a}_t^u - \boldsymbol{x}_t\rangle < \infty \tag{14}$$

In other words, the series $\sum_{t=1}^{+\infty}\frac{1}{t}\langle U_t'(\boldsymbol{x}_t), \boldsymbol{a}_t^u - \boldsymbol{x}_t\rangle$ converges to a finite constant. Note that $\langle U_t'(\boldsymbol{x}_t), \boldsymbol{a}_t^u - \boldsymbol{x}_t\rangle \geq 0$ for any t. If there exists constant $c_2 > 0$ satisfying $\langle U_t'(\boldsymbol{x}_t), \boldsymbol{a}_t^u - \boldsymbol{x}_t\rangle \geq c_2$ for an infinite number of t's, then the series $\sum_{t=1}^{+\infty}\frac{1}{t}\langle U_t'(\boldsymbol{x}_t), \boldsymbol{a}_t^u - \boldsymbol{x}_t\rangle$ could not converge to a finite constant, which is in contrary to (14). Therefore,

$$\langle U_t'(\boldsymbol{x}_t), \boldsymbol{a}_t^u - \boldsymbol{x}_t\rangle \to 0 \text{ as } t \to +\infty \tag{15}$$

Because of $U_t' \to f'$ as $t \to \infty$ and both U_t' and f' are continuous, combining with (15) we have

$$\langle f'(\boldsymbol{x}_t), \boldsymbol{a}_t^u - \boldsymbol{x}_t\rangle \to 0 \text{ as } t \to +\infty \tag{16}$$

Using the "Frank-Wolfe gap" criterion in [25], from (16), we have $\boldsymbol{x}_t \to \boldsymbol{x}^*$ as $t \to +\infty$. In other words, $\boldsymbol{x}_t$ converges in probability to a stationary point $\boldsymbol{x}^*$ of $f(\boldsymbol{x})$ ■

APPENDIX B
PROOF OF THEOREM 2

Consider the sequence $\{U_t\}$ built as follows:

- Given the Bernoulli parameter $p \in (0, 1)$,

$$G_1(\boldsymbol{x}) = \frac{g_1(\boldsymbol{x})}{p}; \quad G_2(\boldsymbol{x}) = \frac{g_2(\boldsymbol{x})}{1-p}$$

- Initialize $f_1^u = G_2(\boldsymbol{x})$,
- For each iteration t, ($t \geq 2$), pick f_t^u randomly from $\{G_1(\boldsymbol{x}),\ G_2(\boldsymbol{x})\}$ with probability $p \in (0, 1)$ where

$$P(f_t^u = G_1(\boldsymbol{x})) = p, \quad P(f_t^u = G_2(\boldsymbol{x})) = 1 - p$$

- Then, we obtain a sequence $\{U_t\}$:

$$U_t := \frac{1}{t}\sum_{h=1}^{t} f_h^u$$

We find out that $\{U_t\}$ is the approximation of objective function $f(\boldsymbol{x})$, and U_t is the average of t random variables $\{f_1^u, f_2^u, \dots, f_t^u\}$.

Let a_t and b_t be the number of times that we have already picked $G_1(\boldsymbol{x})$ and $G_2(\boldsymbol{x})$ respectively in U_t after t iterations. We have $a_t + b_t = t$ and $U_t = \frac{1}{t}(a_t G_1 + b_t G_2)$. We find out that a_t follows the binomial distribution with parameters t và p. Instead of optimizing directly on the true objective function $f(\boldsymbol{x})$, BOPE maximizes the approximation $U_t(\boldsymbol{x})$. Denote $S_t = a_t - tp$ and $f(\boldsymbol{x}) = g_1(\boldsymbol{x}) + g_2(\boldsymbol{x}) = pG_1(\boldsymbol{x}) + (1-p)G_2(\boldsymbol{x})$, the sequence $U_t(\boldsymbol{x})$ is rewritten as:

$$U_t(\boldsymbol{x}) = f(\boldsymbol{x}) + \frac{S_t}{t}(G_1(\boldsymbol{x}) - G_2(\boldsymbol{x})) \tag{17}$$

According to (17), U_t is a sum of objective function $f(\boldsymbol{x})$ and $\frac{S_t}{t}(G_1(\boldsymbol{x}) - G_2(\boldsymbol{x}))$. Thus, U_t is the approximations of objective function $f(\boldsymbol{x})$, and $\frac{S_t}{t}(G_1(\boldsymbol{x}) - G_2(\boldsymbol{x}))$ is the regularization term. According to the law of iterated logarithm [62] and proof of Theorem 1, we obtain $\frac{S_t}{t} \to 0$ as $t \to \infty$ with probability one, then $\frac{S_t}{t}(G_1(\boldsymbol{x}) - G_2(\boldsymbol{x}))$ converges to 0 as $t \to \infty$.

According to the construction of BOPE algorithm, we have

$$\frac{S_t}{t} = \frac{a_t - tp}{t} = \frac{a_t}{t} - p$$

where $E[\frac{a_t}{t}] = p$ and $D[\frac{a_t}{t}] = \frac{p(1-p)}{t}$.Therefore, $E[\frac{S_t}{t}] = 0$ and $D[\frac{S_t}{t}] = \frac{p(1-p)}{t} \to 0$ as $t \to \infty$. In addition, analyzing the function $g(p) = p(1-p)$ with $p \in (0, 1)$, we find out that $g(p)$ reaches a maximum when $p = \frac{1}{2}$, and if $p \to 0$ or $p \to 1$ then $g(p) = p(1-p) \to 0$. Thus, we have

$$U_t(\boldsymbol{x}) = f(\boldsymbol{x}) + (\frac{a_t}{t} - p)(\frac{g_1(\boldsymbol{x})}{p} - \frac{g_2(\boldsymbol{x})}{1-p})$$

Denote $h(t,p) = \frac{a_t}{t} - p$, we have that $h(t,p)$ goes to 0 as $t \to \infty$. Therefore, we have

$$R_t(g_1, g_2, p) = h(t,p)(\frac{g_1(\boldsymbol{x})}{p} - \frac{g_2(\boldsymbol{x})}{1-p}) \to 0 \text{ as } t \to \infty$$

and

$$U_t(\boldsymbol{x}) = f(\boldsymbol{x}) + R_t(g_1, g_2, p)$$

where $R_t(g_1, g_2, p)$ has the role as a regularization term satisfying $h(t,p) = \frac{a_t}{t} - p$ goes to 0 as $t \to \infty$. Thus, the regularization term $R_t(g_1, g_2, p)$ depends heavily on the value of the Bernoulli parameter $p \in (0, 1)$. Therefore, in essence, the Bernoulli parameter $p \in (0, 1)$ is considered as the regularization parameter in order to the BOPE algorithm becomes an effective method for solving the MAP problem. ■

APPENDIX C
EXTRA EXPERIMENTAL RESULTS IN CTMP MODEL

In this section, we present some results of BOPE applying in the CTMP model when changing parameters such as the number of topics K, Dirichlet prior parameter α, offset precision λ and Bernoulli parameter p. CiteULike dataset is used. The results are reported in Figures 20, 21, 22.

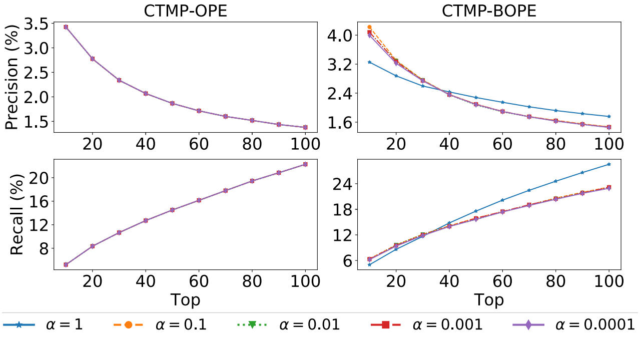

FIGURE 20. We fix offset precision $\lambda = 1000$, the number of topics $K = 100$ and choose Bernoulli parameter $p = 0.7$ then change Dirichlet prior parameter $\alpha \in \{1; 0.1; 0.01; 0.001; 0.0001\}$.

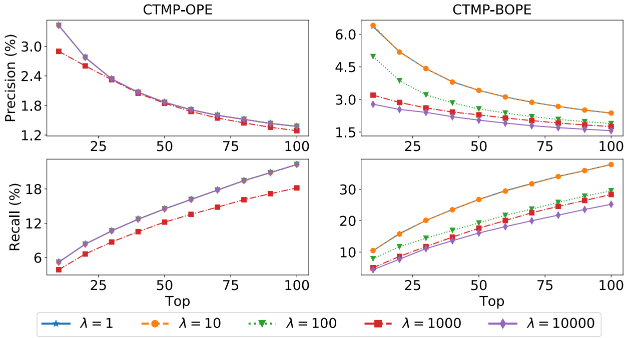

FIGURE 21. We fix Dirichlet prior parameter $\alpha = 1$, the number of topics $K = 100$ and choose Bernoulli parameter $p = 0.7$, then change offset precision $\lambda \in \{1; 10; 100; 1000; 10000\}$.

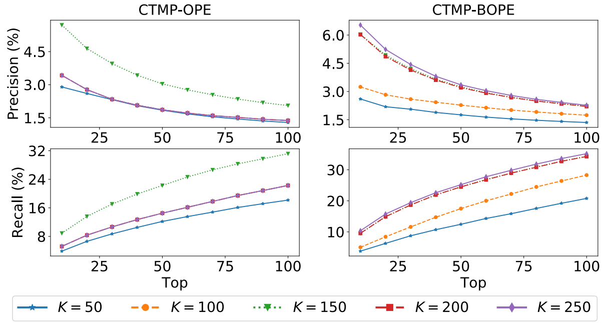

FIGURE 22. We fix the Dirichlet prior parameter $\alpha = 1$, offset precision $\lambda = 1000$ and choose Bernoulli parameter $p = 0.7$, then change the number of topics $K \in \{50; 100; 150; 200; 250\}$.

APPENDIX D
PREDICTIVE PROBABILITY

Predictive Probability shows the predictiveness and generalization of a model M on new data. We followed the procedure in [32] to compute this measurement. For each document in a testing dataset, we divided randomly into two disjoint parts w_{obs} and w_{ho} with a ratio of 80:20. We next did inference for w_{obs} to get an estimate of $\mathbb{E}(\theta^{obs})$. Then we approximated the predictive probability as

$$P(w_{ho}|w_{obs}, \mathcal{M}) \simeq \prod_{(w\in w_{ho})} \sum_{k=1}^{K} \mathbb{E}(\theta_k^{obs})\mathbb{E}(\beta_{kw})$$

$$\text{Log Predictive Probability} = \log \frac{P(w_{ho}|w_{obs}, \mathcal{M})}{|w_{ho}|}$$

where $\mathcal{M}$ is the model to be measured. We estimated $\mathbb{E}(\beta_k) \propto \lambda_k$ for the learning methods which maintain a variational distribution (λ) over topics. Log Predictive Probability was averaged from 5 random splits, each was on 1000 documents.

APPENDIX E
NPMI

NPMI measurements helps us to see the coherence or semantic quality of individual topics. According to [63], NPMI agrees well with human evaluation on interpretability of topic models. For each topic t, we take the set $\{w_1, w_2, \ldots, w_n\}$ of top n terms with highest probabilities. We then computed

$$NPMI(t) = \frac{2}{n(n-1)} \sum_{j=2}^{n} \sum_{i=1}^{j-1} \frac{\log \frac{P(w_j, w_i)}{P(w_j)P(w_i)}}{-\log P(w_j, w_i)}$$

where $P(w_i, w_j)$ is the probability that terms w_i and w_j appear together in a document. We estimated those probabilities from the training data. In our experiments, we chose top $n = 10$ terms for each topic. Overall, NPMI of a model with K topics is averaged as:

$$NPMI = \frac{1}{K} \sum_{t=1}^{K} NPMI(t)$$

REFERENCES

[1] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[2] C. Robert, *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer, 2007.

[3] Z. L. Hariprasad Kodamana and A. A. Biao Huang, "A GMM-MRF based image segmentation approach for interface level estimation," *IFAC-PapersOnLine*, vol. 52, no. 1, pp. 28–33, 2019.

[4] M. Pereyra, "Revisiting maximum-a-posteriori estimation in log-concave models," *SIAM J. Imag. Sci.*, vol. 12, no. 1, pp. 650–670, Jan. 2019.

[5] S. Jameel, Z. Fu, B. Shi, W. Lam, and S. Schockaert, "Word embedding as maximum a posteriori estimation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 6562–6569.

[6] K. Than, X. Bui, T. Nguyen-Trong, K. Truong, S. Nguyen, B. Tran, L. Ngo, and A. Nguyen-Duc, "Can machines learn continuously? A tutorial of the Bayesian approach," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*. Bellingham, WA, USA: SPIE, 2019.

[7] C. Ha, V.-D. Tran, L. Ngo Van, and K. Than, "Eliminating overfitting of probabilistic topic models on short and noisy text: The role of dropout," *Int. J. Approx. Reasoning*, vol. 112, pp. 85–104, Sep. 2019.

[8] H. M. Le, S. Ta Cong, Q. Pham The, N. Van Linh, and K. Than, "Collaborative topic model for Poisson distributed ratings," *Int. J. Approx. Reasoning*, vol. 95, pp. 62–76, Apr. 2018.

[9] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss, "Tightening lp relaxations for MAP using message passing," in *Proc. 24th Conf. Uncertainty Artif. Intell., UAI*, 2008, pp. 503–510.

[10] A. Kumar and S. Zilberstein, "MAP estimation for graphical models by likelihood maximization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1180–1188.

[11] D. Sontag and D. Roy, "Complexity of inference in latent Dirichlet allocation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1008–1016.

[12] S. E. Shimony, "Finding MAPs for belief networks is NP-hard," *Artif. Intell.*, vol. 68, no. 2, pp. 399–410, Aug. 1994.

[13] C. Tosh and S. Dasgupta, "The relative complexity of maximum likelihood estimation, MAP estimation, and sampling," *Proc. Mach. Learn. Res.*, vol. 99, pp. 1–43, Jun. 2019.

[14] T. Hazan, F. Orabona, A. D. Sarwate, S. Maji, and T. S. Jaakkola, "High dimensional inference with random maximum a-posteriori perturbations," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6539–6560, Oct. 2019.

[15] P. Swoboda and V. Kolmogorov, "MAP inference via block-coordinate frank-wolfe algorithm," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11146–11155.

[16] H. Lang, D. Sontag, and A. Vijayaraghavan, "Block stability for MAP inference," in *Proc. Mach. Learn. Res.*, vol. 89, 2019, pp. 216–225.

[17] A. Gane, T. Hazan, and T. Jaakkola, "Learning with maximum a-posteriori perturbation models," in *Artificial Intelligence and Statistics*, 2014, pp. 247–256.

[18] M. Pereyra, "Maximum-a-posteriori estimation with Bayesian confidence regions," *SIAM J. Imag. Sci.*, vol. 10, no. 1, pp. 285–302, Jan. 2017.

[19] T. Helin and M. Burger, "Maximum a posteriori probability estimates in infinite-dimensional Bayesian inverse problems," *Inverse Problems*, vol. 31, no. 8, Aug. 2015, Art. no. 085009.

[20] B. Savchynskyy, J. H. Kappes, P. Swoboda, and C. Schnörr, "Global map-optimality by shrinking the combinatorial search area with convex relaxation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1950–1958.

[21] A. L. Yuille, A. Rangarajan, and A. Yuille, "The concave-convex procedure (CCCP)," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2, 2002, pp. 1033–1040.

[22] J. Mairal, "Stochastic majorization-minimization algorithms for large-scale optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2283–2291.

[23] K. L. Clarkson, "Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm," *ACM Trans. Algorithms*, vol. 6, no. 4, pp. 63:1–63:30, 2010.

[24] E. Hazan and S. Kale, "Projection-free online learning," in *Proc. Annu. Int. Conf. Mach. Learn.*, 2012, pp. 1843–1850.

[25] S. J. Reddi, S. Sra, B. Poczos, and A. Smola, "Stochastic frank-wolfe methods for nonconvex optimization," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2016, pp. 1244–1251.

[26] Z. Allen-Zhu, "Natasha 2: Faster non-convex optimization than SGD," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2680–2691.

[27] Z. Allen-Zhu and Y. Li, "Neon2: Finding local minima via first-order oracles," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 3716–3726.

[28] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to escape saddle points efficiently," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1724–1732.

[29] Z. Allen-Zhu, "How to make the gradients small stochastically: Even faster convex and nonconvex SGD," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1157–1167.

[30] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[31] Y. W. Teh, D. Newman, and M. Welling, "A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1353–1360.

[32] M. Hoffman, D. M. Blei, and D. M. Mimno, "Sparse stochastic inference for latent Dirichlet allocation," in *Proc. 29th Int. Conf. Mach. Learn. (ICML)*, 2012, pp. 1599–1606.

[33] B. Dai, N. He, H. Dai, and L. Song, "Provable Bayesian inference via particle mirror descent," in *Artificial Intelligence and Statistics*, 2016, pp. 985–994.

[34] U. Simsekli, R. Badeau, T. Cemgil, and G. Richard, "Stochastic quasi-Newton langevin Monte Carlo," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 2–27.

[35] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 116–142, Jan. 2004.

[36] B. Luo, Y. Feng, Z. Wang, Z. Zhu, S. Huang, R. Yan, and D. Zhao, "Learning with noise: Enhance distantly supervised relation extraction with dynamic transition matrix," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, 2017, pp. 430–439.

[37] L. Bottou, "Online learning and stochastic approximations," *On-line Learn. Neural Netw.*, vol. 17, no. 9, p. 142, 1998.

[38] L. Bottou and V. Vapnik, "Local learning algorithms," *Neural Comput.*, vol. 4, no. 6, pp. 888–900, Nov. 1992.

[39] K. Than and T. Doan, "Guaranteed inference in topic models," 2015, *arXiv:1512.03308*. [Online]. Available: http://arxiv.org/abs/1512.03308

[40] H. Tuy, "Motivation and overview," in *Convex Analysis and Global Optimization*. Springer, 2016, pp. 127–149.

[41] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[42] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012.

[43] A. B. Dieng, F. J. R. Ruiz, and D. M. Blei, "Topic modeling in embedding spaces," 2019, *arXiv:1907.04907*. [Online]. Available: http://arxiv.org/abs/1907.04907

[44] B. Liu, L. Liu, A. Tsykin, G. J. Goodall, J. E. Green, M. Zhu, C. H. Kim, and J. Li, "Identifying functional miRNA-mRNA regulatory modules with correspondence latent Dirichlet allocation," *Bioinformatics*, vol. 26, no. 24, pp. 3105–3111, Dec. 2010.

[45] J. K. Pritchard, M. Stephens, and P. Donnelly, "Inference of population structure using multilocus genotype data," *Genetics*, vol. 155, no. 2, p. 945, 2000.

[46] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum, "Optimizing semantic coherence in topic models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 262–272.

[47] L. Yao, D. Mimno, and A. McCallum, "Efficient methods for topic model inference on streaming document collections," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, 2009, pp. 937–946.

[48] J. Grimmer, "A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases," *Political Anal.*, vol. 18, no. 1, pp. 1–35, 2010.

[49] H. A. Schwartz, J. C. Eichstaedt, L. Dziurzynski, M. L. Kern, E. Blanco, M. Kosinski, D. Stillwell, M. E. Seligman, and L. H. Ungar, "Toward personality insights from language exploration in social media," in *Proc. AAAI Spring Symp., Analyzing Microtext*, 2013, pp. 1–8.

[50] I. Vayansky and S. A. P. Kumar, "A review of topic modeling methods," *Inf. Syst.*, vol. 94, Dec. 2020, Art. no. 101582.

[51] S. Arora, R. Ge, F. Koehler, T. Ma, and A. Moitra, "Provable algorithms for inference in topic models," in *Proc. 33nd Int. Conf. Mach. Learn., ICML*, New York, NY, USA, Jun. 2016, pp. 2859–2867.

[52] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1303–1347, 2013.

[53] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh, "On smoothing and inference for topic models," in *Proc. 21th Conf. Uncertainty Artif. Intell.*, 2009, pp. 27–34.

[54] I. Sato and H. Nakagawa, "Stochastic divergence minimization for online collapsed variational Bayes zero inference of latent Dirichlet allocation," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, 2015, pp. 1035–1044.

[55] J. Foulds, L. Boyles, C. DuBois, P. Smyth, and M. Welling, "Stochastic collapsed variational Bayesian inference for latent Dirichlet allocation," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, 2013, pp. 446–454.

[56] K. Mai, S. Mai, A. Nguyen, N. Van Linh, and K. Than, "Enabling hierarchical Dirichlet processes to work better for short texts at large scale," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Springer, 2016, pp. 431–442.

[57] J. Tang, M. Zhang, and Q. Mei, "One theme in all views: Modeling consensus topics in multiple contexts," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, 2013, pp. 5–13.

[58] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proc. Nat. Acad. Sci. USA*, vol. 101, pp. 5228–5235, Apr. 2004.

[59] N. Aletras and M. Stevenson, "Evaluating topic coherence using distributional semantics," in *Proc. 10th Int. Conf. Comput. Semantics (IWCS)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2013, pp. 13–22.

[60] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, 2011, pp. 448–456.

[61] P. K. Gopalan, L. Charlin, and D. Blei, "Content-based recommendations with Poisson factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3176–3184.

[62] W. Feller, "The general form of the so-called law of the iterated logarithm," *Trans. Amer. Math. Soc.*, vol. 54, no. 3, pp. 373–402, 1943.

[63] J. H. Lau, D. Newman, and T. Baldwin, "Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality," in *Proc. 14th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2014, pp. 530–539.

XUAN BUI received the B.Sc. degree in applied mathematics and computer science from the University of Science (HUS), Vietnam National University (VNU), Hanoi, Vietnam, in 2003, and the M.Sc. degree in information technology from Thai Nguyen University, Vietnam, in 2007. She is currently the Lecturer with the Faculty of Computer Science and Engineering, Thuyloi University (TLU), Vietnam. She is also a member of the Data Science Laboratory, Hanoi University of Science and Technology, Vietnam. Her research interests include non-convex optimization, probabilistic graphical models, Bayesian inference, and machine learning.

HIEU VU received the B.S. degree from the Hanoi University of Science and Technology (HUST), Vietnam, in 2019. He is currently a member of the VinAI Research Laboratory, VinGroup, Vietnam. His research interests include topic models, stochastic optimization, deep learning, and big data.

OANH NGUYEN received the M.S. degree from the Hanoi University of Science and Technology, Vietnam, in 2001, and the Ph.D. degree from Nancy 2 University, France, in 2010. She is currently a member of the Data Science Laboratory, Hanoi University of Science and Technology. Her recent research interests include image processing and computer vision. She is a member of the Vietnamese Association for Pattern Recognition.

KHOAT THAN received the Ph.D. degree from the Japan Advanced Institute of Science and Technology, in 2013. He is currently an Associate Professor with the Hanoi University of Science and Technology and a Visiting Scientist with VinAI Research. His recent research interests include representation learning, deep generative models, topic modeling, and continual learning. He joined the program committees of various leading international conferences, including ICML, NIPS, IJCAI, and ICLR.