Topic Model for Graph Mining

Junyu Xuan, Jie Lu, *Senior Member, IEEE* Guangquan Zhang, and Xiangfeng Luo, *Member, IEEE*

***Abstract*—Graph mining has been a popular research area because of its numerous application scenarios. Many unstructured and structured data can be represented as graphs, such as, documents, chemical molecular structures, and images. However, an issue in relation to current research on graphs is that they cannot adequately discover the topics hidden in graph-structured data which can be beneficial for both the unsupervised learning and supervised learning of the graphs. Although topic models have proved to be very successful in discovering latent topics, the standard topic models cannot be directly applied to graph-structured data due to the 'bag-of-word' assumption. In this paper, an innovative Graph Topic Model (GTM) is proposed to address this issue, which uses Bernoulli distributions to model the edges between nodes in a graph. It can, therefore, make the edges in a graph contribute to latent topic discovery and further improve the accuracy of the supervised and unsupervised learning of graphs. The experimental results on two different types of graph datasets show that the proposed GTM outperforms the Latent Dirichlet Allocation on classification by using the unveiled topics of these two models to represent graphs.**

***Index Terms*—Graph mining, Topic model, Latent Dirichlet Allocation**

I. Introduction

GRAPH is a structure of a set of nodes where some pairs of nodes are connected by links. Many unstructured and structured data can be represented as graphs. The research about this graph structured data belongs to the graph mining area [1]. The motivation for graph mining is that the edges (formed structures) will contribute to the classification or clustering of the data as compared to instance mining which only considers nodes [1], [2]. In text mining, a document, for example, [3], [4] is composed of some words as nodes and word relations, which can be co-occurrence relations, association relations, or other semantic relations. The classification of these document graphs can improve the accuracy of document retrieval with word vectors as representations. To provide another example, a chemical molecular structure can be represented by a graph with basic elements as nodes and chemical bonds as edges. The classification of these chemical molecular graphs can help to label molecular structures which is normally a very difficult and time-consuming process. Moreover, products, services, website retrieval and many real-world tasks can benefit from such graph mining. However, the existing graph mining algorithms are mainly based on the frequent subgraph representation [5]–[7] which transforms a graph into an instance where each subgraph is a dimension and then the existing instance-based machine learning algorithms can be adopted. The problem with this is that the links between subgraphs in a single graph are omitted. This omission unfortunately loses some valuable information.

On the other hand, although the topic detection in text mining [8] and video processing [9] is a hot research area, the research issue, that of discovering hidden topics in graph-structured data, has not been well solved. For example, in the text mining area, if topics are from scientific papers about a research area, the topics mean the different research directions of this research area, i.e., *cloud computing* and *machine learning*; in the image mining area, if topics are about the images in a scene, the topics mean the different background semantics (e.g. the combinations of objects, like the combination of 'sky' and 'water' can actually be the 'environment'). These discovered topics are useful for many real-world tasks, such as, topic detection and tracking in text mining, image segmentation and retrieval, and dimensionality reduction, but we do not have a suitable solution for graphs since the existing works on topic discovery are only based on instance-represented data [10]–[12]. A challenging question therefore arises: how do we discover hidden topics for graph-structured data?

In order to resolve this research issue, we propose a topic model for graph mining (GTM) in this paper. To the best of our knowledge, we are the first to apply the topic model for graph mining. Although topic models have proved to be very successful in discovering latent topics, the standard topic models cannot be directly applied to graph-structured data because of the 'bag-of-word' assumption. Here, we make an assumption that if there is an edge between two nodes in a graph, these two nodes tend to 'talk' similar content. In GTM, a Bernoulli distribution is adopted to model the existence of an edge parameterized by topics of two linked nodes. By directly the modelling edges, GTM can make the edges contribute to latent topic discovery instead of the process of using frequent subgraphs. Finally, we compare the performance of GTM and Latent Dirichlet Allocation (LDA) on the classification task. The experimental results show that the ability of GTM on document and chemical formula classifications is better than LDA. The discovered topics by GTM are also better than LDA. In other words, topics from GTM can more accurately describe graphs than LDA.

The contributions of this paper are:

J. Xuan is with the Centre for Quantum Computation & Intelligent Systems (QCIS), School of Software, Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS), Australia and the School of Computer Engineering and Science, Shanghai University, China (e-mail: xuanjunyu@shu.edu.cn).

J. Lu is with the Centre for Quantum Computation & Intelligent Systems (QCIS), School of Software, Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS) Australia (e-mail: Jie.Lu@uts.edu.au).

G. Zhang is with the Centre for Quantum Computation & Intelligent Systems (QCIS), School of Software, Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS) Australia (e-mail: Guangquan.Zhang@uts.edu.au).

X. Luo is with the School of Computer Engineering and Science, Shanghai University, China (e-mail: luoxf@shu.edu.cn).

1) An innovative Graph topic model (GTM) for graph-structured data is built by modelling the edges in graphs using Bernoulli distribution, which makes the edges in graphs contribute to the discovered topics;
2) Two inference algorithms: Variational algorithm and Markov Chain Monte Carlo algorithm are developed to resolve the proposed GTM.

The rest of this paper is organized as follows. Some related works are given in Section II. In Section III, we introduce the proposed GTM with the inference algorithms. In Section IV, experiments are conducted to compare traditional LDA with the proposed GTM on the classification. Finally, Section V concludes this study and discusses further work.

II. Related Work

In this section, we will review the state-of-the-art research in two areas: topic models and graph mining. A successful model and also our competitive model will be introduced in more detail.

A. Topic models

Probabilistic Latent Semantic Indexing (pLSI) [13], which is an extension of Latent Semantic Indexing (LSI) [14], can be seen as the first topic model. The original idea of this comes from the sparse document-keyword matrix. LSI uses Singular Value Decomposition (SVD) from the dimension reduction view and pLSI builds a generative model to find the latent classes (topics). However, there is an over-fitting problem in the pLSI model, which is addressed by Latent Dirichlet Allocation (LDA) [11] using a Dirichlet prior to all the topic distributions with the resulting sacrifice of the inference complexity [15]. There are also many extensions of LDA which have considered different aspects of data, such as the label [16], time [12], author [17], emotion [18] and so on. More information about topic models can be found in [19], [20].

There are also some works which try to capture the dependencies using topic models. Here, we class them as three categories according to the dependence level: topic-level, document-level and word-level. At the topic level, a correlated topic model is proposed to capture the relations between hidden topics by replacing the Dirichlet prior with Log-normal prior [21]. An infinite topic tree is learned from the data by a nested Hierarchal Dirichlet Process [22]. At the document-level, the citation relations between scientific papers are considered by a relational topic model [23], [24]. Both topic-level and document level models are still based on the ‘bag-of-word assumption. At the word level, Thomas Griffiths [25] tries to fill this gap by adding the syntactic relations of words in a sentence to the model. The Hidden Markov Model (HMM) [26] is combined with the topic model by assuming that the keywords in a document are generated under an inherent linguistic sequence. Although these two works have proved to be successful, only the linguistic linear relation of all keywords in a sentence or document are considered. Actually, there are many types of relations between keywords in a document and these relations are not limited in a sentence. Our work belongs to the third level, i.e. the word-level. Compared with current works, our model can be extended to the graph mining area, because our model can capture more general relations (e.g. graph structure).

B. Graph mining

Graph mining [27] has become an important research topic and has been successfully applied to numerous applications, like computational biology, chemistry and so on. Compared with the traditional instance-based data representation, the graph-based representation expresses more data information which constitutes the structure of data. The main work in the graph mining area is to find a way to incorporate this structural information into the traditional algorithms, such as classification, clustering, frequent pattern mining, and so on.

Frequent subgraph mining is of significance in graph mining because it is the bridge connecting traditional data mining algorithms and graphs. So, there are plenty of frequent subgraph mining algorithms proposed in the literature which are categorized in terms of ‘general purpose’ and ‘pattern dependent’ algorithms [28]. Based on the graph traversing strategies, BFS and DFS, some algorithms are designed, like FSG [29], DPMine [30], gSpan [31], GASTON [32], etc.

Graph classification provides the labels for the unlabelled graphs using labelled graphs as training data. A similarity measure between two graphs is the basis for the graph classification, because the traditional measures do not work for graphs, e.g. cosine, Euclidean distance, and Minkowski distance. Kernel-based [33] methods are proposed to resolve this problem. Since the direct comparison between two graphs is the NP hard problem, some graph kernels [34], [35] are proposed to measure the distance between the two graphs by considering the properties of graphs, like nodes, edges, paths. The original kernel-based methods [34], [36] are normally time-consuming. For example, the geometric random walk graph kernel [36] requires $O(n^6)$ time. Four approximation methods are proposed to reduce this complexity to $O(n^4)$, and, for some sparse graphs, the time only requires $O(n^2)$ [37]. Subgraph-based graph classification is also a prevalent method [5], [7]. Some boosting approaches have been adopted to select interesting subgraphs from the original big subgraph set [38]. The dual active features [6] and positive labels [39] are considered.

Graph clustering clusters the graphs with similar labels. Apparently, the most important thing for graph clustering is the existence of similarity between two graphs. Some ideas of graph classification can also be adopted here. There are also some other methods, like the entropy-based method [40] and some works try to improve efficiency by employing the parallel algorithm [41].

To sum up, although various methods about graph have been proposed, there is little work on the probabilistic model for graph mining. Accordingly, we have sought to build a probabilistic model for the different kinds of graphs based on the idea of topic models.

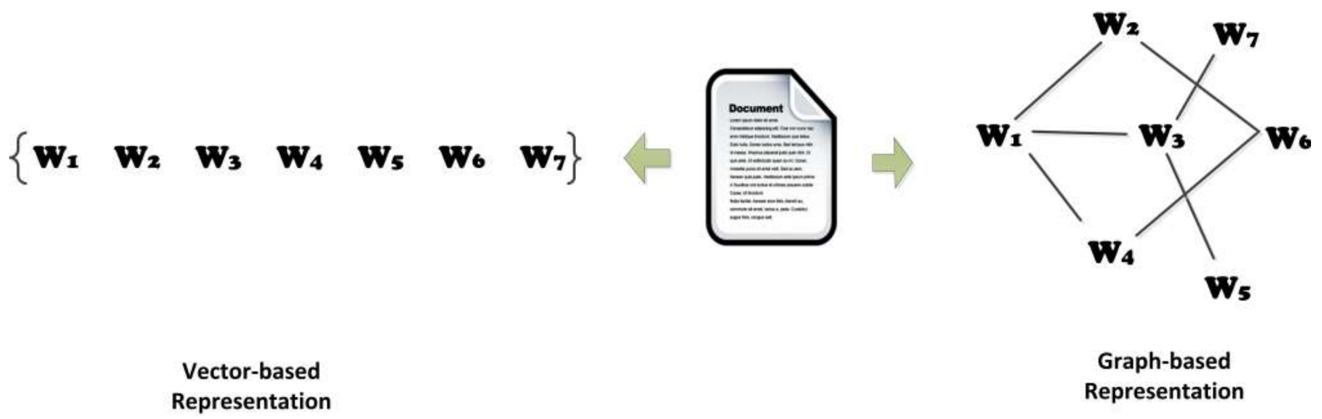

Fig. 1: Instance-based and Graph-based representation of a document. Each node in the graph denotes a word in the document. The edges denotes a kind of relationes between words.

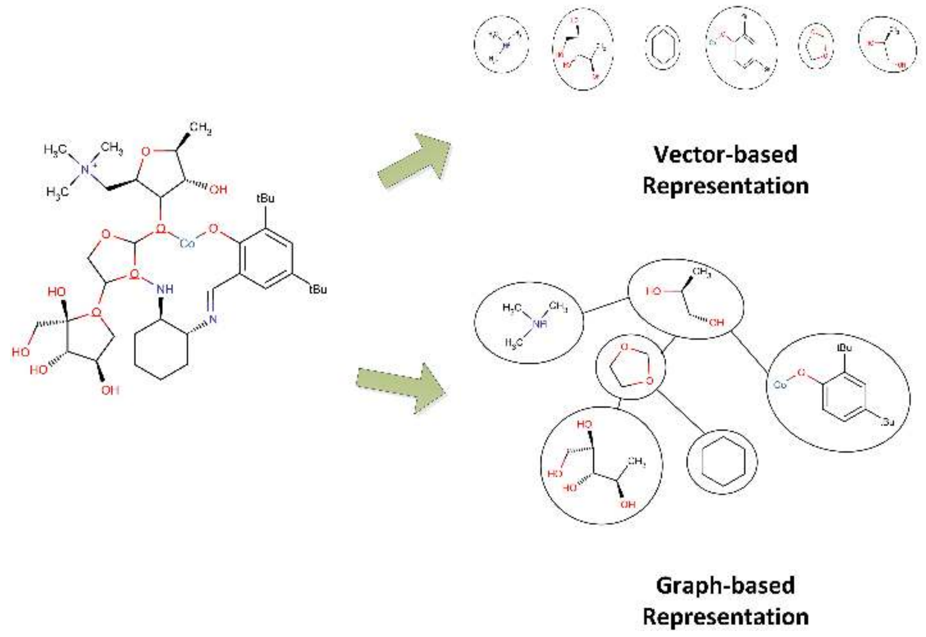

Fig. 2: Instance-based and Graph-based representation of a chemical formula. A chemical formula (left figure) is composed by different elements, like O, C, HO, and so on. The connections between these elements are called bonds between them, which have almost same ability to influence the function of a chemical formula. Here, two different representations are constructed for a same chemical formula.

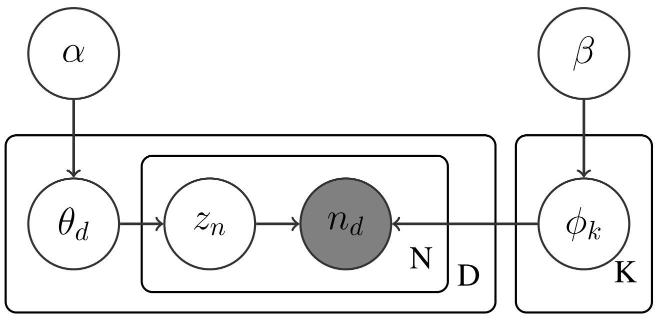

Fig. 3: Graphical representation of LDA

C. Latent Dirichlet Allocation

Since our work is based on Latent Dirichlet Allocation (LDA) and it aims to extend LDA to graphs, we will provide more details about this model. LDA [11] is a generative graphical model, as shown in Fig .3. The documents are selected as the example dataset in line with the original paper, but it should be noted that LDA can be used for any instance-based representation objects. The aim of this model is to discover the underlying topics in a corpus. It assumes that a document is composed by a number of topics with different weights (called topic distribution), and a topic is composed by a number of keywords [1] with different weights (called keyword distributions). The generative process is:

- Draw $\phi_z \sim Dir(\beta)$ for each topic;
- Draw $\theta_d \sim Dir(\alpha)$ for each document;

[1]Keywords are the selected words to express/represent the semantics of the documents. Here, the keywords in this paper are non-stopwords and these are stemmed by the NLP tools. All the words are also given certain POS, and then only the nouns and verbs are kept analogous with other literature, because the document classification is mainly sensitive to nouns and verbs.

- For all keywords in a document:
  - Draw $z_{d,n} \sim Multi(\theta_d)$ for each keyword;
  - Draw $n_d \sim Multi(\phi_{z_{d,n}})$ for each keyword;

The inference for this model is the inverse process of this generative process. The Expectation Maximization algorithm is adopted to get optimized latent variables for maximizing the model likelihood. In this generative process, the topic assignment of each keyword is determined by the topic distribution of the belonged documents, and each keyword is determined by its topic assignment and keyword distribution of topics, as shown in Fig. 3. It is a pity that the relations between keywords in a document have been overlooked in previous studies.

III. Graph Topic Model

In this section, we first present some basic concepts used in this paper. Based on these concepts, we introduce the proposed model followed by two inference algorithms. In order to show the procedure and intrinsic of the algorithms, an illustrative example is given at last.

A. Graph

The definition of graph is given here and its representative ability is shown by two examples: a document graph and a chemical graph. Besides, some basic concepts and notations will also be given, which will be used throughout this paper.

Definition 1 (Graph): A graph g is composed by nodes and edges,

$$g := < V, \{e\} > \tag{1}$$

where V is a node set in a dataset G and $\{e\}$ is an edge set of relations between nodes within this graph. Examples are shown in Fig. 1 and Fig. 2.

Graph can be used to model or represent lots of items. In this paper, we give two examples of graphs, one is document graph and the other is chemical graph.

1) Document Graph: A document is apparently composed by some words, which is the reason why most of researches about documents using a word vector to represent a document. In order to construct a graph for a document, we just need to add relations between these keywords as shown in Fig. 1. The relation in this paper between keywords is selected as the co-occurrence relation. Co-occurrence frequency of two keywords is,

$$f_{co} = \frac{|G_{n_i} \cap G_{n_j}|}{|G|}$$

where G_{n_i} is the documents which contain keyword n_i. An edge e_{n_i,n_j} exists only if the co-occurrence frequency of them exceeds a threshold ρ. This edge means that these two keywords have a big probability to be used to describe a similar topic. If there are a number of topics, it means these two keywords may have similar topic distribution.

The reason why this co-occurrence relation is selected here is that it is a weak-semantic relation between keywords and can be easily and automatically constructed without appealing to other resources. Actually, there are other concrete and rich-semantic relations can be discovered by some other methods, like Resource Description Framework (RDF) [42] or Ontology [43], [44]. However, they normally need the help of outer data resources or the human intervention. For an arbitrary corpus, co-occurrence relation is a better choice.

TABLE I: Notations used in this paper

Symbol	Description
$\|G\|$	the number of graphs in a graph dataset G
N	the number of nodes in a graph dataset G
K	the number of topics in a dataset
θ_g	topic distribution of graph g
$z_{g,n}$	topic assignment of node n of graph g
n_g	node n of graph g
$e^g_{i,j}$	an edge between nodes n_i and n_j in graph g
ϕ_k	node distribution of topic k

2) Chemical Graph: Each chemical formula seems naturally a graph with each chemical element as a node and chemical bonds as edges. However, the nodes may occur twice in a single chemical formula with different positions. Since a treatment function is normally expressed by the combination of basic chemical elements [45], we cannot directly use the original graph structure.

Here, we use the frequent subgraphs in a dataset as the nodes and the links between subgraphs as edges. At first, gSpan[2] is adopted to mine the subgraphs. Each chemical formula is re-represented as a vector of subgraphs,

$$g_c = < sg_1, sg_2, ..., sg_n > \tag{2}$$

where sg_i denotes the weight of ith subgraph in this chemical graph g_c. At last, the subgraph relations are defined as their inclusive relation. For example, if a subgraph sg_i contains another subgraph sg_j, there will be an edge $< sg_i, sg_j >$ between sg_i and sg_j as shown Fig. 2.

Definition 2 (Topic): A topic is a vector of all nodes in a dataset with their weights.

From different datasets, topics have different meanings. For example, the topics from scientific papers can be seen as the different research directions; the topics from chemical formulas can be seen as the different treatment functions, such as activity, toxicity, etc.

Some other frequently used notations in this paper are listed in Table I.

B. Proposed Model

This model is an extension of LDA, so it is also a generative model. The graphical representation of GTM is shown in Fig. 4 and the corresponding generative process is,

1) draw $\phi_k \sim Dir(\beta)$ for each topic;
2) draw $\theta_g \sim Dir(\alpha)$ for each graph;
3) for all nodes in a graph:
   a) draw $z_{g,n} \sim Multi(\theta_g)$ for one node;
   b) draw $n_g \sim Multi(\phi_{z_{g,n}})$ for one node;
4) for all edges in a graph:

draw $e^g_{n_i,n_j} \sim \pi(p^g_{n_i,n_j})$ for an edge e_{n_i,n_j},

$$\begin{aligned} p^g_{n_i,n_j}(e^g_{i,j} = 1) =& f(z_{g,n_i}, z_{g,n_j}, \phi) \\ =& \phi_{z_{g,n_i}} \circ \phi_{z_{g,n_j}} \end{aligned} \tag{3}$$

[2]http://www.cs.ucsb.edu/ xyan/software/gSpan.htm

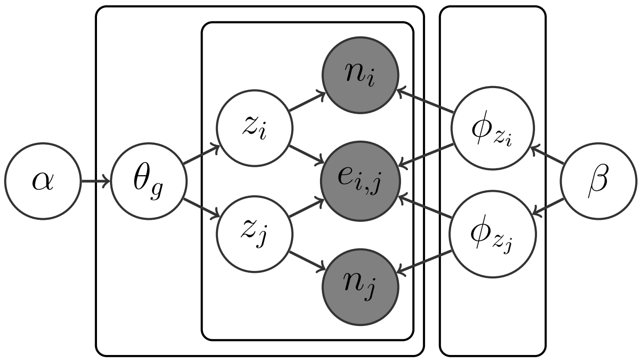

Fig. 4: Graphical representation of GTM

The $p^g_{n_i,n_j}$ is the parameter of Bernoulli distribution of the existing of an edge between two nodes n_i and n_j. It should be noted that the edge generation here is different from the edge definition in Section III.A. The edge definition is a description of the observed data. But the edge generation here is a part of the statistical model assumption. We just learn the data from this model assumption. The difference from LDA and the main idea of this model lie on modelling the edges in graphs using Bernoulli distribution parameterized by the topic distribution of nodes. We can see from the Fig. 4 that $e^g_{n_i,n_j}$ is generated by $\{z_{g,n_i}, z_{g,n_j}, \Phi\}$. It means that the probability of the existence of an edge between two nodes is determined by the similarity of their topic distributions. This similarity is measured by vector inner product between $\phi_{z_{g,n_i}}$ and $\phi_{z_{g,n_j}}$, where $\phi_{z_{g,n_i}}$ is node distribution of topic $\{z_{g,n_i}\}$, as shown in Eq. (3). The more similar topics of two keywords are, the more likely there is an edge between these two nodes. As discussed in next subsection, the learning process for this model will show that the edges between nodes will influence the topic assignment of nodes and then influence the topic distribution of a graph. It is just this influence that makes the discovered topics are better than ones from LDA.

C. Variational Inference

For model learning, variational Expectation Maximization is adopted to learn the posterior distribution of graphs. It has two steps: e-step and m-step. In the e-step, the key work is to compute probability distribution of latent variables of the model. The posterior distribution of latent variables of a graph is,

$$p(\theta_g, \boldsymbol{z_g}, \boldsymbol{\phi_{1:K}} | \boldsymbol{n_g}, \boldsymbol{e_g}, \alpha, \beta) \tag{4}$$

Generally, this distribution is intractable to compute. The idea of variational inference is to use Jensen's inequality to maximize the lower bound on the log likelihood. The original posterior distribution in Eq. (4) is factorized into some selected distributions parameterized by variational parameters.

For GTM, the distributions used to factorize posterior distribution are,

$$\begin{aligned} \theta_g &\sim Dir(\gamma_g) \\ z_{g,n} &\sim Multi(\varphi_{g,n}) \\ \phi_k &\sim Multi(\kappa_k) \end{aligned} \tag{5}$$

where γ_g, $\varphi_{g,n}$ and κ_k are variational parameters. Actually, they are not just three distributions but three distribution families composed by distributions with different parameter values. Then, the Eq. (4) is factorized as,

$$\begin{aligned} &q(\theta_g, \boldsymbol{z_g}, \boldsymbol{\phi_{1:K}} | \gamma_g, \boldsymbol{\varphi_g}, \boldsymbol{\kappa_g}) \\ &= q_\theta(\theta_g|\gamma_g) \prod_{n=1}^{N} q_z(z_{g,n}|\varphi_{g,n}) \prod_{k=1}^{K} q_\phi(\phi_k|\kappa_k) \end{aligned} \tag{6}$$

This is an approximation of the posterior distribution. The distance between this approximation and original posterior distribution can be measured by KL distance, as

$$\begin{aligned} &\log p(\boldsymbol{n_g}, \boldsymbol{e_g}|\alpha, \beta) \\ &= \log E_q\left[\frac{p(\theta_g, \boldsymbol{z_g}, \boldsymbol{\phi_{1:K}}, \boldsymbol{n_g}, \boldsymbol{e_g}|\alpha, \beta)}{q(\theta_g, \boldsymbol{z_g}, \boldsymbol{\phi_{1:K}})}\right] \\ &\geq E_q\Big[\log p(\theta_g, \boldsymbol{z_g}, \boldsymbol{\phi_{1:K}}, \boldsymbol{n_g}, \boldsymbol{e_g}|\alpha, \beta)\Big] \\ &\quad - E_q\Big[\log q\left(\theta_g, \boldsymbol{z_g}, \boldsymbol{\phi_{1:K}}\right)\Big] \end{aligned} \tag{7}$$

Through the adjusting of variational parameters, the distributions that can maximize the Eq. (7) can be found from the variational distribution families. After transforming the searching of variational distributions to an optimization problem, the variational parameters can be computed as,

$$\gamma_{g,k} = \alpha_k + \sum_{n}^{N} \varphi_{g,n,k} \tag{8}$$

and

$$\begin{aligned} &\frac{\partial f(\varphi_{g,n,k})}{\partial \varphi_{g,n,k}} \\ &= \Big(\Psi(\kappa_{k,n}) - \Psi(\sum_{n}^{N} \kappa_{k,n}) + \Psi(\gamma_{g,k}) \\ &\quad - \Psi(\sum_{k}^{K} \gamma_{g,k}) - \log \varphi_{g,n,k} - 1\Big) \\ &\quad + \sum_{n_j \in Ne(n_i)} \left(\zeta^{-1} \cdot \kappa_{k,n_j}^{\varphi_{g,n_j,k}} \cdot \ln \kappa_{k,n} \cdot \kappa_{k,n}^{\varphi_{g,n,k}}\right) \end{aligned} \tag{9}$$

and

$$\begin{aligned} &\frac{\partial f(\kappa_{k,n})}{\partial \kappa_{k,n}} \\ &= \left(\Psi'(\kappa_{k,n}) - \Psi'(\sum_{n}^{N} \kappa_{k,n})\right) (\varphi_{g,n,k} + \beta_n - \kappa_{k,n}) \\ &+ \sum_{n_j \in Ne(n_i)} \left(\zeta^{-1} \cdot \varphi_{g,n_i,k} \cdot \kappa_{k,n_i}^{\varphi_{g,n_i,k}-1} \cdot \kappa_{k,n_j}^{\varphi_{g,n_j,k}-1}\right) \end{aligned} \tag{10}$$

and

$$\zeta = N_{(n_i,n_j)} \sum_{k}^{K} \left(\kappa_{k,n_i}^{\varphi_{g,n_i,k}} \cdot \kappa_{k,n_j}^{\varphi_{g,n_j,k}}\right) \tag{11}$$

where $Ne(n)$ is the number of neighbors of a node n and ζ is the a parameter of Taylor expansion of log probability of Eq. (3) (The detail is shown in Appendix) . We can compute

```
Algorithm 1: Variational Inference for GTM
  Input: Topic number $K$, graph dataset $G$
  Output: $\gamma$, $\varphi$ and $\kappa$
   1: random initialization of variational variables $\gamma$, $\varphi$ and $\kappa$
   2: $i = 1$
   3: while $i \leq max_{iteration}$ do
   4:    for $g = 0$ to $|G|$ do
   5:       Update $\gamma_g$ through Eq. (8)
   6:       Update $\varphi_g$ by gradient-based optimization with
            derivative in Eq. (9)
   7:    end for
   8:    Update $\kappa$ by gradient-based optimization with
         derivative in Eq. (10)
   9:    Update $\zeta$ by gradient-based optimization with
         derivative in Eq. (11)
  10:    i = i + 1.
  11: end while
```

```
Algorithm 2: MCMC Inference for GTM
  Input: Topic number $K$, graph dataset $G$
  Output: $\phi$, $\theta$ and $z$
   1: random initialization of variables $\phi$, $\theta$ and $z$
   2: $i = 1$
   3: while $i \leq max_{iteration}$ do
   4:    for $g = 0$ to $|G|$ do
   5:       Update $\theta_g$ through Eq. (13)
   6:       Update $z_g$ through Eq. (14)
   7:    end for
   8:    Update $\phi$ through Eq. (17)
   9:    i = i + 1.
  10: end while
```

the exact form of γ and ζ, but φ and κ cannot. So, gradient-based optimization method [3] is adopted to get the optimized φ and κ. The whole procedure, named *Variational Inference for GTM*, is shown in Algorithm 1.

After getting the posterior distribution of each graph in Eq. (4), we need to maximize the likelihood of the graph by selecting α and β in the m-step. The Newton method [4] is adopted here. The detail is omitted, because there is no difference from the method used in LDA.

Let us see how the edges in a graph impact on the topic distribution θ_g of this graph. Since γ_g is the variational parameter of θ_g, the value of γ_g will influence the topic distribution θ_g of a graph g. In Eq. (8), it can be seen that γ_g is influenced by topic assignments $\varphi_{g,n}$ of its nodes and $\varphi_{g,n}$ is in turn impacted by its neighbors as the Eq. (9) shown. This is consisted with our former discussion about GTM.

D. Markov Chain Monte Carlo (MCMC) Inference

Another method to get posterior distribution in Eq. (4) for each graph is Gibbs sampling, which construct a Markov chain with stationary distribution as the desired posterior distribution. What we need to do is to find the conditional distributions for each variables in this posterior distribution in the model.

At first, since the prior of θ is Dirichlet distribution and its likelihood is multinomial distribution, the posterior distribution of θ_g conditioned on all other variables is easily found out as,

$$p(\theta_g|\cdots) \sim Dir(\alpha_1 + m_{g,1}, \alpha_2 + m_{g,2}, \cdots, \alpha_K + m_{g,K}) \tag{12}$$

where $m_{g,k}$ is the number of $z_i = k$ in graph g. Eq. (13) relies on the conjugation between Dirichlet distribution and multinomial distribution.

Sampling z will be a little more complicated. We know that the prior for $z_{g,n} = k$ is multinomial distribution parameterized by θ_g. The likelihood should be,

$$\begin{aligned} &p(n|\phi_{z=k}) \cdot \left(\prod_{m \in Ne(n)} p(e_{n,m} = 1|\phi_{z=k}, \phi_{k_m}) \right) \\ &\left(\prod_{j \notin Ne(n)} p(e_{n,j} = 0|\phi_{z=k}, \phi_{k_j}) \right) \end{aligned} \tag{13}$$

where k_m is the topic assignment of node m. The likelihood contains three parts. The first part is for the generation of node n. Second part is for the generation of edges in graph g. It should be noted that the third part, which is for the edges with 0 weights, is also necessary. With the prior and likelihood in hand, the conditional distribution of $z_{g,n} = k$ is given as,

$$\begin{aligned} &p(z_{g,n} = k|\cdots) \\ &\propto \theta_k \cdot p(n|\phi_k) \cdot \left(\prod_{m \in Ne(n)} p(e_{n,m} = 1|\phi_k, \phi_{k_m}) \right) \\ &\left(\prod_{j \notin Ne(n)} p(e_{n,j} = 0|\phi_k, \phi_{k_j}) \right) \end{aligned} \tag{14}$$

For the ϕ_k, its prior is Dirichlet distribution with parameter β. The likelihood should be derived for its conditional distribution. In GTM, the Φ are used to generate nodes and edges. So, the likelihood for node generations of ϕ_k is,

$$\prod_{n:k_n=k} p(n|\phi_k) \tag{15}$$

$n : k_n = k$ are the nodes that are assigned to topic k. And the likelihood for edge generations is,

$$\begin{aligned} &\left(\prod_{e_{n,m}=1 \& k_n=k} p(e_{n,m} = 1|\phi_k, \phi_{k_m}) \right) \\ &\left(\prod_{e_{n,m}=0 \& k_n=k} p(e_{n,m} = 0|\phi_k, \phi_{k_m}) \right) \end{aligned} \tag{16}$$

Here, all the edges with one node assigned to topic k are separated into two groups, one is with weight 1 (existence) and one is with weight 0 (non-existence). Combine the prior

[3]http://en.wikipedia.org/wiki/Gradient_descent
[4]http://en.wikipedia.org/wiki/Newton's_method

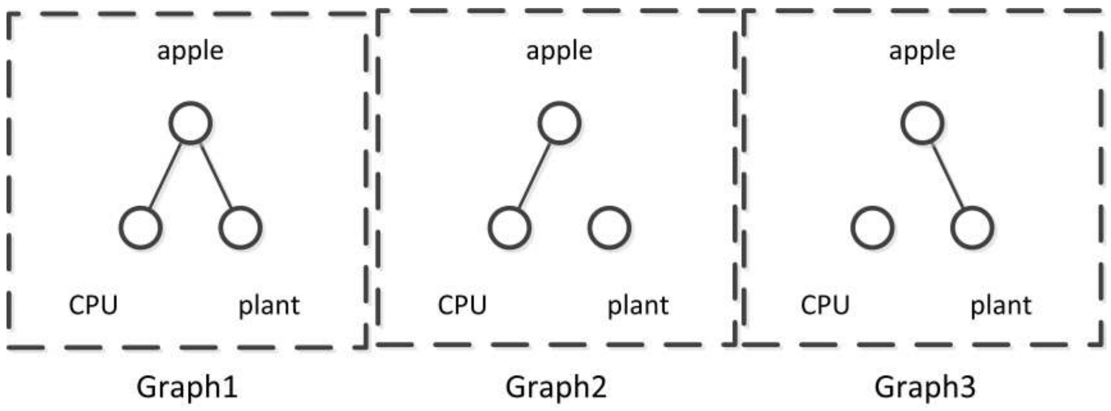

Fig. 5: An illustrative example

and two likelihoods,

$$p(\phi_k|\cdots) \propto p(\phi_k|\beta) \cdot \prod_{n:k_n=k} p(n|\phi_k) \left(\prod_{e_{n,m}=1 \bigcap k_n=k} p(e_{n,m}=1|\phi_k, \phi_{k_m}) \right) \left(\prod_{e_{n,m}=0 \bigcap k_n=k} p(e_{n,m}=0|\phi_k, \phi_{k_m}) \right) \tag{17}$$

The iterative sampling θ_g, $\boldsymbol{z_g}$, $\boldsymbol{\phi_{1:K}}$ for all graphs will get the samples of posterior distribution of the whole dataset. Except for θ_g, $\boldsymbol{z_g}$, $\boldsymbol{\phi_{1:K}}$, the model parameters α and β could also join the sampling without predefining specific values. For the symmetric Dirichlet distribution, the prior for α and β could be Gamma distribution and the likelihoods are also easily computed. The final algorithm, named *MCMC Inference for GTM*, is described in Algorithm 2.

E. An intuitive example to show the implication of GTM

Here, a simple example is made up to show how the algorithms work. Suppose we have a document graph dataset that contains three document graphs as shown in Fig. 5. There are only two hidden topics discussed in these documents: the first is about *planting apple tree* and the second is about *apple computer and electronic equipments*. The topic assignment of each graph can be represented by a two dimensional vector (a, b) in which a denotes the probability of this graph assigned to first topic and b denotes the probability of this graph assigned to second topic. In Fig. 5, first frame denotes a very simple graph in dataset and composed by three nodes (words): *plant*, *CPU* and *apple* with an edge between *apple* and *CPU* and an edge between *apple* and *plant*. The other two graphs have same nodes with the first one but different edges.

At first, we can see that this graph dataset will be equal to the corresponding instance dataset if the edges are omitted. In this situation, all the graphs are identical. Take the first graph as an example, there is one word *CPU* that belongs to second topic *apple computer and electronic equipments*. At the same time, there is one word 'plant' that belongs to first topic *planting apple tree*. Third word *apple* could belong to either one. To sum up, the topic assignment of the first graph is $(0.5, 0.5)$. The other two graphs are same with the first graph.

However, when we consider their structure (edges), three graphs are not identical any more. Take Algorithm 2 as an example. We do the following steps:

1) Input: three graphs and topic number is two;
2) Initialization: randomly set the topic assignments of graphs. Here, we give them the same topic assignments: $(0.5, 0.5)$; topic assignments of keywords of all graphs: $(0.5, 0.5)$; and the keyword assignments (*apple*, *CPU*, *plant*) of topics $(0.33, 0.33, 0.34)$;
3) For the first graph, we can update the topic assignment of the first graph from $(0.5, 0.5)$ to $(0.5, 0.5)$, according to Eq. (13);
4) According to Eq. (14), the topic assignment of *apple* will change from $(0.5, 0.5)$ to $(0.5, 0.5)$, the topic assignment of *CPU* will change from $(0, 1)$ to $(0.2, 0.8)$, and the topic assignment of *plant* will change from $(1, 0)$ to $(0.8, 0.2)$;
5) For the second graph, we can update its topic assignment from $(0.5, 0.5)$ to $(0.3, 0.7)$, according to Eq. (13);
6) According to Eq. (14), the topic assignment of *apple* will change from $(0.5, 0.5)$ to $(0.3, 0.7)$, the topic assignment of *CPU* will change from $(0, 1)$ to $(0.1, 0.9)$, and the topic assignment of *plant* will change from $(1, 0)$ to $(1, 0)$;
7) For the third graph, we can update its topic assignment from $(0.5, 0.5)$ to $(0.7, 0.3)$, according to Eq. (13);
8) According to Eq. (14), the topic assignment of 'apple' will change from $(0.5, 0.5)$ to $(0.7, 0.3)$, the topic assignment of *CPU* will change from $(0, 1)$ to $(0.1, 0.9)$, and the topic assignment of *plant* will change from $(1, 0)$ to $(0.9, 0.1)$;
9) Then, we update keyword assignment of topics using Eq. (17), the first topic from $(0.33, 0.33, 0.34)$ to $(0.3, 0.2, 0.5)$, and the second topic from $(0.33, 0.33, 0.34)$ to $(0.3, 0.5, 0.2)$;
10) Stop until reach the max iteration number.

Using the Algorithm 2, we get different topic assignments of graphs. The topic assignment of the first graph does not change. But the one of second graph changes from $(0.5, 0.5)$

TABLE II: Statistics of document graph dataset

Topic name	document number	all keyword number
earn	3722	14424
acq	2127	

TABLE III: Value of ρ

ρ	average document graph density
0.01	0.3104
0.0042	0.4017
0.0017	0.5023
0.0006	0.6103

(the result of omitting edges) to $(0.3, 0.7)$ (the result of Algorithm 2). It means the second graph is more likely discussing second topic. The reason why the Algorithm 2 gets this result is because there is an edge between apple and CPU. By the effect of this edge, the word *apple* is more likely discussing second topic. Overall, the second graph is more tend to talk about second topic.

From this simple example, we can see that the edges do affect the hidden topic discovery and intuitively these edges can improve the accuracy of discovered topics from the Fig. 5. Next, we will test on the real-world datasets to verify our algorithms.

IV. Experiments and result analysis

In order to verifying the merit of considering edges between nodes, we compare our proposed GTM with LDA on two different types of datasets, one type is document and the other is chemical formula. The objects are represented as instances and graphs and trained by LDA and GTM, respectively. After that, objects are re-represented by the topic distributions that are outputs of both models. This new re-represented objects are used for a typical graph mining task–classification. The accurate classification result means the topics can better represent the semantic of objects. The implementation of LDA is from JGibbLDA[5] . The two inference methods of GTM are implemented by Matlab in this paper for documents and chemical formula, respectively.

A. Document classification

The document dataset used here is Reuters-21578[6] in which documents have been labelled with topics. Two topics, 'earn' and 'acq', are selected and some statistics are shown in Table II after removing documents which have less than 10 keywords (only considering noun and verb). Each document in these two topics is represented by 90% of keywords (only considering noun and verb) ranked by tf-idf [46] in this document. k-Nearest Neighbour algorithm (k-NN) [47] is selected as the classification method, because it is simple and does not do much operations on the features of data comparing with other classification methods, like Decision tree or SVM. The implementation of k-NN comes from Weka[7] with 10 times cross-validation. In order to compare LDA and GTM, the

[5]http://jgibblda.sourceforge.net/#Griffiths04
[6]http://www.daviddlewis.com/resources/testcollections/reuters21578/
[7]http://www.cs.waikato.ac.nz/ml/weka/

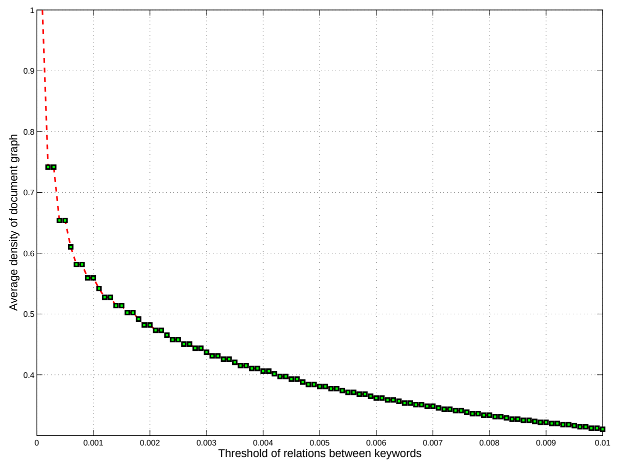

Fig. 6: Threshold of relations between keywords

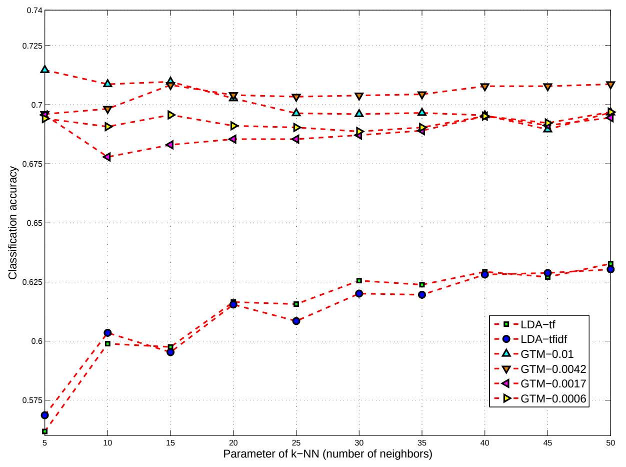

Fig. 7: The influence of parameter of k-NN (the number of neighbours from 5 to 50)

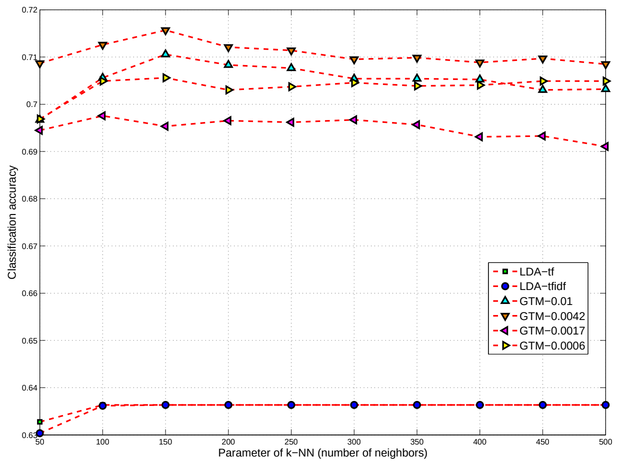

Fig. 8: The influence of parameter of k-NN (the number of neighbours from 50 to 500)

TABLE IV: Statistics of chemical graph dataset

BioassayID	Topic name	chemical formula number	subgraph number
NCI33	*active*	1000	252
	inactive	1000	
NCI47	*active*	1000	325
	inactive	1000	
NCI81	*active*	1000	251
	inactive	1000	

numbers of topics are all set 2 which is just the number of topics of data.

Since relations between keywords are considered in GTM, the number of links in documents need to be given, which is controlled by threshold of co-occurrence frequency ρ. As shown in Fig. 6, the distribution of average density of document graphs and ρ approximate power-law distribution. The average document graph densities are given in Table III in which four kinds of density and corresponding ρ are given.

The results are shown in Fig. 7 and Fig. 8. There are 6 methods are compared in these two figures: LDA-tf (original LDA), LDA-tfidf (use tf-idf to replace tf of keyword weights in original LDA [48]), GTM-0.01 (GTM with $\rho = 0.01$), GTM-0.0042 (GTM with $\rho = 0.0042$), GTM-0.0017 (GTM with $\rho = 0.0017$) and GTM-0.0006 (GTM with $\rho = 0.0006$). To sum all, the efficiency of GTM is better than LDA. The density of average graphs impacts on the efficiency of GTM, because the GTM relies on the relations between keywords. However, it does not mean that the more links the better. As shown in Fig. 7 and 8, the best is $\rho = 0.0042$ and average document graph density is 0.4017. We believe that this value is not fixed and depends on dataset. If there is no co-occurrence of keywords in documents of a corpus and then documents are totally 'separated', GTM is not better than LDA.

B. Chemical formula classification

In this section, we use a common benchmark dataset, *NCI cancer screening dataset*[8]. Each chemical formula is represented as a graph with atoms as nodes and bonds as edges. According to the activity against corresponding cancer, each graph has a label active' or 'inactive'. We use them as the dataset, and select 1000 active graphs and 1000 inactive graphs of which frequent subgraphs are mined by gSpan algorithm[9] with support 30. The statistics are shown in Table IV. We use these graphs as the original data and transform them into new graphs with frequent subgraphs as nodes by the method proposed in Eq. 2.

To compare the performance of GTM with LDA, we use the topic distributions from both LDA and GTM as the new representations of graphs to do classification. The more accurate the discovered topics are, the better the classification results would be. The MCMC inference method is used for GTM in this section. The Fig. 9 shows the convergence of log likelihood of GTM with Gibbs iteration number. The final results are also shown in Fig. 9. Here, four different classifiers are adopted, including k-NN, J48, SVM and NB (implementations are also from Weka). Except the SVM for NCI81, all the classifiers on three datasets indicate that the topics learned from the GTM are better than LDA. It should be noted that these differences are determined by the natures of different classifiers. Some classifiers, like kNN and J48, are more sensitive to the different data representation (the topic distribution from GTM or LDA) and some are not, like NB. This difference is not our focus in this paper. We only use these classifiers to show the topics mined from our proposed GTM are better than the ones from LDA. To sum up, our proposed GTM outweighs LDA.

Normally, the variational inference is more efficient than M-CMC inference, and MCMC inference is more easily extended to some more complicated tasks. In this paper, we just show two basic inferences for graph mining. More graph mining tasks can be benefit from topic models by extending them.

V. Conclusion and further study

In order to extend the topic model for graph mining, we have proposed a Graph Topic model. The innovative premise of this model is that Bernoulli distributions has been used to model the edges between two nodes in a graph, which are parameterized by the similarity between two topics of two linked nodes. Considering the edges of graphs, the discovered topic distribution of a graph by GTM is not just determined by its nodes. Two inference algorithms have been developed to resolve the proposed model. The experimental results on different datasets have verified that the topics discovered by GTM can describe graphs better than the ones from LDA. This improvement is due to the fact that the edges of graphs are considered in our innovative model and this makes the discovered topics far more suitable for the graph data. Therefore, the proposed model can be used for graph mining, including the supervised learning and unsupervised learning. Possible applications that could be improved by our proposed model also include document retrieval and chemical graph labelling.

In the future, we aim to investigate how to incorporate the structural information of graphs into the model and not just consider each edge separately. The reason for this is that there should be some hidden relations between these edges which will impact on their formation as well. Incorporating these structures may further improve the classification accuracy. Another interesting extension of our work is to use nonparametric learning methods to avoid predefining the number of topics, e.g. Hierarchical Dirichlet Processes [49].

Acknowledgment

Research work reported in this paper was partly supported by the Australian Research Council (ARC) under discovery grant DP110103733, DP140101366 and the China Scholarship Council. This work was jointly supported by the National Science Foundation of China under grant no.61471232 and the Key Innovation Program of Shanghai Municipal Education Commission under Grant 13ZZ064.

[8]http://pubchem.ncbi.nlm.nih.gov

[9]http://www.cs.ucsb.edu/x̃yan/software/gSpan.htm

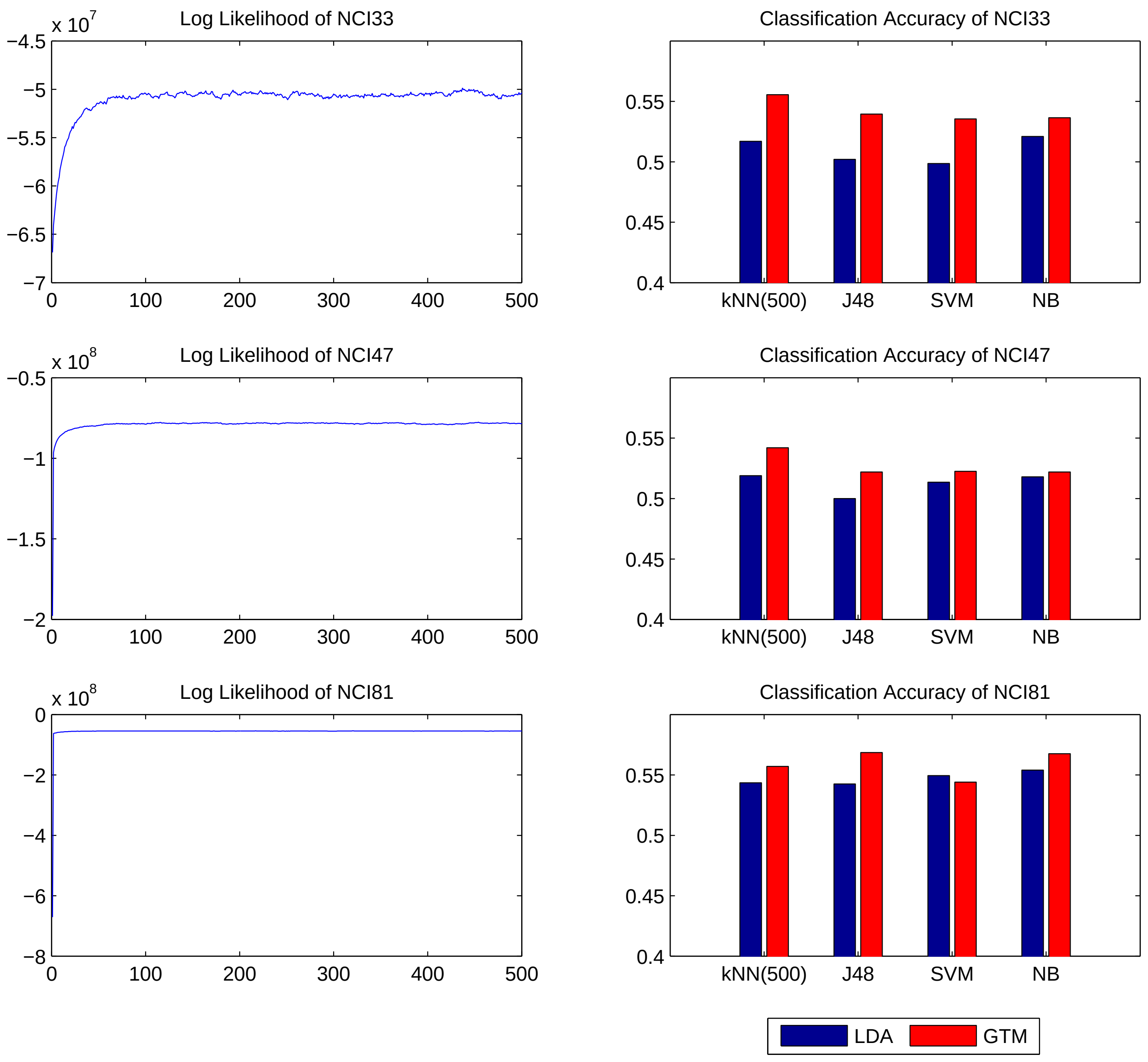

Fig. 9: The log likelihoods of GTM at each iteration by MCMC inference and the comparisons between GTM and LDA on classification using different classifiers. Here, we use four different classifiers: k-nearest neighbors (kNN), decision tree (J48), support vector machine (SVM), Naive Bays (NB).

References

[1] D. Chakrabarti and C. Faloutsos, "Graph mining: Laws, generators, and algorithms," *ACM Computing Surveys (CSUR)*, vol. 38, no. 1, p. 2, 2006.

[2] S. Rehman, A. Khan, and S. Fong, "Graph mining: A survey of graph mining techniques," in *ICDIM '12*, Aug 2012, pp. 88–92.

[3] X. Luo, N. Fang, B. Hu, K. Yan, and H. Xiao, "Semantic representation of scientific documents for the e-science knowledge grid," *Concurrency and Computation: Practice and Experience*, vol. 20, no. 7, pp. 839–862, 2008.

[4] J. Tomita, H. Nakawatase, and M. Ishii, "Calculating similarity between texts using graph-based text representation model," in *CIKM '04*, 2004, pp. 248–249.

[5] J. Vogelstein, W. Roncal, R. Vogelstein, and C. Priebe, "Graph classification using signal-subgraphs: applications in statistical connectomics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1539–1551, 2013.

[6] X. Kong, W. Fan, and P. S. Yu, "Dual active feature and sample selection for graph classification," in *KDD '11*, New York, NY, USA, 2011, pp. 654–662.

[7] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, and C. Faloutsos, "It's who you know: graph mining using recursive structural features," in *KDD '11*, 2011, pp. 663–671.

[8] Q. Liu, H. Huang, and C. Feng, "Micro-blog post topic drift detection based on lda model," in *Behavior and Social Computing*, ser. Lecture Notes in Computer Science, L. Cao, H. Motoda, J. Srivastava, E.-P. Lim, I. King, P. Yu, W. Nejdl, G. Xu, G. Li, and Y. Zhang, Eds. Springer International Publishing, 2013, vol. 8178, pp. 106–118.

[9] J. Varadarajan, R. Emonet, and J.-M. Odobez, "A sequential topic model for mining recurrent activities from long term video logs," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 100–126, 2013.

[10] M. I. Jordan, *Learning in graphical models*. MIT Press, 2004.

[11] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[12] X. Wang and A. McCallum, "Topics over time: a non-markov continuous-time model of topical trends," in *KDD '06*, 2006, pp. 424–433.

[13] T. Hofmann, "Probabilistic latent semantic indexing," in *SIGIR '99*, 1999, pp. 50–57.

[14] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the Association for Information Science and Technology*, vol. 41, no. 6, pp. 391–407, 1990.

[15] T. K. Landauer, D. S. McNamara, S. Dennis, and W. Kintsch, *Handbook of latent semantic analysis*. Psychology Press, 2013.

[16] D. M. Blei and J. D. McAuliffe, "Supervised topic models," *arXiv preprint arXiv:1003.0783*, 2010.

[17] M. Rosen-Zvi, C. Chemudugunta, T. Griffiths, P. Smyth, and M. Steyvers, "Learning author-topic models from text corpora," *ACM Transactions on Information Systems*, vol. 28, no. 1, p. 4, 2010.

[18] S. Bao, S. Xu, L. Zhang, R. Yan, Z. Su, D. Han, and Y. Yu, "Mining social emotions from affective text," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 9, pp. 1658–1670, 2012.

[19] V. Jelisavcic, B. Furlan, J. Protic, and V. Milutinovic, "Topic models and advanced algorithms for profiling of knowledge in scientific papers," in *MIPRO '12*, May 2012, pp. 1030–1035.

[20] A. Daud, J. Li, L. Zhou, and F. Muhammad, "Knowledge discovery through directed probabilistic topic models: a survey," *Frontiers of Computer Science in China*, vol. 4, no. 2, pp. 280–301, 2010.

[21] D. M. Blei and J. D. Lafferty, "A correlated topic model of science," *Annals of Applied Statistics*, vol. 1, pp. 17–35, 2007.

[22] J. Paisley, C. Wang, D. Blei, and M. Jordan, "Nested hierarchical dirichlet processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2014.

[23] J. Chang and D. M. Blei, "Hierarchical relational models for document networks," *Annals of Applied Statistics*, vol. 4, no. 1, pp. 124–150, 2010.

[24] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen, "Joint latent topic models for text and citations," in *KDD '08*, 2008, pp. 542–550.

[25] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum, "Integrating topics and syntax," in *NIPS '05*, 2005, pp. 537–544.

[26] M. Andrews and G. Vigliocco, "The hidden markov topic model: a probabilistic model of semantic representation," *Topics in Cognitive Science*, vol. 2, no. 1, pp. 101–113, 2010.

[27] S. Parthasarathy, S. Tatikonda, and D. Ucar, "A survey of graph mining techniques for biological datasets," in *Managing and mining graph data*. Springer, 2010, pp. 547–580.

[28] C. Jiang, F. Coenen, and M. Zito, "A survey of frequent subgraph mining algorithms," *Knowledge Engineering Review*, vol. 28, no. 1, pp. 75–105, 2013.

[29] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," in *ICDM '01*, 2001, pp. 313–320.

[30] N. Vanetik, E. Gudes, and S. E. Shimony, "Computing frequent graph patterns from semistructured data," in *ICDM '03*, 2002, pp. 458–465.

[31] X. Yan and J. Han, "Closegraph: mining closed frequent graph patterns," in *KDD '03*, 2003, pp. 286–295.

[32] S. Nijssen and J. N. Kok, "A quickstart in frequent structure mining can make a difference," in *KDD '04*, 2004, pp. 647–652.

[33] S. Wang, J. Wang, and F. lai Chung, "Kernel density estimation, kernel methods, and fast learning in large data sets," *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 1–20, Jan 2014.

[34] H. Kashima, K. Tsuda, and A. Inokuchi, "Marginalized kernels between labeled graphs," in *ICML '03*, vol. 3, 2003, pp. 321–328.

[35] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels," *Journal of Machine Learning Research*, vol. 12, pp. 2539–2561, 2011.

[36] W. Imrich and S. Klavzar, *Product graphs structure and recognition*. Wiley, 2000.

[37] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, "Graph kernels," *Journal of Machine Learning Research*, vol. 11, pp. 1201–1242, Aug. 2010.

[38] H. Saigo, S. Nowozin, T. Kadowaki, T. Kudo, and K. Tsuda, "gboost: a mathematical programming approach to graph classification and regression," *Machine Learning*, vol. 75, no. 1, pp. 69–89, 2009.

[39] Y. Zhao, X. Kong, and P. Yu, "Positive and unlabeled learning for graph classification," in *ICDM '11*, 2011, pp. 962–971.

[40] E. Kenley and Y.-R. Cho, "Entropy-based graph clustering: application to biological and social networks," in *ICDM '11*, 2011, pp. 1116–1121.

[41] H. N. Djidjev and M. Onus, "Scalable and accurate graph clustering and community structure detection," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 1022–1029, 2013.

[42] G. Klyne, J. J. Carroll, and B. McBride, "Resource description framework (rdf): concepts and abstract syntax," *W3C Recommendation*, vol. 10, 2004.

[43] K. Amailef and J. Lu, "Ontology-supported case-based reasoning approach for intelligent m-government emergency response services," *Decision Support Systems*, vol. 55, no. 1, pp. 79–97, 2013.

[44] C. Wang, J. Lu, and G. Zhang, "Integration of ontology data through learning instance matching," in *WI '06*, Dec 2006, pp. 536–539.

[45] A. A. Shelat and R. K. Guy, "Scaffold composition and biological relevance of screening libraries," *Nature Chemical Biology*, vol. 3, no. 8, pp. 442–446, 2007.

[46] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[47] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[48] A. T. Wilson and P. A. Chew, "Term weighting schemes for latent dirichlet allocation," in *NAACL HLT '10*, 2010, pp. 465–473.

[49] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the Americamerican Statistical Association*, vol. 101, no. 476, 2006.

Appendix A
Some derivation of variational inference

For Eq. 7, we can expand it as:

$$\begin{aligned}
&\log p(\boldsymbol{n_g}, \boldsymbol{e_g}|\alpha,\beta)\\
\geq &E_q\left[\log p(\theta_g, \boldsymbol{z_g}, \boldsymbol{\phi_{1:K}}, \boldsymbol{n_g}, \boldsymbol{e_g}|\alpha,\beta)\right]\\
&-E_q\left[\log q\left(\theta_g, \boldsymbol{z_g}, \boldsymbol{\phi_{1:K}}\right)\right]\\
=&\sum_n^N E_q\left[\log p\left(w_{g,n}|z_{g,n},\phi\right)\right]+E_q\left[\log p\left(\theta_g|\alpha\right)\right]\\
&+\sum_{(n_i,n_j)} E_q\left[\log p\left(c_{n_i,n_j}|z_{g,n_i}, z_{g,n_j}, \phi_{z_{n_i}}, \phi_{z_{n_j}}\right)\right]\\
&+\sum_k^K E_q\left[\log p\left(\phi_k|\beta\right)\right]+\sum_n^N E_q\left[\log p\left(z_{g,n}|\theta_g\right)\right]\\
&-\sum_k^K E_q\left[\log q_\phi\left(\phi_k|\kappa_k\right)\right]-\sum_n^N E_q\left[\log q_z\left(z_{g,n}|\varphi_{g,n}\right)\right]\\
&-E_q\left[\log q_\theta\left(\theta_g|\gamma_g\right)\right]
\end{aligned} \tag{18}$$

Since many parts of Eq. 18 are similar with other topic models, we just focus on the third part of this equation which is related to the relations between keywords.

$$\begin{aligned}
&\sum_{(n_i,n_j)} E_q\left[\log p\left(e_{n_i,n_j}|z_{g,n_i}, z_{g,n_j}, \phi_{z_{n_i}}, \phi_{z_{n_j}}\right)\right]\\
=&\sum_{(n_i,n_j)} E_q\left[\log\left(\phi_{z_{g,n_i}}\circ\phi_{z_{g,n_j}}\right)\right]\\
\approx&\sum_{(n_i,n_j)} E_q\left[\zeta^{-1}\sum_k^K \phi_{z_{g,n_i},k}\cdot\phi_{z_{g,n_j},k}+\log\zeta-1\right]\\
=&\sum_{(n_i,n_j)}\left(\zeta^{-1}\kappa_{k,n_i}^{\varphi_{g,n_i,k}}\cdot\kappa_{k,n_j}^{\varphi_{g,n_j,k}}+\log\zeta-1\right)
\end{aligned} \tag{19}$$

In Eq. 19, we use the Taylor expansion for function $log(x)$ in order to get the expectation of this function and ζ is the expansion point.

Junyu Xuan received the bachelor's degree in 2008 from China University of Geosciences, Beijing. Currently, he is working toward the dual-doctoral degree in both Shanghai University, China (SHU) and University of Technology, Sydney (UTS), Australia. His main research interests include Machine Learning, Complex Network, and Web Mining.

Jie Lu is a full professor and Head of School of Software at the University of Technology, Sydney. Her research interests lie in the area of decision support systems and uncertain information processing. She has published five research books and 270 papers, won five Australian Research Council discovery grants and 10 other grants. She received a University Research Excellent Medal in 2010. She serves as Editor-In-Chief for Knowledge-Based Systems (Elsevier), Editor-In-Chief for International Journal of Computational Intelligence Systems (Atlantis), editor for book series on Intelligent Information Systems (World Scientific) and guest editor of six special issues for international journals, as well as delivered six keynote speeches at international conferences.

Guangquan Zhang is an associate professor in Faculty of Engineering and Information Technology at the University of Technology Sydney (UTS), Australia. He has a PhD in Applied Mathematics from Curtin University of Technology, Australia. He was with the Department of Mathematics, Hebei University, China, from 1979 to 1997, as a Lecturer, Associate Professor and Professor. His main research interests lie in the area of multi-objective, bilevel and group decision making, decision support system tools, fuzzy measure, fuzzy optimization and uncertain information processing. He has published four monographs, four reference books and over 200 papers in refereed journals and conference proceedings and book chapters. He has won four Australian Research Council (ARC) discovery grants and many other research grants.

Xiangfeng Luo is a professor in the School of Computers, Shanghai University, China. Currently, he is a visiting professor in Purdue University. He received the master's and PhD degrees from the Hefei University of Technology in 2000 and 2003, respectively. He was a postdoctoral researcher with the China Knowledge Grid Research Group, Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), from 2003 to 2005. His main research interests include Web Wisdom, Cognitive Informatics, and Text Understanding. He has authored or co-authored more than 50 publications and his publications have appeared in IEEE Trans. on Automation Science and Engineering, IEEE Trans. on Systems, Man, and Cybernetics-Part C, IEEE Trans. on Learning Technology, Concurrency and Computation: Practice and Experience, and New Generation Computing, etc. He has served as the Guest Editor of ACM Transactions on Intelligent Systems and Technology. Dr. Luo has also served on the committees of a number of conferences/workshops, including Program Co-chair of ICWL 2010 (Shanghai), WISM 2012 (Chengdu), CTUW2011 (Sydney) and more than 40 PC members of conferences and workshops.