Non-Simultaneous Sampling Deactivation during the Parameter Approximation of a Topic Model

Young-Seob Jeong, Sou-Young Jin and Ho-Jin Choi
Department of Computer Science, Korea Advanced Institute of Science and Technology,
373-1 Guseong-dong, Daehakro 291, Yuseong-gu, Daejeon 305-701, Republic of Korea
[e-mail: pinode, nikkijin, hojinc@kaist.ac.kr]
*Corresponding author: Ho-Jin Choi

Received September 6, 2012; revised December 10, 2012; accepted December 21, 2012; published January 29, 2013

Abstract

Since Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA) were introduced, many revised or extended topic models have appeared. Due to the intractable likelihood of these models, training any topic model requires to use some approximation algorithm such as variational approximation, Laplace approximation, or Markov chain Monte Carlo (MCMC). Although these approximation algorithms perform well, training a topic model is still computationally expensive given the large amount of data it requires. In this paper, we propose a new method, called non-simultaneous sampling deactivation, for efficient approximation of parameters in a topic model. While each random variable is normally sampled or obtained by a single predefined burn-in period in the traditional approximation algorithms, our new method is based on the observation that the random variable nodes in one topic model have all different periods of convergence. During the iterative approximation process, the proposed method allows each random variable node to be terminated or deactivated when it is converged. Therefore, compared to the traditional approximation ways in which usually every node is deactivated concurrently, the proposed method achieves the inference efficiency in terms of time and memory. We do not propose a new approximation algorithm, but a new process applicable to the existing approximation algorithms. Through experiments, we show the time and memory efficiency of the method, and discuss about the tradeoff between the efficiency of the approximation process and the parameter consistency.

Keywords: Topic mining, unsupervised learning, efficient parameter approximation

A preliminary version of this paper was presented at the International Conference on Smart Convergence Technologies and Applications (SCTA) on August 7-9, 2012 in Gwangju, Korea. This version includes additional models, concrete analysis and implementation results. This work was supported by the National Research Foundation (NRF) grant (No. 2012-0001001) of Ministry of Education, Science and Technology (MEST) of Korea.

http://dx.doi.org/10.3837/tiis.2013.01.006

1. Introduction

Since Probabilistic Latent Semantic Analysis (PLSA) [1] and Latent Dirichlet Allocation (LDA) [2] were introduced about a decade ago, many revised or extended topic models have appeared. Topic mining is an unsupervised clustering technique that analyzes various types of data typically with bayesian models known as topic models. Each topic model has a unique structure which reflects the hypothesis of its data, hence, the performance of different models must be different - even with the same data. The result of topic mining contains topics and parameters. Each topic is a cluster whose items have different weights summed to 1. The items with greater weights co-occur many more times than the items with smaller weights. This implies that the topics will have reduced dimensions based on the co-occurrence of all of the items such that they represent the latent knowledge of the data. Moreover, all clusters have an identical list of items, although the weight distributions are different. In this sense, topic mining can be said to be a soft unsupervised clustering technique. Typically, the parameters of a topic model are distributions (i.e., topic distributions, word distributions), and they can be obtained by approximation algorithms.

Through various topic models, latent or invisible knowledge of various types of data can be captured successfully. Tag Topic Model (TTM) [3] uses tags chosen by the blogger for blog mining. It also employs a nonlinear dimensionality reduction technique known as Isomap to visualize similarity matrices for tags and content. The Dynamic Topic Model (DTM) [4] captures topic flows or sequences of topics over time. When each document has its time information, topics can then be obtained for each time span. The model hypothesizes that current topics are influenced by previous topics, and this is modeled with a normal distribution. Several topic models, including CorrLDA2, have been proposed to acquire topics from the perspectives of entities [5]. In particular, CorrLDA2 considers a list of entities to be a topic, creating two types of topics: word topics and entity topics. For scene recognition, the Context-Aware Topic Model (CATM) [6] models jointly both global features and local spatial features. It hypothesizes that each image region has a topic drawn from the image and a topic associated with the image label. To obtain topic flows in a document from the perspectives of entities or entity groups, the Sequential Entity Group Topic Model (SEGTM) [7] hypothesizes that each chapter is influenced by its previous chapter. This is modeled using a two-parameter Poisson-Dirichlet Process (PDP).

In all of the above models, the likelihood values of topic models are usually computationally intractable due to the coupling of variable nodes. Thus, training of a topic model requires the use of an approximation algorithm such as variational approximation, Laplace approximation, or Markov chain Monte Carlo (MCMC). In particular, collapsed Gibbs sampling, which is a form of the MCMC method, is known to be a powerful iterative algorithm that is easy to implement [8]. Although approximation algorithms perform well, training of topic models is still computationally expensive given the large amount of data it requires. To solve this problem, many studies have appeared to make the approximation process more efficient in terms of time and memory usage.

In this paper, we propose a new method, called non-simultaneous sampling deactivation, for efficient approximation of parameters in a topic model. In our approach, we do not propose a new approximation algorithm, but a new process which can be applied to existing approximation algorithms. While each random variable is sampled or obtained by a single predefined burn-in period in the traditional training process, our new method is based on the

observation that the random variable nodes in one topic model have all different periods of convergence. In the experiments, the new method will be evaluated under two criteria: (1) time and memory efficiency, and (2) parameter consistency. Through the experimental results, we will show and discuss the usefulness and drawbacks of the proposed method.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 presents the details of the proposed method. Section 4 shows the experimental results in detail. Section 5 concludes the paper.

2. Related Work

As for making approximation processes efficient in terms of time and memory, there have been many studies on not only topic models but also other probabilistic models. For efficient computation of k-means solutions, acceleration algorithms have been proposed [9][10]. The expectation-maximization of Gaussian mixtures is approximated and the quality of the approximation is controlled by regulating the parameter values [11]. To approximate the probabilities of Gaussian mixtures, samples are drawn from products of Gaussian mixture distributions in a branch-and-bound manner [12].

For efficient approximations of topic models, there also have been several studies. Since the parameters are distributions, it is necessary to approximate normalizing constant or denominator value. To obtain denominator value efficiently, an iterative method to update the summation of unnormalized values is proposed [13]. On the other hand, SparseLDA and classification based approximation are proposed [14]. These methods optimize the approximation process by avoiding redundant computations of constant values and by using a non-iterative process based on classification techniques. Several studies on parallel or distributed approximation also have been proposed. For the parallel expectation-maximization of PLSA, the E step and M step processes are combined and the computations are distributed over multiple processors using shared or distributed memory [15]. Approximate Distributed LDA (AD-LDA) and Hierarchical Distributed LDA (HD-LDA) are proposed for distributed parameter approximation for LDA [16]. AD-LDA simply performs an inference of LDA for each processor independently, while HD-LDA directly models the parallel process using LDA mixture components. Although the parallel approximation process on a multi-processor system usually involves communication between the processors, Dirichlet Compound Multinomial LDA (DCM-LDA) allows the user to avoid this type of communication [17].

For the issue of efficient parameter approximation of topic models, we take a novel approach that existing approaches have not considered. Our idea is based on the observation that random variable nodes of a topic model have different convergence timings. In this paper, we investigate this idea by defining a new process in which the nodes can be deactivated at different steps.

3. Non-Simultaneous Sampling Deactivation

This section presents first the terminology used in this paper to avoid confusion, together with a review of the collapsed Gibbs sampling of topic models as a basis to demonstrate the usefulness of the new method. In addition, we briefly review two topic models, the Author Topic Model (ATM) [18] and the Aspect Sentiment Unification Model (ASUM) [19], which will be used in the experiments. Thereafter, we propose our method, non-simultaneous

sampling deactivation, in detail and discuss its usefulness, in particular, how it can be applied to collapsed Gibbs sampling.

3.1 Terminology

As we will discuss topic models and parameter approximation process, it is necessary to define the terminology used to avoid confusion when we describe the new method. For this purpose, we will use the graphical representation of LDA in **Fig. 1**. The symbol T represents the total number of topics, D the total number of documents, and N the number of words for each document.

- **Parameter node** : An objective node to approximate in the graphical representation. For example, in **Fig. 1**, the two nodes θ and Φ are parameter nodes of the LDA model. The number of parameters of node θ is D. In other words, node θ can be seen as a vector $\{\theta_1, \theta_2, \ldots, \theta_D\}$, where θ_d is a parameter of document d. Similarly, the number of parameters of node Φ is T.
- **Hyper parameter node** : A regulation node which is to be set manually by a human user. For instance, the two nodes α and β in **Fig. 1** are hyper-parameter nodes.
- **Random variable node** : A node whose value is randomly sampled under certain criteria or distributions. In **Fig. 1**, node z is a random variable node to sample. The total number of random variables of node z is $D \times N$.
- **Observation node** : A node w that is to be observed from the data. Observation node is shaded to indicate the implicit nature. When we apply the LDA model to corpora, the number of observations of node w will be the total number of words.

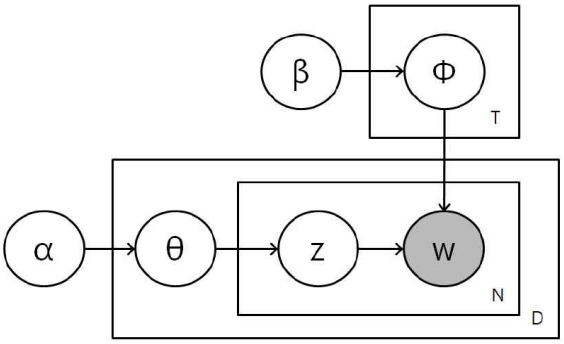


Fig. 1. A graphical representation of Latent Dirichlet Allocation (LDA) [2].

3.2 Collapsed Gibbs Sampling

Since LDA was introduced, topic models are generally assumed to have hyper-parameters, like LDA. The hyper-parameters can be seen as prior knowledge of other parameters, and therefore represent a measurement of distributions. To simplify the form of mathematical likelihood, many topic models employ conjugate priors which make the posterior distribution equal to the prior distribution. For example, as shown in **Fig. 1**, LDA has two conjugate prior nodes α and β. They can be seen as prior knowledge of θ and Φ, respectively. If we consider a

conjugate pair which consists of the Dirichlet distribution p($\theta|\alpha$) and the multinomial distribution p($z|\theta$), the posterior distribution p($\theta|z$) takes the form of a Dirichlet distribution. Although topic models employ conjugate pairs, they are still computationally expensive because it is necessary to obtain the integral in all cases of distributions. Moreover, coupling between the variables of a topic model makes it impossible to obtain the exact likelihood [2].

There are several algorithms to approximate an intractable likelihood, such as variational approximation, Laplace approximation, and the Markov chain Monte Carlo (MCMC) method. In this paper, we chose to use collapsed Gibbs sampling, which is a MCMC method, because it is easy to implement and is powerful [8]. Gibbs sampling, not collapsed, is basically an iterative random process that updates a latent value of each item by sampling it based on distributions over observations. Thus, Gibbs sampling makes large proportions larger and small ones smaller. In topic mining, it gathers co-occurring observations for each topic. Specifically, when one or more items are marginalized out, it is termed collapsed Gibbs sampling. When we apply the collapsed Gibbs sampling to the classical LDA, it marginalizes out θ and Φ, so that we can deal with them exactly. The conditional probability of topic z_{di} of i-th word in document d will be:

$$p(z_{di} = t \mid \boldsymbol{z}^{-di}, \boldsymbol{w}, \alpha, \beta) = \frac{(C_{i,t}^{-di} + \beta)}{(\sum_j C_{j,t}^{-di} + W\beta)} \frac{(C_{d,t}^{-di} + \alpha)}{(\sum_k C_{d,k}^{-di} + T\alpha)} \quad (1)$$

where $C_{(\cdot)}^{-di}$ is a count that does not include the topic assignment of the i-th word in the document d. In the numerators of the right side, $C_{i,t}^{-di}$ is the frequency of vocabulary i assigned to the topic t without including the i-th word in the document d, and $C_{d,t}^{-di}$ is the number of the assignments of the topic t in the document d without including the i-th word in the document d. The denominators of the right side are just for normalization. For each iteration, every topic assignment z_{di} is firstly removed, then it is resampled using the conditional probability (1) given the topic sequence z^{-di} and the word sequence w. We expect that it will be converged after enough iterations because the dependence of z_{di} on any particular other variable for each iteration is very weak [8][21]. When every random variable nodes of a topic model are converged at step X, then the period between a start point and the step X is called as a burn-in period. Thus, after the burn-in period, it does not have to continue the inference process. The burn-in period can be obtained from observations of the inference process. The hyper-parameters α and β are used as prior knowledge to smoothen the conditional probability. This means that the convergence will be faster if the prior knowledge is consistent to the dataset, but it will be slower otherwise. Further, with a larger dataset, the influence of the hyper-parameters will be decreased because the frequency values are relatively greater than the hyper-parameters.

Note that it is not our purpose to compare our method with collapsed Gibbs sampling. The proposed method is a new process applicable to existing approximation algorithms. We chose collapsed Gibbs sampling to demonstrate the efficiency of the new method. In the experiments, we will show the usefulness of our method using two cases: (1) collapsed Gibbs sampling with the new method, and (2) collapsed Gibbs sampling without the new method.

3.3 Two Example Topic Models

In this subsection, we briefly introduce two topic models which are chosen to demonstrate the performance of the new method to be presented in the following subsection. They are the Author Topic Model (ATM) [18] and the Aspect Sentiment Unification Model (ASUM) [19]. Our method assumes that the random variable nodes of a topic model have different

convergence timings. Accordingly, it is the basic requirement for applying our method that the topic model should have two or more random variable nodes. For this reason, we chose the well-known model, ATM, and one of more recent models, ASUM, to which we apply our proposed method.

3.3.1 Author Topic Model (ATM)

ATM captures a topic distribution for each author. The model uses the hypothesis that each document is written by multiple authors who have an interest in different areas or topics. The model also hypothesizes that, for each document, every corresponding author is equally dominant. A graphical representation of ATM is depicted in **Fig. 2**. The boundary D is the number of documents, and N is the number of words for each document. The model has two hyper-parameter nodes α and β. Parameter node θ is the topic distribution, and the total number of authors is A. Parameter node Φ is the word distribution of a topic, and the number of topics is T. Random variable node x is uniformly sampled from the observation node a_d, which is a list of authors of the document. Random variable node z is sampled under corresponding distribution θ. Therefore, for each observation w, there are two corresponding random variables x_w and z_w. The value of the random variable x_w will be one of the authors, while the value of the random variable z_w will be one of the topics.

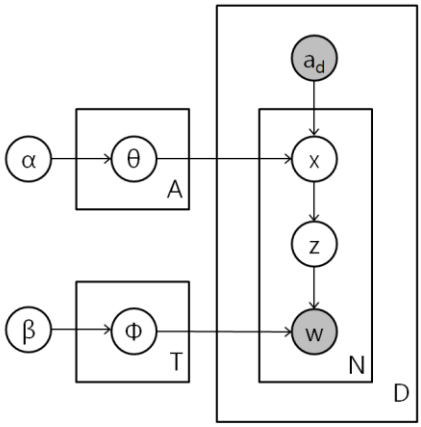


Fig. 2. A graphical representation of Author Topic Model (ATM) [18].

ASUM works on the hypothesis that each document has topic distributions for each sentiment. To obtain the topics of sentiments, it employs a predefined list of sentiment seed words. For example, the term ‘good’ or ‘great’ implies positive sentiment, while ‘bad’ or ‘worse’ implies negative sentiment. Based on the list of sentiment words, the values of the hyper-parameters of the model are asymmetrically initialized. For example, if there are a positive sentiment and a negative sentiment, then there will be two hyper-parameters β_{pos} and β_{neg}, where $\beta_a = \{\beta_{a,1}, \beta_{a,2}, \ldots, \beta_{a,i}, \ldots, \beta_{a,V}\}$, and V is the number of unique words. If we want to force the term ‘bad’ to do not appear in the positive sentences, then we just initialize $\beta_{pos,i} = 0$ where i is the index of the term ‘bad’ among the V unique words. A graphical representation of ASUM is depicted in **Fig. 3**. The boundary D is the number of documents, M is the number of sentences for each document, and N is the number of words for each sentence. The model has three hyper-parameter nodes α, β, and γ. Parameter node θ is the topic distribution, and the total number of sentiments is S. Thus, each document has a topic distribution for each sentiment.

Parameter node Φ is the word distribution of a topic. There are T topics for each sentiment. Therefore, the total number of topics is $T \times S$. Each document has one parameter node π which represents the sentiment distribution. The random variable node s is sampled for each sentence under the corresponding sentiment distribution, and the random variable node z is sampled for each sentence under the corresponding topic distribution θ. Therefore, for each sentence m, there are two corresponding random variables s_m and z_m. The value of s_m will be one of the sentiments, while the value of z_m will be one of the topics.

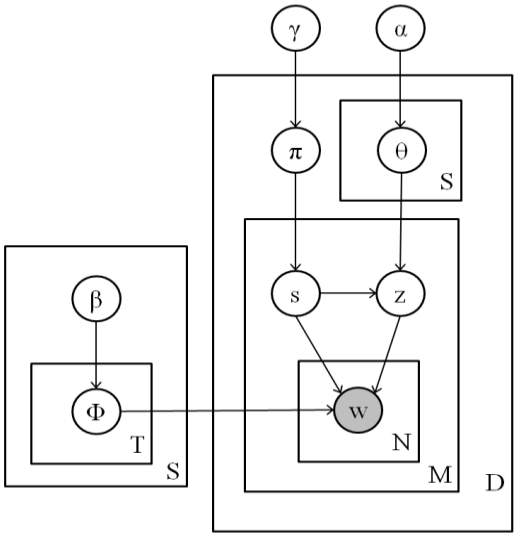


Fig. 3. A graphical representation of Aspect Sentiment Unification Model (ASUM) [19].

3.4 Non-Simultaneous Sampling Deactivation

Our proposal is not about developing a new approximation algorithm but about defining a new process which can be applied to existing approximation algorithms for topic models. In this paper, we choose the collapsed Gibbs sampling method as the approximation algorithm to which we apply the new process.

During the parameter approximation process using collapsed Gibbs sampling, random variables of a topic model are sampled at every step, and the process continues by a predefined burn-in period. In other words, the process treats all random variable nodes as if they all have a single period for convergence. However, we observed that different random variable nodes have different convergence timings. Therefore, the process does not have to use a single burn-in period for all the nodes. In **Fig. 4**, a sample observation obtained from ATM with 20 topics and 980 authors is depicted, in which each random variable node is converged at different iteration step. For instance, the convergence timing of node x appears to be around the 400th step, while the convergence timing of node z is around the 800th step. According to the traditional process, these two nodes will be deactivated concurrently around the 800th step. If these two nodes were deactivated separately, i.e., non-simultaneously, the entire approximation process would have become more efficient in terms of time and memory. This strategy is implemented in this paper. Further, we have also observed a simple rule that a random variable node converges faster than its following node. For example, in **Fig. 2**, node x must be converged faster than node z. The convergence timing gap is caused by two reasons. First, node x is in front of node z such that node z will never converge until node x is converged,

as node z is dependent on node x. Second, node x has a smaller dimension than node z. It is clear that the number of authors for each document is smaller than the total number of topics. A node with smaller dimension is more likely to be converged faster because it is based on random sampling process. For example, if a variable can have one of the K distinct states, then the state of the variable will be changed with the probability (K-1)/K if it is on the uniform distribution. This implies that it will be more likely that the state of variable is changed with a larger value of K. Moreover, even if it is not in uniform distribution, it also will be more likely that the state of the variable is changed with a larger value of K, because it is based on the random sampling process.

It is important that the dependency, which is the first reason, is a primary reason, and the size of dimension just makes the convergence gap larger. In other words, if the node z is dependent on the node x, then node z will never be converged until the node x is not converged even if the dimension of the node x is larger than that of the node z. Similarly, in **Fig. 3**, node *s* converges faster than node *z* for the same reasons.

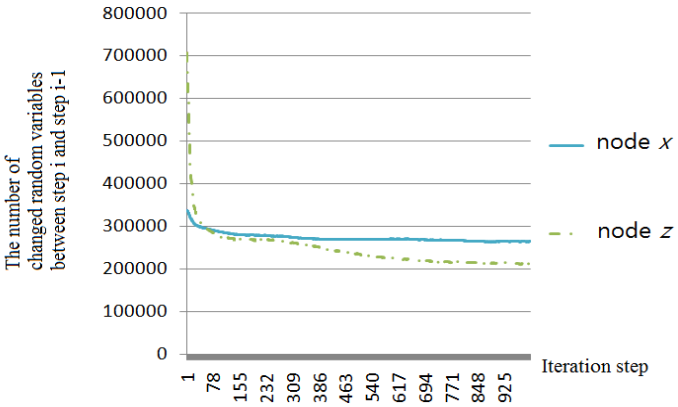


Fig. 4. An observation from ATM with 20 topics and 980 authors. The horizontal axis is the iteration step, and the vertical axis is the number of changed random variables between step i and step i-1.

Note that we do not directly consider the convergence timings of parameter nodes but that we do consider the convergence timings of random variable nodes, as we observed that random variable nodes are usually connected in order. Therefore, we can use a simple rule. In contrast, parameter nodes are not typically connected in order. For instance, in **Fig. 2**, the two parameter nodes θ and Φ are not connected in order. Therefore, whether they have sequential or different convergence timings is not guaranteed. Thus, the new method allows each random variable to be deactivated at a different step. To be specific, if we call a random variable node which is not converged an active node, every active node is checked regardless of whether it is converged or not for each approximation step. If the active node *r* is converged, then the sampling of node *r* is stopped for the remaining steps. The new method does not simultaneously deactivate the random variable nodes. Hence, it is referred to as non-simultaneous sampling deactivation. A formal algorithm of the method is described below.

Algorithm – Non-simultaneous sampling deactivation with collapsed Gibbs sampling

Input: (1) data, (2) a topic model with K random variable nodes,
(3) burn-in period p, (4) regulation parameter λ
Initilization: (1) For every k-th random variable node, the status S_k = active.
(2) The number of active nodes $K_a = K$.
(3) The values of all random variables are initialized (at random).
(4) The values of all regulation parameters (λ's) are set by manually.
(5) The hyper parameters are initialized.
for each step i
If ($i > p$) or ($K_a == 0$), then { break }.

for each node r *of* K_a *active nodes*
n_r = *the number of random variables of the node r.*
Sample new values of the n_r *variables.*
end

Update all the parameter values.

for each node r *of* K_a *active nodes*
r_{i-1} = *values of node r at step i-1. (i.e.,* $[r_{i-1,1}, r_{i-1,2}, \ldots, r_{i-1,n_r}]$ *)*
r_i = *values of the node r at step i. (i.e.,* $[r_{i,1}, r_{i,2}, \ldots, r_{i,n_r}]$ *)*
v^r_i = *the number of different or changed items between* r_{i-1} *and* r_i.
v^r_{i-1} = *the number of different or changed items between* r_{i-2} *and* r_{i-1}.
$c_r = | v^r_i - v^r_{i-1} |$.
λ_r = *a threshold value for the node r.*
If (c_r / n_r) $< \lambda_r$, *then*
{
S_r = *deactivated.*
$K_a = K_a - 1$.
}.
end
end

The burn-in period p is usually determined by considering the size of the data, the number of random variables, and the dimensions of random variables manually. The regulation parameter λ is a vector, each item of which is the threshold value for the convergence timing of each random variable node. Thus, the vector controls the deactivation timing of every random variable node. The threshold values of λ can be set manually by considering three factors: (1) The dimensions of the nodes, (2) The values of the hyper-parameters, (3) The trade-off between the inference efficiency and the parameter consistency. The first factor, the dimension of a node, means the number of distinct states of the node. With a larger dimension of a node, it is more likely the state of the variable is changed because the inference is based on random sampling process. If we increase the dimension of a particular node, then its state would become more likely to change during the iterations, which causes its convergence timing to be delayed. Since the parameter λ is a vector of threshold values for all random variable nodes, it should be better to set larger threshold values to the nodes which have larger

dimensions. The second factor, the values of the hyper-parameter, influences the speed of convergence. This phenomenon, however, will not be a dominant factor for a large dataset, as discussed in the section 3.2. If we set a particular threshold values to λ by considering the first two factors, then we can revise the parameter λ to control the trade-off between the inference efficiency and the parameter consistency. The inference efficiency is about the spent time and the memory usage during the inference, and the parameter consistency is about how much the parameter values are acceptable or comprehensible by human when we apply the proposed method. In other words, when we apply the new method, the parameter values must be different from the case that the new method is not applied to. Therefore, it is necessary to investigate how much the parameter values will be different when we apply the new method. The more details and its experimental results will be discussed in the experiment section.

To explain the algorithm effectively, we assume that the algorithm is applied to ATM. As the model is designed to obtain the topic distributions of authors, data must be in the form of documents in which each document is written by multiple authors. Let us assume that there are A unique authors, D documents, and N_d words in each document d. At the initialization step, the two random variable nodes x and z are initialized as active. Further, all random variables are randomly initialized. As there are multiple random variables for each random variable node, the total number of random variables to initialize must be a large number. For example, as shown in **Fig. 2**, the number of random variables of node x in the document d is N_d, while the number of random variables of node z is also N_d. Therefore, $2\times N\times D$ random variables in total are initialized, where N is the average number of words of each document. Based on the initialized random variables, the values of the two parameters θ and Φ are computed. As the last process of the initialization step, the hyper-parameters α and β are usually initialized symmetrically. Symmetric initialization means that all items of a vector have the same values. For instance, $\alpha = \{0.1, 0.1, \ldots, 0.1\}$.

When the iterative approximation process begins, it makes a decision as to whether the process must be terminated or not for each step i. If it reaches the burn-in period p or if there is no active node, the process is terminated. Thus, if the method is applied to a model which has only one random variable node, then the deactivation step of the node will be identical to the burn-in period p. This implies that the method is useful only to the topic models which have two or more random variable nodes. If the process of i-th step is determined to continue, new values of only the active nodes are sampled. For example, in the document d, assuming that the number of random variables of the node x is A_d, and the number of random variables of node z is N_d. When the node x is active, there are $A_d\times N_d$ random variables to sample in the document d. In contrast, when the node x is deactivated, then there are only N_d random variables to sample. This implies that the improvement of inference efficiency with the method will be larger when the node z has more random variables to sample, because $A_d\times N_d$ will be much larger than N_d with the larger N_d. As the new values of the random variables must influence the two parameter nodes θ and Φ, it is necessary to update the parameter values. Thereafter, each active random variable node is checked as to whether or not it must be deactivated. Assuming that the node z is being checked at step i, then $n_z = N\times D$ and $z_i = \{ z_{i,1}, \ldots, z_{i,k}, \ldots, z_{i,n_z} \}$, where $z_{i,k}$ is a value of the k-th random variable of node z. The dimension of $z_{i,k}$ is the number of total topics, and the number of different items v^z_i between z_i and z_{i-1} will be $1 \le v^z_i \le n_z$. Similarly, $1 \le c_z \le n_z$ such that $0 \le (c_z / n_z) \le 1$, where n_z is a normalizing constant. Therefore, the threshold λ_z, which is an item of the regulation parameter λ, is an important real value that controls the deactivation timing of the node z.

To summarize, the most important task of deactivation is to determine whether a particular node must be deactivated or not, for each iteration. The new method has a trade-off between

the inference efficiency and the parameter consistency. In terms of the parameter consistency, it is difficult to automatically make a judgment whether the parameters are comprehensible by human or not. Therefore, the new method allows people to control the level of decision by the regulation parameter λ. The inference efficiency and the parameter consistency will be explained in section 4.

4. Experiment

In this section, we first describe the datasets for the experiment and then present the computer and parameter settings. Next, we describe the evaluation process to demonstrate the performance of the proposed method on two criteria: (1) time and memory efficiency, (2) parameter consistency. As described earlier, we use the two models ATM and ASUM and choose the collapsed Gibbs sampling algorithm to apply the new method. To obtain reliable results, we perform 5 times for each criterion, and obtain average results.

4.1 Dataset

In the experiment, we evaluated the new method on two datasets: (1) the New Testament of the Bible, and (2) NIPS papers. The former was used for ASUM, while the latter was used for ATM. The reason behind this biased usage of datasets is the structural differences of the models. ATM has the random variable node x, which is uniformly sampled from a list of corresponding authors. This implies that ATM is designed for documents in which each document has more than one author. If each document has only one author, then ATM must give a result similar to LDA. Each of the NIPS papers usually has multiple authors; thus, the NIPS papers are clearly a suitable dataset for ATM. In contrast, each chapter of the New Testament of the Bible usually has a single author. ASUM has the random variable node s, which is sampled under a corresponding sentiment distribution. This implies that ASUM is designed for documents in which each document has many sentiment terms, such as ‘good’, ‘excellent’, and ‘bad’. NIPS papers typically do not have sentiment terms because the papers are reasonable and not emotional. In contrast, the New Testament of the Bible has many sentiment terms. Thus, the New Testament of the Bible is clearly a suitable dataset for ASUM. Specifically, ASUM requires a list of sentiment seed words. The seed words are used for asymmetric initialization of hyper parameters to get topics for each sentiment. Therefore, we employed PARADIGM+ as used in earlier work [19], which has two sentiments, a positive sentiment and a negative sentiment. The New Testament of the Bible has 27 documents in total, from Matthew to The Book of Revelations. It has 10,005 sentences and 75,264 words. The dataset of the NIPS papers is composed of papers from five years (1987 to 1991) from NIPS conferences. It has a total of 573 documents, 83,939 sentences, 796,474 words, and 980 authors.

4.2 Computer and Parameter Setting

The objective of the new method is to make the parameter approximation process more efficient; thus, it is important to represent specific information of the machine. We used an Intel(R) Core(TM)2 CPUs running at 1.86GHz and 1.87GHz. The system had 2GB of RAM and runs the operating system Windows 7 Enterprise K. We implemented and evaluated the new method on the Java platform. Further, we initialized most of the prior parameters symmetrically, as the initial values do not have much of an effect on the result for a large dataset. We set $\alpha = 0.1$ and $\beta = 0.01$ for ATM, and set $\alpha = 0.1$ and $\gamma = 0.1$ for ASUM. Particularly, we set asymmetrically $\beta = 0.01$ for the same sentiment and $\beta = 0$ for the opposite

sentiment, as in earlier research [19] on ASUM. The regulation parameter λ was symmetrically initialized as 0.01. Based on several observations, we set 3,000 as the burn-in period for ATM, and 2,000 for ASUM.

4.3 Evaluation Process

As described earlier, there are two evaluation criteria: (1) time and memory efficiency, (2) parameter consistency. For each criterion, to emphasize the usefulness of the method, we compare performances between two cases: (1) the collapsed Gibbs sampling with the method, (2) the collapsed Gibbs sampling without the method.

Assuming that we perform the comparison on ATM, as depicted in **Fig. 5**, the approximation process is divided into two cases, and the cases are performed in parallel. The red line represents the first case, and the blue line does so for the second case. As node *x* is converged faster than node *z*, the approximation process is divided at *step A*, in which node *x* is deactivated. The approximation process of the first case is terminated at *step D*, at which point node *z* is deactivated. In contrast, the approximation process of the second case is terminated at *step B*, at which point both nodes are deactivated concurrently. Further, we continued the approximation of the second case to reach the 'golden state' at a certain *step C*. The 'golden state' is used to measure parameter consistency as the second criterion. For the experiment, the number of steps between *step B* and *step C* was set as 1,000. The reason for considering the step C as 'golden state' is that the period between start point and the *step B* is the burn-in period. Once the burn-in period has passed, all parameters of the model are converged. That is, we regard all values of the parameters obtained after the burn-in period as correct, because we do not propose a new approximation algorithm, but about improving the iterative process of existing algorithms. Therefore, parameters obtained at the step *B* must be similar or consistent with parameters of the step *C*, because both steps *B* and *C* are regarded as correct. The parameters obtained at the *step D* will be useless if they are too different from the parameters of 'golden state'. In other words, the parameter difference between *step B* and 'golden state' is used to measure how much the parameters of *step D* are different from the parameters of 'golden state'. As described in the subsection 4.2, we set the burn-in period *p* of ATM as 3,000, while we set the burn-in period *p* of ASUM as 2,000 based on several observations of parameter inference.

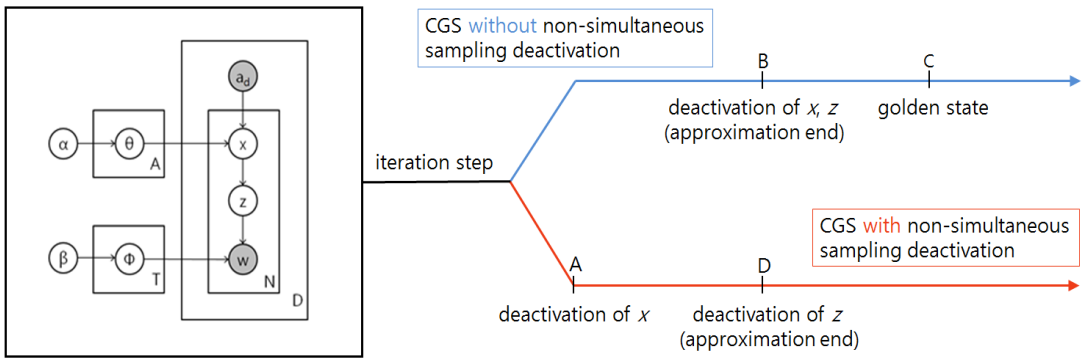


Fig. 5. A graphical representation of the evaluation process on ATM.

4.4 Time and Memory Efficiency

During the approximation process, we obtain the statistics on elapsed time and Java memory usage for the two cases of our experiment: (1) the collapsed Gibbs sampling using the proposed method, and (2) the collapsed Gibbs sampling without using the method. Elapsed time and memory usage for the first case are obtained at step D, while obtained at step B for the second case. **Fig. 6** shows the performance comparison in the elapsed time between ATM and ASUM, in which the case using our method clearly shows less elapsed time, because the process does not need to sample deactivated random variable nodes in each step. **Fig. 7** shows the performance comparison in the memory usage between the two models, and again, the case using our method shows less usage of memory. The performance gap between the two cases tends to become larger as the number of topics becomes larger because of the increase in the number of random variables. Also note that the improvement with ATM is larger than the improvement with ASUM. It is worth noting that we did not revise or improve the two models. We proposed a new method applicable to topic models which have two or more random variable nodes. Therefore, the improvement difference is caused by the difference in the dataset size (i.e., 796,474 words and 83,939 sentences in the NIPS paper dataset, compared to 75,264 words and 10,005 sentences in the New Testament Bible). With more words and sentences, there exist more random variables to sample, therefore, it will be more sensitive when a particular random variable node is deactivated. When node x of ATM is deactivated, for example, there exist 796,474 random variables for the node z to sample. In contrast, when node s of ASUM is deactivated, only 10,005 random variables exist to sample. As there are many random variables to sample with ATM than ASUM, the improvement with ATM is larger than the improvement with ASUM.

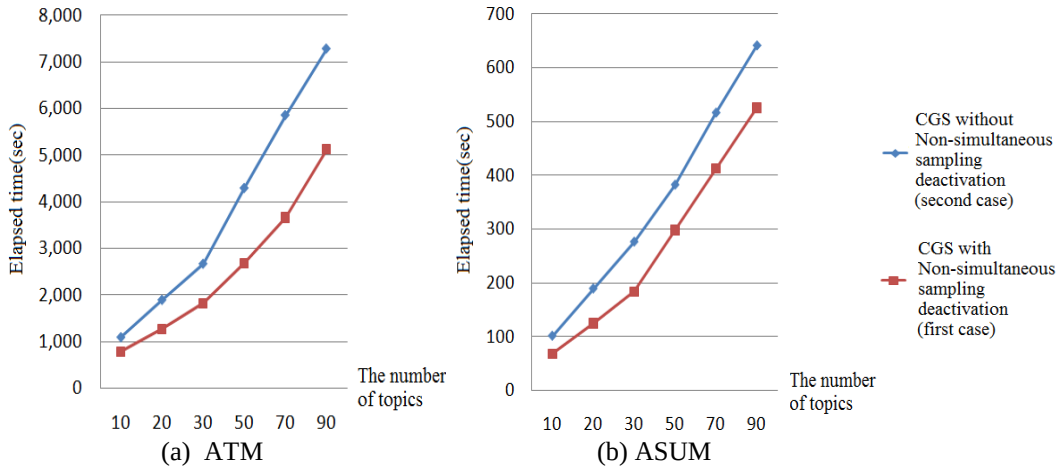


Fig. 6. Elapsed time of ATM and ASUM. The vertical axis means the elapsed time and the horizontal axis represents the number of topics.

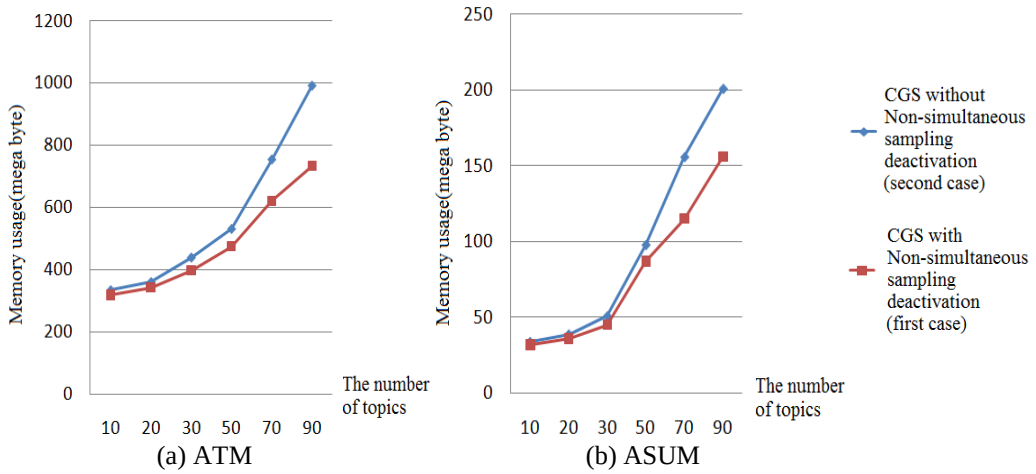


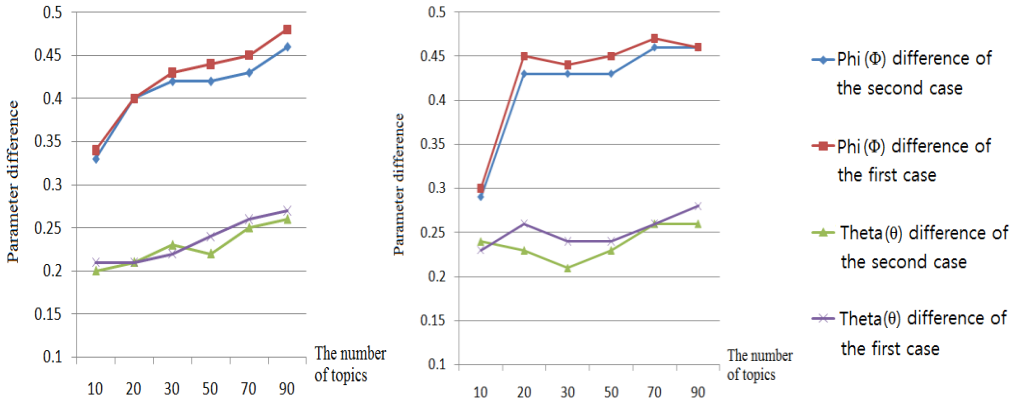
Fig. 7. Memory usage of ATM and ASUM. The vertical axis means memory usage and the horizontal axis is the number of topics and the vertical axis is the memory usage.

4.5 Parameter Consistency

The new method causes differences in the parameter values, as it deactivates converged random variables earlier than the other nodes. If the parameter difference is too great, the method becomes useless. Therefore, we need to investigate how large differences in parameter values arise when the new method is applied. The parameter values of the first case were obtained at *step D*, and the parameter differences were computed via a comparison with the parameter values obtained at the 'golden state', *step C*. To be specific, each parameter difference is computed by the Hellinger distance between *step D* and *step C*. Similarly, the parameter values of the second case were obtained at *step B*, and the parameter differences were computed by a comparison with the parameter values of the 'golden state'. It is worth noting that the parameter differences in the second case must be very small because the model is converged after *step B*.

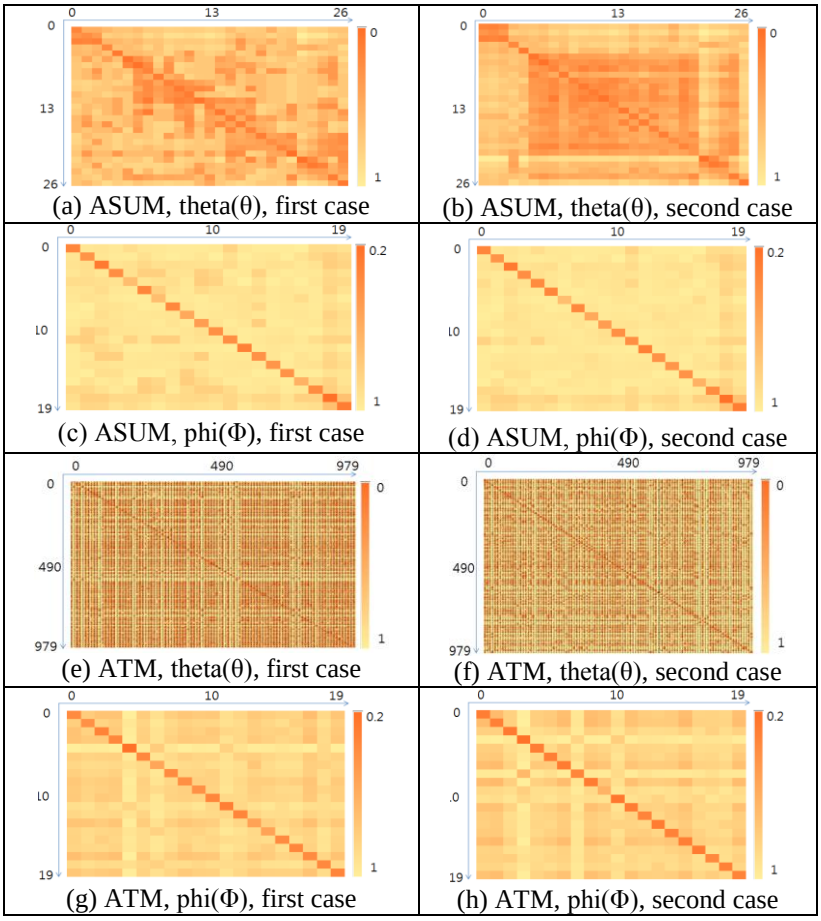
ATM has two parameter nodes, θ and Φ, while ASUM has three parameter nodes, θ, Φ, and π. For both of the models, we obtained the differences between the two parameters θ and Φ. In particular, as the dimension size of parameter π is 2, the difference for π was too small in both cases. In **Fig. 8**, the parameter differences of the models are depicted. The differences for π are not represented in the graph for the reason described above. In both models, the first case generally has slightly greater difference values than the second case. This implies that the new method leads to small differences in the parameter values, which in turn can lead to incorrect comprehension by humans. However, if the differences are small enough, a human user can comprehend the result without misunderstanding and the method makes the approximation process more efficient without a loss of information. To check whether or not this will lead to confusion, we depicted the parameter differences of each parameter in **Fig. 9**. Every matrix is a square because the number of columns or rows is a dimension of the parameter. For each cell, the Hellinger distance is computed by a comparison with the 'golden state', after which it has a bright color if its Hellinger distance value is large. Therefore, the diagonal dark cells show that the parameter is consistent. If we examine (c) and (d) in **Fig. 9**, similar levels of parameter differences are observed in both cases. This implies that the parameter Φ of the first case will be comprehensible without misunderstand. In (e), (f), (g), and (h), similar results can be observed. On the other hand, there is a notable gap in the confusion matrices of (a) and (b). The

parameter, however, is consistent because the diagonal cells have low difference values. Consequently, there is a tradeoff between the efficiency of the approximation process and the small parameter differences. Hence, the decision-making process is done by the human user.



(a) ATM (b) ASUM

Fig. 8. Parameter differences of ATM (a) and ASUM (b). The horizontal axis is the number of topics and the vertical axis represents the parameter difference.



(a) ASUM, theta(θ), first case (b) ASUM, theta(θ), second case

(c) ASUM, phi(Φ), first case (d) ASUM, phi(Φ), second case

(e) ATM, theta(θ), first case (f) ATM, theta(θ), second case

(g) ATM, phi(Φ), first case (h) ATM, phi(Φ), second case

Fig. 9. The confusion matrices representing each parameter difference on both models.

5. Conclusion

In this paper, we presented a new method to make approximation process of topic models more efficient. The proposed method is not about developing a new approximation algorithm but about defining a new process, based on the observation that within a topic model, different random variable nodes have different timings of convergence. The proposed deactivation method takes the strategy of non-simultaneous sampling, and allows converged random variables to be deactivated faster than the other nodes in each approximation step. Through the experiments, we showed the efficiency gain in time and memory, and uncovered the tradeoff between the efficiency of the approximation process and the parameter consistency. Although we chose the collapsed Gibbs sampling method to apply our approach in this paper, the proposed method is applicable to other approximation algorithms such as variational approximation [20]. As future work, we plan to apply the proposed method to other algorithms like variational approximation, and also plan to observe performance by varying the regulation parameter λ. In addition, we will investigate better ways to check timings for deactivation, which is very challenging because this type of evaluation relies highly on human comprehension of the parameters.

References

[1] Thomas Hofmann, "Probabilistic latent semantic analysis," in *Proc. of 15th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 289-296, July 30-August 1, 1999. Article (CrossRef Link)

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993-1022, March 2003.Article (CrossRef Link)

[3] Flora S. Tsai, "A tag-topic model for blog mining," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5330-5335, May 2011. Article (CrossRef Link)

[4] David M. Blei and John D. Lafferty, "Dynamic topic models," in *Proc. of 23rd International Conference on Machine Learning (ICML)*, pp. 113-120, June 25-29, 2006. Article (CrossRef Link)

[5] David Newman, Chaitanya Chemudugunta, and Padhraic Smyth, "Statistical entity-topic models," in *Proc. of 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 680-686, August 20-23, 2006. Article (CrossRef Link)

[6] Zhenxing Nu, Gang Hua, Xinbo Gao, and Qi Tain, "Context aware topic model for scene recognition," in *Proc. of 25th IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2743-2750, June 16-21, 2012. Article (CrossRef Link)

[7] Young-Seob Jeong and Ho-Jin Choi, "Sequential entity group topic model for getting topic flows of entity groups within one document," in *Proc. of 16th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, pp. 366-378, May 29-June 1, 2012. Article (CrossRef Link)

[8] Thomas L. Griffiths and Mark Steyvers, "Finding scientific topics," in *Proc. of the National Academy of Sciences of the United States of America (PNAS)*, pp.5228-5235, April 6, 2004. Article (CrossRef Link)

[9] Khaled Alsabti, Sanjay Ranka, and Vineet Singh, "An efficient K-means clustering algorithm," in *Proc. of 1st IPPS/SPDP Workshop on High Performance Data Mining*, March 30-April 3, 1998. Article (CrossRef Link)

[10] Dan Pelleg and Andrew Moore, "X-means: extending K-means with efficient estimation of the number of clusters," in *Proc. of 17th International Conference on Machine Learning (ICML)*, pp. 727-734, June 29-July 2 , 2000. Article (CrossRef Link)

[11] Andrew W. Moore, "Very fast EM-based mixture model clustering using multiresolution kd-trees," in *Proc. of 12th Conference on Advances in Neural Information Processing*

Systems(NIPS), pp. 543-549, November 29-December 3, 1998. Article (CrossRef Link)
[12] Alexander T. Ihler, Erik B. Sudderth, William T. Freeman, and Alan S. Willsky, "Efficient multiscale sampling from products of Gaussian mixtures," in *Proc. of 17th Conference on Advances in Neural Information Processing Systems (NIPS)*, December 9-11, 2003. Article (CrossRef Link)
[13] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling, "Fast collapsed gibbs sampling for latent dirichlet allocation," in *Proc. of 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pp. 569-577, August 24-27, 2008. Article (CrossRef Link)
[14] Limin Yao, David Mimno, and Andrew McCallum, "Efficient methods for topic model inference on streaming document collections," in *Proc. of 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pp. 937-946, June 28-July 1, 2009. Article (CrossRef Link)
[15] Raymond Wan, Vo Ngoc Anh, and Hiroshi Mamitsuka, "Efficient probabilistic latent semantic analysis through parallelization," in *Proc. of 5th Asia Information Retrieval Symposium on Information Retrieval Technology (AIRS)*, pp. 432-443, October 21-23, 2009. Article (CrossRef Link)
[16] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling, "Distributed inference for latent dirichlet allocation," in *Proc. of 21th Conference on Advances in Neural Information Processing Systems (NIPS)*, pp. 1081-1088, December 3-6, 2007. Article (CrossRef Link)
[17] David Mimno and Andrew McCallum, "Organizing the OCA: learning faceted subjects from a library of digital books," in *Proc. of 7th ACM/IEEE-CS joint conference on Digital libraries (JCDL)*, pp. 376-385, June 18-23, 2007. Article (CrossRef Link)
[18] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth, "The author-topic model for authors and documents," in *Proc. of 20th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 487-494, July 7-11, 2004. Article (CrossRef Link)
[19] Yohan Jo and Alice H. Oh, "Aspect and sentiment unification model for online review analysis," in *Proc. of 4th ACM international conference on Web search and data mining (WSDM)*, pp. 815-824, February 9-12, 2011. Article (CrossRef Link)
[20] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, no. 2, pp. 183-233, November, 1999. Article (CrossRef Link)
[21] Yee Whye Teh, David Newman, and Max Welling, "A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation," Advances in Neural Information Processing Systems (NIPS) 2006, December 4-9, 2006. Article (CrossRef Link)

Young-Seob Jeong is currently a PhD student in the Dept. of Computer Science at KAIST. He was a member of ACM-ICPC of Hanyang university, from 2005 to 2009. He also was a head of the Hanyang ACM-ICPC on 2007 and guided two teams to reach Asia Final programming contest. His current research interests include topic modeling, deep learning, and action prediction based on various sensor data.

Sou-Young Jin is currently a PhD student in the Dept. of Computer Science at KAIST. She received her MS from KAIST in 2012 and her BS from Dongguk University in 2010. Her research interests include vision-based activity recognition, data mining, and object recognition.

Ho-Jin Choi is currently an associate professor in the Dept. of Computer Science at KAIST. In 1982, he received a BS in Computer Engineering from Seoul National University, Korea, in 1985, an MSc in Computing Software and Systems Design from Newcastle University, UK, and in 1995, a PhD in Artificial Intelligence from Imperial College, London, UK. From 1982 to 1989, he worked for DACOM, Korea, and between 1995 and 1996, worked as a post-doctoral researcher at Imperial College. From 1997 to 2002, he served as a faculty member at Korea Aerospace University, Korea, then from 2002 to 2009 at Information and Communications University (ICU), Korea, and since 2009 he has been with the Dept. of Computer Science at KAIST. Between 2002 and 2003, he visited Carnegie Mellon University (CMU), Pittsburgh, USA, and has been serving as an adjunct professor of CMU for the program of Master of Software Engineering (MSE). Between 2006 and 2008, he served as the Director of Institute for IT Gifted Youth at ICU. Since 2010, he has been participating in the Systems Biomedical Informatics National Core Research Center at the Medical School of Seoul National University. Currently, he serves as a member of the boards of directors for the Software Engineering Society of Korea, for the Computational Intelligence Society of Korea, and for Korean Society of Medical Informatics. His current research interests include artificial intelligence, data mining, software engineering, and biomedical informatics.