Detecting Communities with Multiplex Semantics by Distinguishing Background, General and Specialized Topics

Di Jin, Kunzeng Wang, Ge Zhang, Pengfei Jiao, Dongxiao He, Francoise Fogelman-Soulié, Xin Huang

Abstract—Finding semantic communities using network topology and contents together is a hot topic in community detection. Existing methods often use word attributes in an indiscriminate way to help finding communities. Through the analysis we find that, words in networked contents often embody a hierarchical semantic structure. Some words reflect a background topic of the whole network with all communities, some imply the high-level general topic covering several topic-related communities, and some imply the high-resolution specialized topic to describe each community. Ignoring such semantic structures often leads to defects in depicting networked contents where deep semantics are not fully utilized. To solve this problem, we propose a new Bayesian probabilistic model. By distinguishing words from either a background topic or some two-level topics (i.e. general and specialized topics), this model not only better utilizes the networked contents to help finding communities, but also provides a clearer multiplex semantic community interpretation. We then give an efficient variational algorithm for model inference. The superiority of this new approach is demonstrated by comparing with ten state-of-the-art methods on nine real networks and an artificial benchmark. A case study is further provided to show its strong ability in deep semantic interpretation of communities.

Index Terms— Community Detection, Bayesian Probabilistic Model, Multiplex Semantics, Variational Inference

1 Introduction

A complex system, typically composed of many components interacting with each other, can be modeled as a complex network. Community detection, one of the most important tasks in network analysis, can be applied to many areas such as the detection of terrorist groups, targeted advertising, document clustering, and so on. In addition, community detection can also be used to promot other network analysis tasks, e.g. link prediction, influence maximization, etc.

The traditional methods that use network topology to find communities typically suppose that the links within communities are dense while those between communities are sparse. These methods can be mainly divided into two categories. The first is the heuristic-based methods, including spectral algorithms [1], [2], dynamic methods [3], [4] and modularity optimization methods [5], [6]. The second is the model-based methods which mainly depends on the probabilistic modelling and statistical inference [7], [8]. However, with the expansion of network sizes, the noise in networks is often unavoidable. To further improve community detection performance on real networks with noise, some researchers have proposed several algorithms [9], [10] that integrate both network topology and semantic (or textual) contents of networks, when such content is available. By introducing the content information, these methods can not only improve community detection performance but also find the semantic interpretation of communities, which typically refers to topics that represent the functions of communities.

The existing methods that use network topology and semantic content together typically consider that all the word attributes are equally important in exploring and explaining communities. That is, the words in the networked content are often used identically and indiscriminately to help distinguish communities. However, the difference of the words' topic levels typically exists in real-world networks. It is often the truth that some words reflect the common topic information of the whole network with all communities and do not help distinguish communities. So we call them the *background topic*. While some other words can mainly help distinguish communities, they may also embody different levels. On the one hand, some words reflect the commonness of several topic-related communities which form a high-level topic covering multi-communities. We call them the *general topic*. On the other hand, some words reflect the high-resolution semantics of each community. We call them the *specialized topic*.

Take a paper citation network as an example (Fig. 1). In the network, each node represents a paper and each link represents the citation relationships between two papers. The background topic words of the whole network, e.g. abstract, introduction and conclusion, reflect the common information of all the scientific papers (the top level of Fig. 1). Furthermore, we select a typical node in this citation network, e.g. a classical network community detection article [11] written by Girvan and Newman, and analyze its semantic contents. Through a statistical analysis of the

- *D. Jin, K. Wang, G. Zhang, P. Jiao, D. He and F. Fogelman-Soulié are with College of Intelligence and Computing, Tianjin University, Tianjin, China. E-mail: {jindi, wangkunzeng, zge2016, pjiao, hedongxiao}@tju.edu.cn, francoise.soulie@outlook.com*
- *X. Huang is with Department of Computer Science, Hong Kong Baptist University, Hong Kong, China. (Corresponding author) Email: xinhuang@comp.hkbu.edu.hk*

xxxx-xxxx/0x/$xx.00 © 200x IEEE Published by the IEEE Computer Society

topic words of this paper we find that, besides the background topic, there also exists an obvious two-level topic structure. To clearly reflect the differences between these two levels of topic words, we divide them into two word clouds, denoting general topics and specialized topics, respectively. The general topic words refer to a large area of complex network analysis (the middle level of Fig. 1). While, the core focus of the article [11] is to present a novel perspective on the analysis of complex networks, i.e. community detection, which is exhibited by the specialized topic words (the bottom level of Fig. 1). In this case, the general and specialized topics work together to form the basic semantics of this article. Also of note, a general topic can derive several specialized topics though they share different levels of descriptive words. For example, the word "network" belongs to a general topic only, while "community" can only belong to a specialized topic under this general topic, and they cannot switch. However, the existing algorithms typically did not consider this natural multiplex semantics hidden in the networked contents, leading to that this rich language information is not fully utilized to help detect and profile communities.

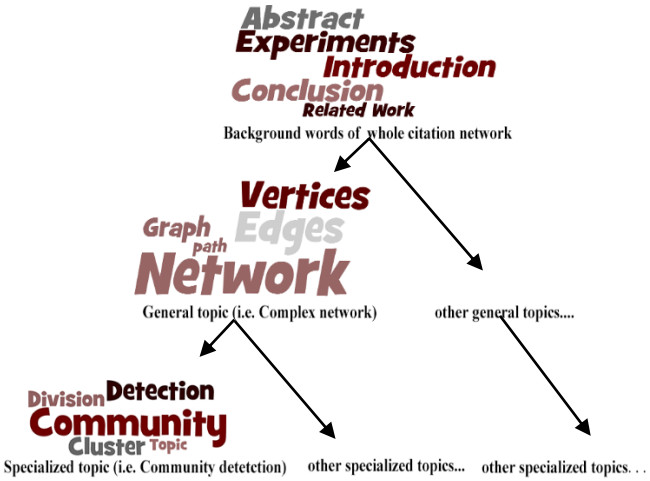


Fig. 1. A word cloud for the background words of the whole citation network and the two-level topic words of article [11] written by Girvan and Newman. From top to bottom is the word cloud of the words from the background topic of the whole citation network, general topics and specialized topics of article [11]. In word clouds of general and specialized topics, the word size represents the frequency of this word in the paper. Each general topic derives multiple specialized topics.

To solve this problem, we propose a novel probabilistic generative model in the paper. By sampling the word attributes from either the background topic, or the general or specialized topics, we model the networked contents with multiplex semantics. We then model the network topology with communities by assuming that the nodes within the same community have the same (and degree preserving) link pattern to connect with the rest of the network. And meawhile, we introduce a two-step state transition mechanism to describe the latent relationships between communities and the multiplex semantics. Through this new modelling strategy, our model can not only better utilize the rich language information in networked contents to help find communities, but also profile each community more clearly using the natural hierarchical semantics. We give an efficient variational inference algorithm to learn the model.

The contributions of this work are as follows:

1) We observe that word attributes in networked contents may come from different topic levels and play different roles in helping find and profile communities. We give a new model, by distinguishing a background topic and the general and specialized topics of words, to describe the semantic contents. The hierarchical use of contents enables the new model to not only find communities with similar interests, but also provide background semantic interpretation for the whole network contents and the two-level semantic interpretation to each community.

2) We give a Bayesian treatment on the model to detect communities with informative explanations. We transform the model inference into a problem of maximum a posteriori (MAP) learned by an efficient optimization algorithm based on variational Bayesian inference.

3) The superior performance of this new approach has been tested on nine real networks and an artificial benchmark, by comparing with ten baselines. We further demonstrate its strong interpretation ability through a case study analysis, by considering the background topic of the network and the two-level semantics of communities.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 presents our Bayesian model and generative process for semantic community detection. Section 4 presents the efficient variational inference algorithm. Section 5 reports experimental results. Finally, Section 6 concludes and highlights this paper.

2 Related Work

A large variety of community detection algorithms have been proposed in the past decades. Conventional community detection algorithms focus on the network topologies, including spectral partition [1], [2], Markov dynamics [3], [4], modularity optimization [5], [6], statistical inference [7], [8], hierarchical clustering [11], [12] and heuristic-based approaches [13], [14], [15], [16]. Especially, statistical inference-based community detection methods have been actively studied due to their solid theoretical foundation and superior performance. For example, stochastic block models (SBM) [17] are among the most prominent statistical models for community analysis in complex networks. A recent extension is the degree corrected SBM (DCSBM) [7] which incorporates a node degree correction to model degree heterogeneity. A further extension is the infinite degree corrected SBM (IDCSBM) [18] which formulates the degree corrected block model as a non-parametric Bayesian model, incorporating a parameter to control the amount of degree correction. In addition to using statistics to improve the SBM, there also exist extensions such as the hierarchical SBM (HSBM) [19] which introduces an affinity mechanism to jointly model SBMs at different levels. However, the above methods only consider the network topology and ignore the other rich information such as semantic contents which are often available in real networks.

In recent years, increasing interest revolves around node contents in networks, particularly the network content (attributes of nodes). It is believed that nodes with similar attributes are more likely to be assigned to the same

community. In this paper, we mainly focus on the case where these attributes are text-based. For content analysis, one type of popular methods is the topic model, e.g. Latent Dirichlet Allocation (LDA) [20]. For the problem of detecting communities in networks, researchers agree that the combination of network topologies and contents is certainly important. The advantages of this combination for community detection are twofold:

1) Semantic information derived from contents or text attributes may capture deep knowledge of the nature of communities and is beneficial to compensate for noisy topological information, thus it improves the performance of community detection. Several approaches have been proposed to combine network topologies and contents for community detection. For example, Sun *et al*. [21] use a user-specified set of attributes, as well as the links from different relations in heterogeneous information networks. They assume that different types of links may present different levels of semantic importance, which are learned in order to enable the effectiveness of community detection. Wu *et al*. [22] take community detection as a problem of clustering similar nodes and propose a new method, namely SAGL. SAGL merges the global importance of nodes into the local edge strength to depict the topological structure, and further combines the node attribute similarity with a self-adjusted parameter to balance the effect of topology. As a result, SAGL can have more balanced and reasonable communities. Yang *et al*. [23] introduce a unified framework that combines a popularity (and productivity) link model and a discriminative content model, which is different from the generative models for community detection. This model uses node popularities to calculate the probability whether two nodes are connected, and then incorporates content information into the link model to estimate community memberships. Zhan *et al*. [24] take the attributed graph clustering as a dynamic cluster formation game. By assuming that a balanced solution of attribute graph clustering can be found by solving a set of Nash equilibrium problems, they propose a self-learning algorithm which is to find the corresponding balanced solution of attributed graph clustering. Liu *et al*. [25] treat attribute graph clustering as a multiobjective optimization problem, which is to optimize modularity Q and the node similarity metric together. They then propose a multiobjective evolutionary algorithm, based on the above idea to find a set of Pareto optimal solutions for community detection. Pei *et al*. [26] propose a nonnegative matrix tri-factorization (NMTF) based clustering framework with three types of graph regularization in social networks. This method integrates social relations and node contents, and three types of regularizations can capture the user similarity, message similarity, and user interaction, respectively. Although the above methods can improve community detection by incorporating node contents, they do not consider the explaination of communities using this semantic information, and also ignore the hierarchical use of semantics.

2) Text contents on networks can also provide the chance of finding semantics of communities, i.e. to offer semantic explanation to each community. Semantic interpretation typically refers to finding topics which reveal functions or interests on communities. For example, Wang *et al*. [27] propose a nonnegative matrix factorization model, namely Semantic Community Identification (SCI), with two matrices, one for community memberships and the other for community attributes. This method uses the node attributes to improve the effectiveness of community detection and provides semantic interpretations to the resulting network communities as well. But SCI makes communities and topical clusters share a same set of parameters, which may be too strong an assumption. Liu *et al*. [28] treat the network as a dynamic system. By introducing the principle of content propagation, they integrate the aspects of structure and content. Then, the nature of communities can be described by analyzing the stable state of the dynamic system. But their propagation strategy assumes that members in the same community own a strong consistency, and ignores the multi-level semantics in networked contents, and thus tends to explain communities with common information. The Bayesian Attributed Graph Clustering (BAGC) method [29] is a Bayesian generative model devised to leverage the structural and attribute information in clustering an attributed graph, while avoiding the artificial design of a distance measure. But BAGC also assumes that communities are topics, which is a strong assumption. It also uses a method similar to LDA to model semantics, ignoring the difference between different topic levels embodied in semantic contents. Hu *et al*. [30] propose a new model, namely TARA, which differs from BAGC mainly in the way of modeling contents. TARA designs a three-layer hierarchical structure (node-attribute-value) to model the multivalued attributes (instead of the binary form in BAGC), and thus can utilize more information to find communities. Although the utilization of more types of contents (i.e. multivalued attributes instead of the binary form) makes TARA more powerful, the nature of its semantic representation is still similar to BAGC, and thus suffers from the same issues. To sum up, all these methods consider the semantics of communities but neglect the differences of topic levels of words existing in real life, making their semantics a mixture of different levels of topics and thus gives blurred explanations to communites.

Also of note, we have previously published a conference paper to briefly present and provide preliminary validation for the idea of community detection with hierarchical semantics [31]. Those results are significantly extended in this paper by supplementing multiplex semantic information with background topic, the improved model generative process, new mathematical derivation as well as more experimental validations. To be specific, 1) the preliminary conference paper only considers the two-level topics, i.e. the general and specialized topics. But by the observation on many real-world networked contents we find that, besides the simple two-level topics, there typically exists a type of background topic which reflects the background information of the whole network. In our previous work, however, the background information was mixed among the two-level topics, so that the boundary between these two-level topics is not clear. In the present work, we take the background topic into account in modelling the networked contents. As a result, the corpus be-

comes almost irrelevant to the background topic when describing the two-level topics of communities, and thus these topic levels become more clearly defined. This background topic can also be visualized to further benefit our understanding of the whole network. 2) Unlike the two-level topics, the background topic does not connect directly with any specific community since it reflects the semantics of the whole network with all communities. That is, one cannot use the same way of describing the two-level topics to model the new background topic. Instead, in the new model the integration of background should be independent from any community while work together with the two-level topics to generate the whole networked contents. 3) Training of the model with multiplex semantics is also more challenging since it is more complicated and can overfit easier. The framework of variational Bayesian inference itself does not easily to overfit due to its regularization property of the lower bound, while we still need to elaborately derive a new variational inference method based on this new model which incorporates this background topic.

3 The Bayesian Model

We first introduce the notations and objectives in Section 3.1 and describe the overview of the proposed new model in Section 3.2. We then present its generative process in Section 3.3. Finally, we define this model in Section 3.4.

3.1 Notations and the Problem

We consider an undirected and unweighted *attributed network* G with n nodes and m (word) attributes. We use an adjacency matrix A = $(a_{ij})_{n\times n}$ to represent the relations among the n nodes. That is, if there is an edge between nodes v_i and v_j, we have $a_{ij} = 1$, and 0 otherwise. The attribute matrix is denoted by W = $(w_{ik})_{n\times m}$. That is, if v_i has the k^{th} attribute, then $w_{ik} = 1$, and 0 otherwise.

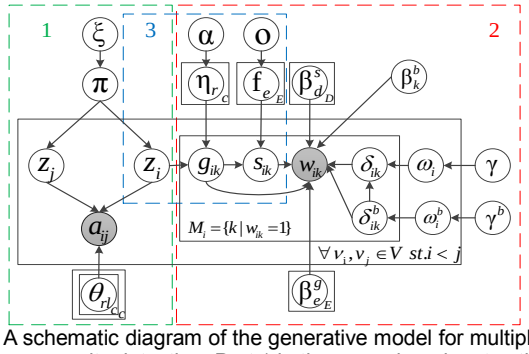


Fig. 2. A schematic diagram of the generative model for multiplex semantic community detection. Part 1 in the green box denotes the topological component describing network communities. Part 2 in the red box denotes the content component describing multiplex semantics. Part 3 in the blue box denotes the probabilistic transition mechanism connecting the two previous parts.

Given the network G, our objectives are to 1) recognize the words in the content text as generated by background topic or two-level semantic topics; 2) partition G into c node communities, E general topics and D specialized topics based on network topology and contents, 3) explore the correlation between network communities and the two-level topical clusters, 4) describe communities using both specialized topics (to show the particular interests) and general topics (to show the shared attributes of several similar communities). Though each of these four problems is technically challenging, our novel probabilistic generative model can solve all four problems at the same time.

TABLE 1
MAIN SYMBOLS USED

Types	Signs	Descriptions
X : Observed variables	A	Adjacency matrix
	W	Node-attribute matrix
	c	No. of communities
	E, D	No. of general topics, No. of specialized topics
I : Latent variables	z_i	Community assignment of node v_i
	δ^b_{ik}	If $\delta^b_{ik}=1$: w_{ik} is generated from background topic, and 0 otherwise
	δ_{ik}	1) $\delta_{ik}=0$: w_{ik} is generated from a general topic; 2) $\delta_{ik}=1$: w_{ik} is generated from a specialized topic
	g_{ik}	General topic for attribute word w_{ik}
	s_{ik}	Specialized topic for attribute word w_{ik}
Π : Model parameters with prior	ω^b_i	Parameter for generating δ^b_{ik}
	ω_i	Parameter for generating δ_{ik}
	π_r	Probability that node v_i belongs to community r
	η_{re}	Probability that v_i is in e^{th} general topic given it belongs to r^{th} community
	f_{ed}	Probability that v_i is in d^{th} specialized topic given it belongs to e^{th} general topic
Ξ : Model parameters without prior	θ_{rl}	Probability that r^{th} and l^{th} communities are connected
	β^b	Probability that background topic generates w_{ik}
	β^g_{ek}	Probability that e^{th} general topic generates w_{ik}
	β^s_{dk}	Probability that d^{th} specialized topic generate w_{ik}
Hyper-parameters	ξ, α, o	Acting as priors of the corresponding model parameters with conjugate distributions
	γ, γ^b	

3.2 Overview of the Model

To achieve the above objectives, we develop a new probabilistic generative model, i.e. Background and Two-Level Semantic Community (BTLSC). The graphical model representation of BTLSC is shown in Fig. 2, with symbols defined in Table 1. The model includes three main parts. The first one is the topological component (in green box in Fig. 2), which describes the network with community structures. By modeling that all nodes in the same community share the same (or similar) link probability when connecting to other nodes in the network, the model allows that nodes in each community own the same link pattern. The nodes' degree preservation mechanism is also introduced into this model, making it support that nodes with larger degrees are more likely to be connected. The second one is the content component (in red box in Fig. 2), which describes the multiplex semantics of networked contents and serves as the *core* of this model. By allowing that each word is sampled from either a background topic, or a general or specialized topic, we generate the textual contents of networks with multiplex semantics which is often observed in real life. The third one is the transition component (in blue box in Fig. 2), which connects the previous two parts. It describes the probabilistic transitions from communities to general topics, and further to specialized topics. Through these two state transitions, the model can not only describe the latent relationships between communities, general topics as well as specialized topics, but also becomes robust even when their relationships are complicated and do not

match well. Each of these three parts owns its particular advantage by itself to describe the real-world networked data while we seamlessly incorporate them altogether. As a result, the new model naturally describes network communities with multiplex semantics and embodies richer language and topological information.

3.3 The Generative Process

We then give the specific generative process of the BTLSC model below. Step 1 to 4(b) tell how we generate model parameters using the fixed hyper-parameters, which will be described in details in section 3.3.1. Steps 4(c) to 4(e3.3) serve as the *core* of this model, which tells how we generate the observed and latent quantities using model parameters derived, with detailed explanations in section 3.3.2.

1. Sample $\pi \sim \text{Dirichlet}(\xi)$
2. For each community $r \in \{1,2,...,c\}$:
   (a)Sample $\eta_r \sim \text{Dirichlet}(\alpha)$
3. For each general topic $e \in \{1,2,...,E\}$:
   (a)Sample $f_e \sim \text{Dirichlet}(o)$
4. For each node v_i with $i \in \{1,2,...,n\}$:
   (a)Sample $\omega_i^b \sim \text{Beta}(\gamma_0^b, \gamma_1^b)$
   (b)Sample $\omega_i \sim \text{Beta}(\gamma_0, \gamma_1)$
   (c)Sample community assignment $z_i \sim \text{Multinomial}(\pi)$
   (d)For each node v_j with $j \in \{i+1, i+2,...,n\}$:
      (d.1) Sample edge $a_{ii} \sim \text{Bernoulli}(d_i d_j \theta_{z_i z_j})$
   (e)For each word w_{ik} with $k \in \{1,2,...,m\}$:
      Sample $\delta_{ik}^b \sim \text{Bernoulli}(\omega_i^b)$
      -if δ_{ik}^b=1
        (e.1) Sample $w_{ik} \sim \text{Multinomial}(\beta^b)$
      -else if δ_{ik}^b=0
        (e.2) Sample $g_{ik} \sim \text{Multinomial}(\eta_{z_i})$
        (e.3) Sample $\delta_{ik} \sim \text{Bernoulli}(\omega_i)$
          -if $\delta_{ik} = 0$
            (e.3.1) Sample $w_{ik} \sim \text{Multinomial}(\beta_{g_{ik}}^g)$
          -else if $\delta_{ik} = 1$
            (e.3.2) Sample $s_{ik} \sim \text{Multinomial}(f_{g_{ik}})$
            (e.3.3) Sample $w_{ik} \sim \text{Multinomial}(\beta_{s_{ik}}^s)$

3.3.1 Generate Parameters with Conjugate Prior

We take a Bayesian treatment on the model generation process. Instead of assuming a fixed value of each parameter in set Π, we treat ω^b, ω, π, H and F as random variables and place conjugate *prior* distributions on them. We then introduce how to generate these model parameters using hyper-parameters ξ, α, o, γ^b and γ. In the generative process, all the hyper-parameters are set a predefined value, such as that suggested in LDA.

1. We first use a Dirichlet distribution to generate model parameter $\pi = (\pi_1, \pi_2, ..., \pi_r)$ (step 1), where π_r represents the probability that node v_i belongs to community r, subject to $\pi_r \in [0,1]$ and $\sum_{r=1}^{c} \pi_r = 1$. Then, this Dirichlet distribution can be defined as:

$$p(\pi \mid \xi) = \frac{\Gamma(\sum_{r=1}^{c} \xi_r)}{\prod_{r=1}^{c} \Gamma(\xi_r)} \prod_{r=1}^{c} \pi_r^{\xi_r - 1}, \quad (1)$$

   in which $\Gamma(\bullet)$ is the Gamma function. This distribution is parameterized by the hyper-parameter, a positive real c-dimensional vector $\xi=(\xi_1, \xi_2, ..., \xi_r)$. The choice of Dirichlet distribution on π (and also the distribution on ω^b, ω, H and F below) is not arbitrary. We will give the reason at the end of subsection 3.3.
2. We also use Dirichlet distribution to generate the matrix of parameters $H = (\eta_{re})_{c \times E}$ (step 2(a)), where each row η_r represents the distribution of general topics over community r. H can be also taken as a probabilistic transition matrix from communities to general topics, subject to $\eta_{re} \in [0,1]$ and $\sum_{e=1}^{E} \eta_{re} = 1$. The distribution is then defined as:

$$p(\eta_r \mid \alpha) = \frac{\Gamma(\sum_{e=1}^{E} \alpha_e)}{\prod_{e=1}^{E} \Gamma(\alpha_e)} \prod_{e=1}^{E} \eta_{re}^{\alpha_e - 1}. \quad (2)$$

   Hyper-parameter $\alpha = (\alpha_1, \alpha_2, ..., \alpha_E)$ is an E-dimensional vector. All communities share the same α.
3. Similar to H, we also use the Dirichlet distribution to generate $F = (f_{ed})_{E \times d}$ (step 3(a)). F is a matrix of probabilistic transition from general topics to specialized topics, where each row f_e is the specialized topic distribution over general topic e, subject to $f_{ed} \in [0,1]$ and $\sum_{d=1}^{D} f_{ed} = 1$. Then the density function is given by:

$$p(f_e \mid o) = \frac{\Gamma(\sum_{d=1}^{D} o_d)}{\prod_{d=1}^{D} \Gamma(o_d)} \prod_{d=1}^{D} f_{ed}^{o_d - 1}, \quad (3)$$

   where the hyper-parameter $o=(o_1, o_2, ..., o_d)$ is a D-dimensional vector, shared by all the general topics.
4. We use a Beta distribution to generate model parameters $\omega^b = (\omega_1^b, \omega_2^b, ..., \omega_n^b)$ (step 4 (a)), where ω_i^b is the parameter of Bernoulli distribution. By using the Bernoulli distribution, we can get the value of δ_{ik}^b, 0 or 1. If $\delta_{ik}^b = 1$, w_{ik} will be generated from the background topic, and 0 otherwise. The Beta distribution, with two hyper-parameters (γ_0^b and γ_1^b shared by all nodes), is defined as:

$$p(\omega_i^b \mid \gamma_0^b, \gamma_1^b) = \frac{\Gamma(\gamma_0^b + \gamma_1^b)}{\Gamma(\gamma_0^b)\Gamma(\gamma_1^b)} (\omega_i^b)^{\gamma_0^b - 1} (1 - \omega_i^b)^{\gamma_1^b - 1}. \quad (4)$$

5. We also use a Beta distribution to generate model parameters $\omega = (\omega_1, \omega_2, ..., \omega_n)$ (step 4 (b)), where ω_i is the parameter of Bernoulli distribution. Through this distribution, we can get the value of δ_{ik}, 0 or 1. If $\delta_{ik} = 0$, w_{ik} will be generated from a general topic. If $\delta_{ik} = 1$, w_{ik} will be generated from a specialized topic. This Beta distribution with hyper-parameters (γ_0 and γ_1) is defined as:

$$p(\omega_i \mid \gamma_0, \gamma_1) = \frac{\Gamma(\gamma_0 + \gamma_1)}{\Gamma(\gamma_0)\Gamma(\gamma_1)} (\omega_i)^{\gamma_0 - 1} (1 - \omega_i)^{\gamma_1 - 1}. \quad (5)$$

3.3.2 Generate Observed and Latent Quantities

After the model parameters (with conjugate prior) have been generated, we then use all the parameters to generate the observed and latent quantities, which is the key to the generative process of this model.

1. First, we sample the community label z_i of each node v_i independently from a multinomial distribution (step 4(c)), which is defined as:

$$p(z_i = r \mid \pi) = \pi_r, \quad r = 1,2,...,c. \quad (6)$$

   We have shown how to sample π (and also H, F, ω^b, ω) from a conjugate prior distribution in the previous subsection. So here, without loss of generality, we assume that those parameters are given in advance.

2. Assume that z_i and z_j are community labels of nodes v_i and v_j, which have been sampled in the above step. We then sample each edge a_{ij} between v_i and v_j from a Bernoulli distribution (step 4(d.1)), defined as:

$$p(a_{ij} \mid d_i d_j \theta_{z_i z_j}) = (d_i d_j \theta_{z_i z_j})^{a_{ij}} (1 - d_i d_j \theta_{z_i z_j})^{1-a_{ij}}, \quad (7)$$

where a_{ij} is a binary variable value, 0 or 1. It describes the fitting of the model to the network topology from a degree-corrected stochastic block model [7], in which $\Theta = (\theta_{rl})_{c \times c}$ serves as the block matrix and d_i is the degree of node v_i. This model typically performs well in fitting network topology with community structures.

3. To determine whether each attribute w_{ik} of node v_i is generated from the background topic, we bring in a binary variable δ_{ik}^b from a Bernoulli distribution, parameterized by ω_i^b (step 4(e)). It is then defined as:

$$p(\delta_{ik}^b \mid \omega_i^b) = (\omega_i^b)^{\delta_{ik}^b} (1 - \omega_i^b)^{1-\delta_{ik}^b}. \quad (8)$$

The value of δ_{ik}^b indicates the generative process of words in the next step. If $\delta_{ik}^b = 1$, w_{ik} is a background topic word. In this case, we can generate this word directly (step 4(e.1)) with the following distribution:

$$p(w_{ik} \mid \beta^b) = (\beta_k^b)^{w_{ik} \delta_{ik}^b}, \quad (9)$$

where $\beta^b = (\beta_k^b)_{1 \times m}$, in which β_k^b denotes the probability that the k^{th} word attribute is generated from the background topic, which is irrelevant to any node v_i, subject to $\sum_{k=1}^m \beta_k^b = 1$ and $\beta_k^b \in [0,1]$. In this situation, the whole generative process will end here.

4. However, if $\delta_{ik}^b = 0$, which means that w_{ik} is not generated from the background topic, the generation process will continue. Let we have got the community assignment z_i of each node v_i, then we should sample the general topic assignment g_{ik} of word attribute w_{ik} of this node v_i via a multinomial distribution (step 4(e.2)), which is defined as:

$$p(g_{ik} = e \mid \eta_{z_i}) = (\eta_{z_i, e})^{w_{ik}(1-\delta_{ik}^b)}, \quad (10)$$

where $\eta_{z_i, e}$ denotes the probability that community z_i selects the e^{th} general topic, which meets $\sum_{e=1}^E \eta_{z_i, e} = 1$ and $\eta_{z_i, e} \in [0,1]$, for $z_i = 1...c$. The meaning of H has been explained in step 2 in the previous subsection.

5. Thereafter, to determine whether the word attribute w_{ik} of node v_i is generated from a general or a specialized topic, we use a binary variable δ_{ik}, sampled from a Bernoulli distribution parameterized by ω_i, as a indicator (step 4(e.3)). It is defined as:

$$p(\delta_{ik} \mid \omega_i) = [\omega_i^{\delta_{ik}} (1 - \omega_i)^{1-\delta_{ik}}]^{(1-\delta_{ik}^b)}. \quad (11)$$

Then, the successive generative process will be determined by the value of δ_{ik}. That is,

1) If $\delta_{ik} = 0$, w_{ik} will be generated from a general topic. Recall that in step 3, we have identified this general topic. So here we just need to generate the word attribute w_{ik} of node v_i. We sample each word from a multinomial distribution (step 4 (e.3.1)), defined as:

$$p(w_{ik} \mid \beta_{g_{ik}}^g) = (\beta_{g_{ik}, k}^g)^{w_{ik}(1-\delta_{ik}^b)(1-\delta_{ik})}, \quad (12)$$

where $\mathrm{B}^g = (\beta_{ek}^g)_{E \times m}$, in which $\beta_{ek}^g = p(w_{ik} = 1 \mid g_{ik} = e)$ denotes the probability that the k^{th} word attribute is generated from the e^{th} general topic, which is irrelevant to any node v_i and meets $\sum_{k=1}^m \beta_{g_{ik}, k}^g = 1$ and $\beta_{g_{ik}, k}^g \in [0,1]$, for $g_{ik} = 1...E$.

2) In contrast, if $\delta_{ik} = 1$, w_{ik} will be generated from a specialized topic, given the general topic label g_{ik} of w_{ik}. Then, first, we need to sample the specialized topic from a multinomial distribution (step 4(e.3.2)):

$$p(s_{ik} = d \mid \mathrm{f}_{g_{ik}}) = (f_{g_{ik}, d})^{w_{ik}(1-\delta_{ik}^b)\delta_{ik}}, \quad (13)$$

where $f_{g_{ik}, d}$ denotes the probability that the general topic g_{ik} selects the d^{th} specialized topic, which meets $f_{g_{ik}, d} \in [0,1]$ and $\sum_{d=1}^D f_{g_{ik}, d} = 1$, for $g_{ik} = 1...E$. The specific meaning of F has been explained in step 3 in the previous subsection. We then need to generate the attribute w_{ik} of node v_i from a multinomial distribution (step 4(e.3.3)), defined as:

$$p(w_{ik} \mid \beta_{s_{ik}}^s) = (\beta_{s_{ik}, k}^s)^{w_{ik}(1-\delta_{ik}^b)\delta_{ik}}, \quad (14)$$

where $\mathrm{B}^s = (\beta_{dk}^s)_{D \times m}$, in which $\beta_{dk}^s = p(w_{ik} = 1 \mid s_{ik} = d)$ denotes the probability of the k^{th} attribute of node v_i being negated by the d^{th} specialized topic, subject to $\sum_{k=1}^m \beta_{s_{ik}, k}^s = 1$ and $\beta_{s_{ik}, k}^s \in [0,1]$, for $s_{ik} = 1...D$.

Also of note, the choice of Dirichlet and Beta distributions as priors to π, H, F, ω^b and ω here are not arbitrary. They are conjugate priors to multinomial and Bernoulli distributions, respectively. This will give a closed-form expression for the posterior and provide mathematical convenience when we derive inference on the Bayesian model.

3.4 The Model Definition

Based on the above generative process, we then give the formulation of this Bayesian model (which represents the underlying joint probability distribution) as follows:

$$P(\mathrm{A}, \mathrm{W}, \mathrm{z}, \Delta^b, \Delta, \mathrm{G}, \mathrm{S}, \pi, \mathrm{H}, \mathrm{F}, \omega^b, \omega \mid \Theta, \mathrm{B}^b, \mathrm{B}^g, \mathrm{B}^s, \xi, \alpha, \mathrm{o}, \gamma^b, \gamma) = \begin{pmatrix} p(\pi \mid \xi)\, p(\mathrm{H} \mid \alpha)\, p(\mathrm{F} \mid \mathrm{o})\, p(\omega^b \mid \gamma^b)\, p(\omega \mid \gamma) \\ p(\mathrm{z} \mid \pi)\, p(\mathrm{A} \mid \Theta, \mathrm{z})\, p(\Delta^b \mid \omega^b)\, p(\Delta \mid \omega) \\ p(\mathrm{G} \mid \mathrm{H}, \mathrm{z})\, p(\mathrm{S} \mid \mathrm{F}, \mathrm{G})\, p(\mathrm{W} \mid \mathrm{B}^g, \mathrm{G}, \mathrm{B}^s, \mathrm{S}, \Delta^b, \Delta) \end{pmatrix}, \quad (15)$$

where

$$p(\mathrm{H} \mid \alpha) = \prod_{r=1}^c p(\eta_r \mid \alpha),$$

$$p(\mathrm{F} \mid \mathrm{o}) = \prod_{e=1}^E p(\mathrm{f}_e \mid \mathrm{o}),$$

$$p(\omega^b \mid \gamma^b) = \prod_{i=1}^n p(\omega_i^b \mid \gamma_0^b, \gamma_1^b),$$

$$p(\omega \mid \gamma) = \prod_{i=1}^n p(\omega_i \mid \gamma_0, \gamma_1),$$

$$p(\mathrm{A} \mid \Theta, \mathrm{z}) = \prod_{i<j} p(a_{ij} \mid d_i d_j \theta_{z_i z_j}),$$

$$p(\Delta^b \mid \omega^b) = \prod_{i=1}^n \prod_{k=1}^m p(\delta_{ik}^b \mid \omega_i^b),$$

$$p(\Delta \mid \omega) = \prod_{i=1}^n \prod_{k=1}^m p(\delta_{ik} \mid \omega_i)^{(1-\delta_{ik}^b)},$$

$$p(\mathrm{W} \mid \mathrm{B}^b) = \prod_{i=1}^n \prod_{k=1}^m p(w_{ik} \mid \beta^b)^{w_{ik}\delta_{ik}^b},$$

$$p(\mathrm{G} \mid \mathrm{H}, \mathrm{z}) = \prod_{i=1}^n \prod_{k=1}^m p(g_{ik} \mid \eta_{z_i})^{w_{ik}(1-\delta_{ik}^b)},$$

$$p(\mathrm{W} \mid \mathrm{B}^g, \mathrm{G}) = \prod_{i=1}^n \prod_{k=1}^m p(w_{ik} \mid \beta_{g_{ik}}^g)^{w_{ik}(1-\delta_{ik}^b)(1-\delta_{ik})},$$

$$p(\mathrm{S} \mid \mathrm{F}, \mathrm{G}) = \prod_{i=1}^n \prod_{k=1}^m p(\mathrm{s}_{ik} \mid f_{g_{ik}})^{w_{ik}(1-\delta_{ik}^b)\delta_{ik}},$$

$$p(\mathrm{W} \mid \mathrm{B}^s, \mathrm{S}) = \prod_{i=1}^n \prod_{k=1}^m p(w_{ik} \mid \beta_{s_{ik}}^s)^{w_{ik}(1-\delta_{ik}^b)\delta_{ik}},$$

where the sub functions have all been defined in (1) to (14). For brevity, we will short this joint probability distribution as $P(\mathrm{A}, \mathrm{W}, \mathrm{z}, \Delta^b, \Delta, \mathrm{G}, \mathrm{S}, \pi, \mathrm{H}, \mathrm{F}, \omega^b, \omega)$ in the rest of this paper.

4 THE MODEL INFERENCE

In this section, we give an efficient variational Bayesian inference algorithm to learn the model. We first introduce the basic idea, and then show the detailed inference process. At last, we give an algorithmic procedure and the computational complexity of this algorithm.

4.1 The Basic Idea

Based on the above model, the task of clustering the observed quantities X = (A, W) can be transformed as a standard probabilistic inference problem, i.e. to find a set of parameters that can maximize the posterior probability distribution of this model, which is to find:

$$\mathrm{z}^*,\mathrm{G}^*,\mathrm{S}^*,\Delta^{b*}\Delta^* = \arg\max_{\mathrm{z,G,S},\Delta^b,\Delta} P(\mathrm{z,G,S},\Delta^b,\Delta\,|\,\mathrm{A,W}),$$

where $P(\mathrm{z,G,S},\Delta^b,\Delta\,|\,\mathrm{A,W})$ is the posterior distribution of z, G, S, Δ^b and Δ given A and W (as well as Θ, B^g, B^s, α, o, γ^b, γ, ξ). Intuitively, the optimal z^*, G^*, S^*, Δ^{b*} and Δ^* correspond to values which can best explain the adjacency matrix A and the attribute matrix W of this given network. Despite its conceptual simplicity, this probabilistic inference problem is in fact notoriously hard to solve. The posterior distribution of this model is defined as:

$$P(\mathrm{z,G,S},\Delta^b,\Delta\,|\,\mathrm{A,W},\Theta,\beta^b,\mathrm{B}^g,\mathrm{B}^s,\alpha,\mathrm{o},\gamma^b,\gamma,\xi)$$
$$= \int\int\int\int\int\left(P(\mathrm{z,G,S},\Delta^b,\Delta,\pi,\mathrm{H,F},\omega^b,\omega\,|\,\mathrm{A,W},\Theta,\beta^b,\mathrm{B}^g,\mathrm{B}^s,\alpha,\mathrm{o},\gamma^b,\gamma,\xi)\right)d\pi\; d\mathrm{H}\; d\mathrm{F}\; d\omega^{\mathrm{b}}d\omega,$$

where

$$P(\mathrm{z,G,S},\Delta^b,\Delta,\pi,\mathrm{H,F},\omega^b,\omega\,|\,\mathrm{A,W},\Theta,\beta^b,\mathrm{B}^g,\mathrm{B}^s,\alpha,\mathrm{o},\gamma^b,\gamma,\xi)$$
$$= \frac{\left(P(\mathrm{z,G,S},\Delta^b,\Delta,\pi,\mathrm{H,F},\omega^b,\omega,\ \mathrm{A,W}\,|\,\Theta,\beta^b,\mathrm{B}^g,\mathrm{B}^s,\alpha,\mathrm{o},\gamma^b,\gamma,\xi)\right)}{\sum_{\mathrm{z,G,S,}\atop\Delta^b,\Delta}\int\int\int\int\int\left(P(\mathrm{z,G,S},\Delta^b,\Delta,\pi,\mathrm{H,F},\omega^b,\omega,\ \mathrm{A,W}\,|\,\Theta,\beta^b,\mathrm{B}^g,\mathrm{B}^s,\alpha,\mathrm{o},\gamma^b,\gamma,\xi)\right)d\pi d\mathrm{H}d\mathrm{F}d\omega^b d\omega},$$

which is shorten as $P(\mathrm{z,G,S},\Delta^b,\Delta,\pi,\mathrm{H,F},\omega^b,\omega\,|\,\mathrm{A,W})$ for brevity. But due to the integrals over model parameters π, H, F, ω^b and ω, it does not have a closed-form expression.

Since the calculation of the true posterior distribution is intractable, we develop an efficient variational algorithm to solve this probabilistic inference problem. The basic idea is to approximate our objective of the true posterior distribution $P(\mathrm{z,G,S},\Delta^b,\Delta,\pi,\mathrm{H,F},\omega^b,\omega\,|\,\mathrm{A,W})$ by a new variational distribution q. Here we restrict the variational distribution q to a family of distributions that factorize as:

$$q(\mathrm{z},\Delta,\Delta^b,\mathrm{G,S},\pi,\mathrm{H,F},\omega^b,\omega\,|\,\tilde{\Phi},\tilde{\mathrm{T}},\tilde{\mathrm{T}}^b,\tilde{\mathrm{P}},\tilde{\Sigma},\tilde{\xi},\tilde{\mathrm{A}},\tilde{\mathrm{O}},\tilde{\mathrm{R}}^b,\tilde{\mathrm{R}})$$
$$= \begin{pmatrix} q(\mathrm{z}\,|\,\tilde{\Phi})q(\Delta\,|\,\tilde{\mathrm{T}})q(\Delta^b\,|\,\tilde{\mathrm{T}}^b)q(\mathrm{G}\,|\,\tilde{\mathrm{P}})q(\mathrm{S}\,|\,\tilde{\Sigma}) \\ q(\pi\,|\,\tilde{\xi})q(\mathrm{H}\,|\,\tilde{\mathrm{A}})q(\mathrm{F}\,|\,\tilde{\mathrm{O}})q(\omega^b\,|\,\tilde{\mathrm{R}}^b)q(\omega|\,\tilde{\mathrm{R}}) \end{pmatrix}, \quad (16)$$

where $\tilde{\Phi}$, $\tilde{\mathrm{T}}$, $\tilde{\mathrm{T}}^b$, $\tilde{\mathrm{P}}$, $\tilde{\Sigma}$, $\tilde{\xi}$, $\tilde{\mathrm{A}}$, $\tilde{\mathrm{O}}$, $\tilde{\mathrm{R}}^b$ and $\tilde{\mathrm{R}}$ are the *variational parameters*. This definition of the family of variational distributions is not arbitrary. In fact, the sub distributions in (16) take exactly the same parametric forms as the sub functions in (1) to (14). (Detailed definitions of these sub distributions are given in Appendix A.) In addition, the variational parameters are free to vary, while the hyper-parameters are fixed throughout the generative process. For simplicity, we will abbreviate this variational distribution as $q(\mathrm{z},\Delta,\Delta^b,\mathrm{G,S},\pi,\mathrm{H,F},\omega^b,\omega)$ in the following.

Supposing that we find the variational distribution q which is most similar to the true posterior distribution, we can then find the communities by:

$$\begin{aligned}
\mathrm{z}^* &= \arg\max_{\mathrm{z}} P(\mathrm{z}\,|\,\mathrm{A,W}) \\
&= \arg\max_{\mathrm{z}} \sum_{\mathrm{G,S}\atop\Delta^b,\Delta}\int\ldots\int\left(P(\mathrm{z,G,S},\Delta^b,\Delta,\ \pi,\mathrm{H,F},\omega^b,\omega|\mathrm{A,W})\right)d\pi\; d\mathrm{H}\; d\mathrm{F}\; d\omega^b d\omega \\
&\simeq \arg\max_{\mathrm{z}} \sum_{\mathrm{G,S}\atop\Delta^b,\Delta} q(\mathrm{z},\Delta^b,\Delta,\mathrm{G,S},\pi,\mathrm{H,F},\omega^b,\omega)d\pi\; d\mathrm{H}\; d\mathrm{F}\; d\omega^b d\omega \\
&= \prod_{i=1}^{n}\arg\max_{\mathrm{z}} q(\mathrm{z}_i\,|\,\tilde{\varphi}_{\mathrm{i}}) \\
&= \prod_{i=1}^{n}\arg\max_{r}\tilde{\varphi}_{ir}.
\end{aligned}$$

Similar to the way of deriving the MAP of the latent quantity z^*, we can also get the background topic as well as the general and specialized topics easily.

4.2 Optimizing Variational Parameters

Recall that our goal is to find the variational distribution q in the family that is closest to the true posterior distribution $P(\mathrm{z,G,S},\Delta^b,\Delta,\pi,\mathrm{H,F},\omega^b,\omega|\mathrm{A,W})$. This is now equivalent to optimizing variational parameters $\tilde{\Phi}$, $\tilde{\mathrm{T}}$, $\tilde{\mathrm{T}}^b$, $\tilde{\mathrm{P}}$, $\tilde{\Sigma}$, $\tilde{\xi}$, $\tilde{\mathrm{A}}$, $\tilde{\mathrm{O}}$, $\tilde{\mathrm{R}}^b$ and $\tilde{\mathrm{R}}$ with respect to some suitable distance measure. To measure the distance between the variational distribution $q(\mathrm{z},\Delta,\Delta^b,\mathrm{G,S},\pi,\mathrm{H,F},\omega^b,\omega)$ and the true posterior distribution $P(\mathrm{z,G,S},\Delta^b,\Delta,\pi,\mathrm{H,F},\omega^b,\omega\,|\,\mathrm{A,W})$, we can adopt the Kullback-Leibler (*KL*) divergence [32] which is commonly used in information theory, defined as:

$$KL(q\,\|\,P) = \sum_{\mathrm{z,G,S,}\atop\Delta^b,\Delta}\int\int\int\int\int\left(q(\mathrm{z},\Delta^b,\Delta,\mathrm{G,S},\pi,\mathrm{H,F},\omega^b,\omega)\times \log\frac{q(\mathrm{z},\Delta^b,\Delta,\mathrm{G,S},\pi,\mathrm{H,F},\omega^b,\omega)}{P(\mathrm{z,G,S},\Delta^b,\Delta,\pi,\mathrm{H,F},\omega^b,\omega)}\right)d\pi d\mathrm{H}d\mathrm{F}d\omega^b d\omega, \quad (17)$$

which is a function of variational parameters ($\tilde{\Phi}$, $\tilde{\mathrm{T}}$, $\tilde{\mathrm{T}}^b$, $\tilde{\mathrm{P}}$, $\tilde{\Sigma}$, $\tilde{\xi}$, $\tilde{\mathrm{A}}$, $\tilde{\mathrm{O}}$, $\tilde{\mathrm{R}}^b$ and $\tilde{\mathrm{R}}$) and model parameters (Θ, B^g and B^s). Our objective now becomes finding the optimal variational parameters that can minimize his KL divergence. However, this problem is also infeasible since true posterior distribution $P(\mathrm{z,G,S},\Delta^b,\Delta,\pi,\mathrm{H,F},\omega^b,\omega\,|\,\mathrm{A,W})$ is exactly what we strive to approximate in the first place. So, instead of directly minimizing this *KL* divergence, we solve an equivalent maximization problem, defined as:

$$\tilde{L}(q) = \sum_{\mathrm{z,G,S,}\atop\Delta^b,\Delta}\int\int\int\int\int\left(q(\mathrm{z},\Delta^b,\Delta,\mathrm{G,S},\pi,\mathrm{H,F},\omega^b,\omega)\times \log\frac{P(\mathrm{z,G,S},\Delta^b,\Delta,\pi,\mathrm{H,F},\omega^b,\omega,\mathrm{A,W})}{q(\mathrm{z},\Delta^b,\Delta,\mathrm{G,S},\pi,\mathrm{H,F},\omega^b,\omega)}\right)d\pi d\mathrm{H}d\mathrm{F}d\omega^b d\omega. \quad (18)$$

The equivalence between these two optimization problems can be easily derived by noticing that they sum up to a constant for a given network:

$$KL(q\,\|\,P) + \tilde{L}(q) = \log P(\mathrm{A,W}).$$

Then, to maximize the objective function $\tilde{L}(q)$, we need to take the derivatives of $\tilde{L}(q)$ with respect to variational parameters ($\tilde{\Phi}$, $\tilde{\mathrm{T}}$, $\tilde{\mathrm{T}}^b$, $\tilde{\mathrm{P}}$, $\tilde{\Sigma}$, $\tilde{\xi}$, $\tilde{\mathrm{A}}$, $\tilde{\mathrm{O}}$, $\tilde{\mathrm{R}}^b$ and $\tilde{\mathrm{R}}$) and model parameters without priors (Θ, B^g and B^s), and set these derivatives to zeroes, since it has a closed-form expression. Then, we get the expressions of the parameters which need to be updated. But here for clarity, we show the detailed procedure of derivations and the expression of parameters to be updated in Appendix B.

4.3 Algorithm Summary and Complexity Analysis

At last, we give the algorithmic procedure of BTLSC below. When it converges, we will get the variational parameters $\tilde{\Phi}$, $\tilde{\mathrm{P}}$ and $\tilde{\Sigma}$, and can use them to calculate model parameters H and F. We can then derive the community assignments of nodes using $\tilde{\Phi}$, the relationship between communities and general topics using H, the relationship between general and specialized topics using F, as well as the relationship between communities and specialized topics using H × F (i.e. the matrix product of H and F). We can also find the background topic, general topics as well as the specialized topics using β^b, B^g and B^s, respectively.

Considering the sparsity of the networked data, the computational complexity of this new algorithm BTLSC is $O(T(n^2c^2 + fcE + nmED + ec^2 + f))$, where n, e, m, f, c, E, D, and T are the numbers of nodes, links, word attributes, non-zero node-attribute pairs (in attributed matrix), communities, general topics, specialized topics, and iterations for convergence. Given a large sparse network, we often have $O(n) = O(e) = O(f)$. In general, T, c, E and D can also be taken as constants compared with the network sizes. In this case, the computational complexity of our algorithm can be simplified as $O(n^2 + nm)$. This can be further reduced to near linear via some speedup techniques such as stochastic optimization as proposed and used in [33].

Algorithm 1: Iterative optimization procedure

Input: A, W, c, E, D, the convergence threshold κ, and the maximum number of iterations $count_{\max}$

Output: $\tilde{\Phi}$, H, F, β^b, B^g, B^s

1. Randomly initialize the variational parameters in Π and the model parameters in Ξ
2. Set $count$ =1
3. **repeat:**
   (a) update $\tilde{\xi}$, $\tilde{\Phi}$, $\tilde{\mathrm{P}}$, $\tilde{\Sigma}$, $\tilde{\mathrm{A}}$, $\tilde{\mathrm{O}}$, $\tilde{\mathrm{T}}^b$, $\tilde{\mathrm{T}}$, $\tilde{\mathrm{R}}^b$, $\tilde{\mathrm{R}}$, Θ, β^b, B^g and B^s via (B.6) to (B.21) in Appendix B
   (b) compute $\tilde{L}(q^{(count)})$
   (c) $count$=$count$+1
   Until $\tilde{L}(q^{(count)}) - \tilde{L}(q^{(count-1)}) < \kappa$ **or** $count > count_{\max}$
4. Calculate H and F using $\tilde{\Phi}$, $\tilde{\mathrm{P}}$ and $\tilde{\Sigma}$ derived above

5 Experiments

We first introduce the experiment setup which includes the datasets and performance metrics, and then evaluate the effectiveness of our algorithm in comparison with ten state-of-the-art community detection methods on nine real-world networks. Thereafter, we test the scalability of this algorithm on real and artificial datasets. And finally, we use an online music system to assess its interpretability.

5.1 Experiment Setup

Datasets. We use nine real-world networks with known communities for the comparison of BTLSC with other methods in terms of both effectiveness and efficiency. We also use the LASTFM dataset [34] from *Last.fm*, a famous British online music service, in which each user is described by 11,946 attributes including a list of most listened music artists and tag assignments. Because LASTFM does not have ground-truth communities, we did not use it in Section 5.2 for quantitative evaluation. Instead, we use it as a case study to show the ability of our new method for the multiplex semantic interpretation of communities. The statistical properties of these networks are shown in Table 2.

TABLE 2
THE STATISTICS OF REAL-WORLD NETWORKS

Datasets	n	e	m	c	Descriptions [34], [35], [36]
Texas	187	328	1,703	5	The WebKB dataset consists of four sub-
Cornell	195	304	1,703	5	networks from four US universities in
Washington	230	446	1,703	5	Texas, Cornell, Washington and Wiscon-
Wisconsin	187	328	1,703	5	sin, respectively.
Facebook	1045	26749	576	9	A subnetwork (id 107) of Facebook
Twitter	171	796	578	7	A subnetwork (id 629863) of Twitter
Citeseer	3,312	4,732	3,703	6	A Citeseer citation network
Cora	2,708	5,429	4,972	7	A Cora citation network
PubMed	19,729	44,338	500	3	Publications on PubMed
LASTFM	1,892	12717	11,946	-	The "friendship" network from *Last.fm*

The Real-world networks used in this paper. n, e, m and c are the numbers of nodes, edges, attributes, and communities of the network, and "-" means the absence of ground-truth communities.

Performance metrics. The methods compared may provide disjoint or overlapping community structures, so we choose different evaluation metrics in these two cases.

For disjoint community structures, since all the nine networks have ground-truth communities, we adopt accuracy (AC) [37], normalized mutual information (NMI) [37] and adjusted Rand index (ARI) [38] to compare the detected and ground-truth communities. To be specific, if the set of the detected communities is C and that of the ground-truth communities is C^*, the accuracy AC is defined as:

$$AC(C,C^*) = \frac{1}{n}\sum_{i=1}^{n}\delta(C_i^*, map(C_i)),$$

where $\delta(r,s)$ is the delta function which equals to 1 if r=s and 0 otherwise, and $map(C_i)$ is the mapping function that maps each community C_i to the index of the i^{th} community in C^*. The best mapping can be found by using the Kuhn-Munkres algorithm [39]. Besides, the normalized mutual information (NMI) is defined as:

$$NMI(C,C^*) = \frac{MU(C,C^*)}{\max(H(C),H(C^*))},$$

where $H(C) = \sum_{C_i} P(C_i)\log(P(C_i))$, is the entropy of the set of communities C, $P(C_i) = |C_i|/|C|$ and

$$MU(C,C^*) = \sum_{C_i,C_j^*} p(C_i,C_j^*)\log\frac{p(C_i,C_j^*)}{p(C_i)p(C_j)}$$

is the mutual information between C and C^*, where

$$p(C_i,C_j^*) = \left|C_i \cap C_j\right| \Big/ |C_i|.$$

In addition, the adjusted Rand index (ARI) is defined as:

$$ARI(C,C^*) = \frac{RI(C,C^*) - E(RI)}{\max(RI) - E(RI)},$$

where

$$RI(C,C^*) = \frac{a+b}{C_2^{n_{sample}}},$$

with $a = \sum_{i=1}^{n}\delta(C_i,C_i^*)$ and $b = \sum_{i=1}^{n}(1-\delta(C_i,C_i^*))$. If $C_i = C_i^*$, $\delta(C_i,C_i^*) = 1$, and 0 otherwise. $C_2^{n_{sample}}$ denotes all the possible combinations of the samples. $E(RI)$ denotes the expectation of Rand index. The range of ARI is [-1,1].

Some of the baseline methods in the evaluations provide overlapping communities which cannot be compared through AC, NMI and ARI in general. Thus, we use three other metrics, i.e. F-score [9], Jaccard similarity [9] and Omega Index [40], to evaluate the overlapping structures. F-score metric $F(C,C^*)$ between C and C^* is defined as:

$$F(C,C^*)=\frac{1}{2|C^*|}\sum_{C_i^*\in C^*}\max_{C_j\in C}F(C_i^*,C_j)+\frac{1}{2|C|}\sum_{C_j\in C}\max_{C_i^*\in C^*}F(C_i^*,C_j),$$

where $F(C_i^*,C_j)$ evaluates the F1 score between C_i^* and C_j. Jaccard metric $JAC(C,C^*)$ measures the Jaccard similarity between C and C^*, which is defined as:

$$JAC(C,C^*)=\sum_{C_i^*\in C^*}\frac{\max_{C_j\in C}JAC(C_i^*,C_j)}{2|C^*|}+\sum_{C_j\in C}\frac{\max_{C_i^*\in C^*}JAC(C_i^*,C_j)}{2|C|},$$

in which $JAC(C_i^*,C_j)$ evaluates the Jaccard similarity between C_i^* and C_i. Besides, Omega index is the overlapping version of the adjusted Rand index (ARI), which is defined as:

$$O(C,C^*)=\frac{O_u(C,C^*)-O_e(C,C^*)}{1-O_e(C,C^*)}.$$

$O_u(C,C^*)$ denotes the percentage of node pairs (with one node in community C and the other in C^*), defined as:

$$O_u(C,C^*)=\frac{1}{N}\sum_j|t_j(C)\cap t_j(C^*)|,$$

where $t_i(C)$ denotes a set of node pairs in the j^{th} community, and N the number of all node pairs in the network. $O_e(C,C^*)$ denotes the expectation of O_u, which can be defined as:

$$O_e(C,C^*)=\frac{1}{N^2}\sum_j|t_j(C)||t_j(C^*)|.$$

5.2 Quality Evaluation of Community Detection

We evaluate the performance of our method BTLSC for detecting communities on nine real-world networks with ground truth communities. The networks used are described in Table 2.

We use three types of baseline methods in the comparison. The first type includes the methods using network topology alone for community detection, i.e. DCSBM [7], IDCSBM [18] and BigCLAM [8]. The second uses node attributes only, which includes SMR [41]. The third uses network topology and node attributes together to find communities, including: SCI [27], MOEA-SA [25], ASCD [42], RSECD [43], CESNA [9] and DCM [44]. The source codes of all the methods compared are obtained from their authors, and we use their default parameters. All methods require the number of communities *c* to be pre-specified (except for MOEA-SA), so that we make it the same as that in ground truth. To our approach, we set the number of background topic, specialized topics and general topics respectively to 1 and respectively 1 and 1/2 times the number of communities, i.e. we set *D*=*c* and *E*=*c*/2 in Algorithm 1.

The experiment results for disjoint community detection are shown in Table 3. As shown, our algorithm BTLSC performs best on 5, 3 and 5 out of 9 networks in terms of AC, NMI and ARI, respectively. On the remaining networks where our BTLSC does not perform best, it is still competitive with that of the best baselines in most cases. To be specific, our method BTLSC is on average 0.1517, 0.1566, 0.1846, 0.0938, 0.0643, 0.0606 and 0.0413 more accurate than DCSBM, IDCSBM, SMR, SCI, ASCD-NMI, ASCD-ARC and RSECD in terms of AC; and 0.1606, 0.1796, 0.2433, 0.1870, 0.1652, 0.1208, 0.1159 and 0.0273 better than these methods in terms of ARI. Besides, in terms of NMI, our method BTLSC is competitive with RSECD (i.e. BTLSC is only 0.0159 less accurate than RSECD), and performs better than all the other methods. This validates the effectiveness of our new method in general.

TABLE 3
COMPARISON IN TERMS OF AC, NMI AND ARI

Metrics	Datasets	Methods								
		Topo	Topo	Cont	Both	Both	Both	Both	Both	Both
		DCSBM	IDCSBM	SMR	SCI	MOEA-SA	ASCD-NMI	ASCD-ARC	RSECD	BTLSC
AC [0,1]	Texas	0.4809	0.3128	0.4754	0.6230	N/A	0.6266	0.6066	0.6043	**0.6831**
	Cornell	0.3795	**0.5683**	0.3179	0.4564	N/A	0.4821	0.4921	0.5179	0.5179(2)
	Washington	0.3180	0.4608	0.4977	0.5115	N/A	0.5269	0.5269	0.5739	**0.6267**
	Wisconsin	0.3282	0.3702	0.4084	0.5038	N/A	0.5305	0.5267	**0.6792**	0.5458(2)
	Facebook	0.4519	0.3134	0.3615	0.5104	N/A	0.4782	0.4382	0.3912	**0.6548**
	Twitter	0.6049	0.3176	0.3827	0.5062	N/A	0.5527	0.5789	0.5375	**0.6288**
	Cora	0.3848	**0.5379**	0.3087	0.4062	N/A	0.5096	0.4963	0.3684	0.4878(4)
	Cite	0.2657	-	0.3028	0.2798	N/A	0.3263	0.3810	**0.4867**	0.3787(3)
	PubMed	0.5364	-	0.3995	0.4739	N/A	0.5038	0.5037	0.5845	**0.5921**
	AVG	0.4167	0.4118	0.3838	0.4746	N/A	0.5041	0.5078	0.5271	**0.5684**
NMI [0,1]	Texas	0.1665	0.0608	0.0355	0.1784	0.0942	0.2205	0.2088	0.3034	**0.3121**
	Cornell	0.0969	0.1334	0.0845	0.1144	0.1559	0.1618	0.1733	**0.3030**	0.1767(2)
	Washington	0.0987	0.0391	0.0730	0.1237	0.1574	0.1830	0.1768	**0.3389**	0.2734(2)
	Wisconsin	0.0314	0.1159	0.0721	0.1703	0.1252	0.2056	0.1953	**0.4489**	0.1375(5)
	Facebook	0.2940	0.2702	0.0940	0.2080	0.3291	**0.5838**	0.5785	0.3759	0.5642(3)
	Twitter	0.5748	0.1018	0.0326	0.4300	0.4504	0.6465	0.6557	0.6326	**0.6652**
	Cora	0.1707	**0.3789**	0.1328	0.1926	0.1120	0.3247	0.3305	0.1540	0.3204(4)
	Cite	0.0413	-	0.0118	0.0487	**0.3226**	0.0966	0.1361	0.2230	0.1570(3)
	PubMed	0.1228	-	0.0004	0.0559	0.1530	0.1485	0.1434	0.1760	**0.1769**
	AVG	0.1775	0.1572	0.0596	0.1691	0.2111	0.2857	0.2887	**0.3251**	0.3092(2)
ARI [-1,1]	Texas	0.1156	0.1940	-0.0693	0.1077	0.1671	0.1905	0.1897	0.3134	**0.4076**
	Cornell	0.1011	0.0137	-0.0084	0.0367	0.0898	0.0716	0.0725	**0.2551**	0.1969(2)
	Washington	0.0447	0.0361	0.1037	0.0662	0.0331	0.1015	0.1146	**0.3682**	0.3431(2)
	Wisconsin	0.1082	0.0581	0.0832	0.0435	0.1134	0.1449	0.1346	**0.4318**	0.1592(2)
	Facebook	0.1089	0.0979	0.0239	0.1457	0.1936	0.2082	0.1984	0.1342	**0.4896**
	Twitter	0.2780	0.0570	0.0583	0.1459	0.3141	0.2573	0.2655	0.2699	**0.3249**
	Cora	0.1168	0.2035	0.0237	0.1486	0.0021	0.2253	0.2345	0.1170	**0.2497**
	Cite	0.0185	-	0.0570	0.0216	0.0379	0.0849	0.1170	**0.2188**	0.1052(3)
	PubMed	0.1422	-	0.0035	0.0659	0.0269	0.0939	0.0923	0.1114	**0.1888**
	AVG	0.1133	0.0943	0.0306	0.0869	0.1087	0.1531	0.1580	0.2466	**0.2739**

Comparison of algorithms with disjoint community structures in terms of AC, NMI and ARI. "Topo", "Cont" and "Both" depict the types of algorithms which use network topology alone, node content alone, or both topology and content. "AVG" denotes the average performance of each algorithm in terms of each metric. Best results are in bold. The number after BTLSC indicates its rank among all the methods when it is not the best. "-" means runtime > 100 hours or out-of-memory. As the number of communities got by MOEA-SA may be not the same as that of ground-truth, we cannot calculate its AC values and mark it as "N/A". ASCD-ARC and ASCD-NMI are two versions of ASCD.

TABLE 4
COMPAEISON IN F-SCORE, JACCARD, OMEGA INDEX

Metrics	Datasets	Methods			
		Topo	Both	Both	Both
		BigCLAM	CESNA	DCM	BTLSC
F-score [0,1]	Texas	0.2064	0.2354	0.1115	**0.4192**
	Cornell	0.1323	0.2348	0.1438	**0.4433**
	Washington	0.1335	0.2191	0.1245	**0.4757**
	Wisconsin	0.1284	0.2317	0.1045	**0.3649**
	Twitter	0.3979	0.4382	0.1057	**0.5691**
	Facebook	0.4006	**0.4905**	0.3921	0.4354(2)
	Cora	0.1889	0.3105	0.0343	**0.4661**
	Cite	0.0930	0.3380	0.0250	**0.3412**
	PubMed	0.0772	0.2797	0.0038	**0.5691**
	AVG	0.1954	0.3087	0.1161	**0.4538**
Jaccard [0,1]	Texas	0.1218	0.1357	0.0603	**0.3022**
	Cornell	0.0718	0.1347	0.0795	**0.2989**
	Washington	0.0725	0.1240	0.0672	**0.3415**
	Wisconsin	0.0701	0.1314	0.0554	**0.2430**
	Twitter	0.2613	0.2963	0.0575	**0.4514**
	Facebook	0.2894	**0.3818**	0.2846	0.3289(2)
	Cora	0.1089	0.1910	0.0176	**0.3259**
	Cite	0.0501	0.0173	0.0127	**0.2204**
	PubMed	0.0404	0.1626	0.0019	**0.4077**
	AVG	0.1207	0.1750	0.0946	**0.3244**
Omega Index [-1,1]	Texas	-0.0177	-0.0013	-0.0017	**0.2272**
	Cornell	-0.0027	0.0071	-0.0040	**0.0760**
	Washington	0.0208	0.0166	-0.0009	**0.1665**
	Wisconsin	-0.0148	-0.0070	0.0006	**0.0529**
	Twitter	**0.1037**	0.0997	0.0084	0.0108(3)
	Facebook	0.1739	**0.1754**	0.0086	-0.0158(4)
	Cora	0.1430	0.0532	0.0006	**0.2493**
	Cite	0.0000	0.0000	0.0002	**0.0089**
	PubMed	0.0235	0.0045	0.0000	**0.1888**
	AVG	0.0477	0.0387	0.0013	**0.1072**

Comparison of algorithms with overlapping community structures in terms of F-score, Jaccard and Omega Index.

As a supplement, the comparison with baseline methods for finding overlapping communities is shown in Table 4. In this case, our method outperforms 8, 8 and 7 of 9 networks in terms of F-score, Jaccard and Omega index, respectively. On the remaining networks (e.g. Facebook) where our method does not perform best, it is still competitive with the best baseline method (i.e. CESNA). On average, the performance of our BTLSC is always the best in terms of each of the three metrics. In more details, BTLSC is on average 0.2584, 0.1451 and 0.3377 more accurate than BigCLAM, CESNA and DCM in F-score; 0.2307,0.1494 and 0.2298 more accurate than these methods in Jaccard; and 0.0595, 0.0685 and 0.1059 better in Omega index. These results further validate the effectiveness of the proposed new approach on community detection.

In summary, our method outperforms almost all of the methods compared in terms of the six metrics (see both Tables 3 and 4). The season may be mainly that:
1) Our algorithm outperforms both the topology-based methods (e.g. DCSBM, IDCSBM, BigCLAM) and content-based methods (e.g. SMR). In particular, our algorithm is better than DCSBM, even though DCSBM and our method share similar mechanism to deal with network topologies. This demonstrates that making use of content information by appropriate ways indeed helps improve the quality of the discovered network communities.
2) Compared to the algorithms that use both network topology and contents, our algorithm BTLSC still has obvious advantages. This is also not surprising. For example, in SCI and CESNA, the network communities and topical clusters (which describe only a single level of topics of words) share the same set of latent qualities. This is too strong an assumption to describe semantics as well as their relationship with communities. In contrast, our BTLSC models the multiplex semantics of words, and also describes the intrinsic relationship between communities and these semantics, which is a more natural way. RSECD and ASCD both are methods based on NMF, and also do not consider the hierarchical use of word semantics; while our BTLSC is based on probabilistic inference, and describes well the multiplex semantics of words. MOEA-SA is a multi-objective algorithm and DCM is based on heuristic optimization, and they also do not consider the hierarchical semantic structure of networked contents. To sum up, these methods all ignore the existing hierarchical structure of semantics which leads to inaccurate fitting to the textual contents. We adopt a reasonable generative mechanism, which robustly solves the interaction between communities, background topic and two-level topics. Through the more natural fitting to the semantic information, the quality of community detection is finally improved. In addition, it is often observed that users tend to communicate frequently over certain topical interests (i.e. the specialized topics) and then form a community; the common interest tendency (i.e. the general topics) among communities is not unrelated; and some semantic information (i.e. the background topic) can reflect the background of all communities in the network. Our method shows well this phenomenon that typically exists in real networks and thus leads to better performance.

5.3 Efficiency Comparison

We also report the runtimes of all the methods compared on all the datasets used, as shown in Table 5. The methods using network topology alone (e.g. BigCLAM) and that using node attributes alone (e.g. SMR) typically run faster than the methods using network topology and node attributes together. Among the methods using both these two sources of information, the efficiency of our method BTLSC is only lower to DCM which, however, has a much lower accuracy than BTLSC. BTLSC runs faster than CESNA and SCI on large networks while slower on small networks. (To be specific, compared with SCI, BTLSC runs 61.11%, 3.85% and 34.78% slower only on small networks Texas, Washington and Wisconsin; but runs 8.42%, 60.83%, 82.76% and 84.04% faster on the largest 4 networks Facebook, Cora, Citeseer and Pubmed, as well as 8.70% and 5.15% faster on small networks Cornell and Twitter. Similarly, BTLSC runs 52.63%, 5.00%, 17.39%, 6.90% and 31.43% slower than CESNA on small networks Texas, Cornell, Washington, Wisconsin and Twitter; but runs 79.29%, 82.59%, 85.29%, 98.92% faster on the largest 4 networks Facebook, Cora, Citeseer and Pubmed.) BTLSC runs faster than the remaining four algorithms on all the networks. This result, to some extent, validate the scalability of our new algorithm BTLSC on some large-scale networks.

TABLE 5
RUNTIMES OF DIFFERENT ALGORITHMS

Datasets/ Runtime(s)	Methods										
	DCSBM	IDCSBM	BigCLAM	SMR	SCI	MOEA-SA	CESNA	DCM	ASCD-NMI	RSECD	BTLSC
Texas	6.4	4.6e1	2.3e-1	5.6e-1	1.8	3.2e3	1.9	3e-1	4.8	7.4	2.9
Cornell	9.6	4.8e1	1.6e-1	7.7e-1	2.3	2.8e3	2.0	1.3	1.9	7.1	2.1
Washington	4.1	5.4e1	1.7e-1	3.1e-1	2.6	4.9e3	2.3	1.1	9.2	8.8	2.7
Wisconsin	6.1	6.9e1	1.9e-1	4.7e-1	2.3	6.4e3	2.9	1.5	8.6	9.8	3.1
Twitter	5.1	4.5e1	1.0e-2	9.1e-1	9.7e-1	1.7e3	7e-1	2e-1	8.6e-1	5.1	9.2e-1
Facebook	3.9e2	9.1e2	4.2	2.2	9.5	1.8e5	4.2e1	2.6	3.2e1	7.7e1	8.7
Cora	1.3e3	4.5e3	4.5e-1	1.4e1	1.2e2	1.1e6	2.7e2	8.2	1.3e2	1.1e3	4.7e1
Cite	1.0e3	-	3.0e-2	3.0e1	2.9e2	1.1e6	3.4e2	2.2e1	1.4e2	4.0e3	5.0e1
PubMed	1.2e4	-	4.5	1.9e3	5.7e3	2.7e6	8.4e4	9.7e2	1.1e3	3.7e5	9.1e2

Running times of different algorithms in terms of seconds. ASCD-ARC and ASCD-NMI have similar running times, so we only show the results of ASCD-NMI. All methods run on a Dell workstation (Intel® Xeon® CPU E5-2680 V3@2.5GHz processor with 128 Gbytes of main memory).

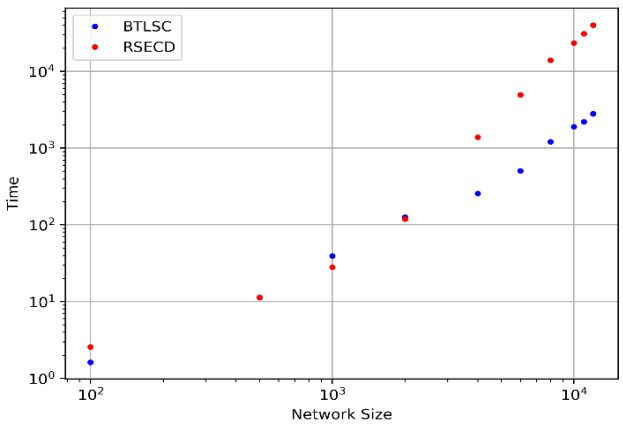


Fig. 3. Scalability test. Runtimes of BTLSC and RSECD on 10 groups of artificial networks. Each point represents the average runtime of 20 randomly sampled networks on the same size. Both x- and y-axes are log-scaled so that the trends can be easily distinguished.

We further use some artificial networks to compare the efficiency between our BTLSC and the best baseline in accuracy, i.e. RSECD, which is also most similar to BTLSC. We use the method introduced in [10] to generate networks with node contents. To be specific, we first generate the network topology based on Girven-Newman model [11]. For each network with n nodes, the nodes were divided into 4 communities. Each node has on average z_{in} edges connecting to the nodes of the same community and z_{out} edges connecting to the nodes of other communities, and $z_{in} + z_{out} = 16$. Thereafter, we generate a $4h$-dimensional binary attributes for each node v_i to form 4 attribute clusters, correctly corresponding to the 4 communities generated above. In more details, for each node in the s^{th} cluster, we use a binomial distribution with mean $\rho_{in} = h_{in}/h$ to generate a h-dimensional binary vector as its $((s-1)\times h+1)^{th}$ to $(s\times h)^{th}$ attributes and generated the rest attributes by a binomial distribution with mean $\rho_{out} = h_{out}/3h$, (with $4h$ = 200 and $h_{in} + h_{out} = 16$).

To make the above benchmark serve well in this efficiency comparison, we vary the network size n from 100 to 12,000 (with n = 100, 500, 1,000, 2,000, 4,000, 6,000, 8,000, 10,000, 11,000 and 12,000), and keep $z_{in} = 14$ and $h_{in} = 14$ fixed (making the network have a relative clear community and cluster structure). The functions of BTLSC and RSECD with the increase of net-work size are shown in Fig. 3. As shown, BTLSC is indeed more scalable than RSECD in general, especially when the network size increases. After investigation, we find the possible reason. That is, the variational inference mechanism used in BTLSC makes it typically converge faster than RSECD (which is based on NMF and adopts multiplicative update rules), especially on large-scale networks.

5.4 The Case Study Analysis on LASTFM

The dataset we used in the case study analysis is LASTFM which has been introduced in section 5.1. Except for the one background topic which represents the background information of the whole network, we set the number of communities and the number of specialized topics identically to 38 ($c = D = 38$), as suggested by [27]. As for the number of general topics E, we performed experiments by changing this number. We observed that, when E is biger than 4, there will apprear some highly overlapped general topics. So we set this number to 4 ($E = 4$) in this test.

Under the above setting, we obtained 4 groups of topic-related communities under 4 general topics, in which each community corresponds to a clear specialized topic. The background topic words of the whole network are shown in Fig. 4. Also of note, here we only show some of the communities in each group to display the two-level semantics in these groups with Figs. 5, 6, 7 and 8, due to space limit.

First, we show the words that are related to the background topic of the whole context in Fig. 4. Since the network comes from an online music system, *last.fm*, it is not strange that all the words in the background topic are related to music and do not reflect any specific information of music. For instance, "pop" and "dance" are two top words in the word cloud and are typical background words for music. The reason for the appearance of "British" is that *last.fm* is born in UK. "Instrumental" is also a word which can reflect any type of musics, since any type of music needs instruments. So, by modelling them separately as the background topic, these words will not mix with the general and specialized topic words, making the semantic representation more distinctive.

Fig. 4. Word cloud of the background topic of the whole LASTFM network. The word size is proportional to the probability that this word belongs to the background topic.

Then, we introduce the four music groups which respectively share the four general topics. The first we found in the LASTFM network is a group of topic-related communities of electronic music lovers, as shown in Fig. 5. For example, the words such as "electronic" and "electropop" in the general topic #1 are suitable for the description of almost all types of electronic music. On the other hand, these communities sharing the same general topic are also formed by fans from different branches of electronic music. To be specific, community #16 is composed of "high techno"

music lovers. The word with the highest probability in this community is "techno", which is a classic electronic music that can be compared with another electronic music "house". Community #33 is a group of fans who loves the "dubstep" music, and the origin of dubstep is affiliated with "post-punk". Dubstep was located in London, and thus it is reasonable that our algorithm finds the word "London" in the specialized topic of community #33. "New wave" is also a branch of electronic music, as shown in community #29. It has retained many characteristics of punk music. So, the word "punk" also shows up in the specialized topic of community #29. Community #27 gathers the "lounge" music fans. This is a form of music which is also called "chill-out". In addition, the lounge is considered to belong to "oldies" due to its relaxed and gentle tune. "Trance" fans gather in community #19. "Trance" music originated from hardcore music. "Vocal trance" is also an important style of trance music.

Fig. 5. The first group of topic-related communities which shares general topic #1. The top center word cloud shows the keywords of general topic #1. The five word clouds around the general topic show the specialized topic words of communities #16, #33, #19, #27 and #29, respectively. The word size is proportionate to the probability of this word belonging to a general or specialized topic.

The second group of communities is related to rock music, which is shown in Fig. 6. Words in general topic #2 reflect the detailed information of rock music. To be specific, community #1 gathers "heavy- metal" music lovers. There seems to be another form of rock in general topic #1, which is "nu-metal", and also known as "grunge". "Punk" fans were gathered in community #30 by our algorithm since we found "punk" and "punk-rock" in the specialized topic of community #30 with high probabilities. The fans of "Progressive-rock" are centered in community #6. Community #12 is formed by the fans of "alternative-rock" which is often used to be compared with "indie-rock".

The third group of communities corresponds to general topic #3, which is shown in Fig. 7. Communities #36 and #38 both belong to jazz music. In this general topic, our algorithm found the words "Jazz", "blues" and "Ragtime", which all tell that jazz comes from Blues and Ragtime. "Smooth-jazz" and "acid-jazz", as shown in the specialized topic word cloud of community #36, are both fusion jazz. Community #38 gathers lovers of "funk", which is a kind of black jazz and typically mixed with rap.

The last group corresponds to the general topic #4, which is shown in Fig. 8. The words of general topic #4 say that this may be a group of pop music enthusiasts. The specialized topic of community #28 is related to Japanese pop, which is also called "J-pop". Community #8 is dominated by the fans of "R&B" and "hip-pop". "Soundtrack" and "Folk" lovers also formed their communities separately, i.e. communities #14 and #26, respectively.

This case study shows that the new algorithm indeed has the ability of finding the background topic of the whole network, describing each community by specialized topics, and providing the common semantic content of communities with similar interests.

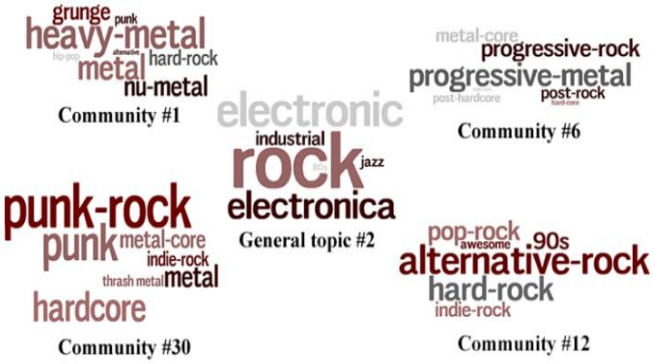


Fig. 6. The second group of communities which shares general topic #2. The central word cloud shows the keywords of general topic #2 and the surrounding four word clouds show the specialized topics of communities #1, #30, #6 and #12, respectively.

Fig. 7. The third group of communities shares general topic #3. The central word cloud shows keywords of the general topic #3. The two word clouds on both side are the specialized topic words corresponding to communities #36 and #38, respectively.

Fig. 8. The fourth group of communities shares general topic #4. The central word cloud shows key words of general topic #4. The surrounding four word clouds denote the specialized topic words of communities #28, #8, #14 and #26, respectively.

6 Conclusions and Discussion

In this paper, a probabilistic generative model, namely BTLSC, has been proposed to find and profile communities. The new model can not only detect communities more accurately, but also offers a rich explanation of communities through the structured utilization of semantic contents. To be specific, BTLSC recognizes whether the words can belong to a background topic or a two-level topic. It uses the background topic to reflect the commonality of the whole

network, and the general and specialized topics to explain the communities in clearer and different semantic granularities. The model is trained under a variational inference framework. We perform a series of experiments to test BTLSC by comparing with ten state-of-the-art methods. These results show the surpreiority of BTLSC in finding communities, both in term of accuracy and computing speed. We also provide a case study to show the power of BTLSC in explaining communities.

In this work, we mainly focus on the two-level semantics (i.e. the general and specialized topics) of communities except for the background topic for the whole network. Of course, there may be higher semantic levels in more complicated cases, while we simplify them all to a specialized topic-level here since two-level is typically a most important extension from the one- to multi-level cases and is often satisfactory to find and profile network communities. In addition, taking too many topic levels into account will often case poor matching between network topology and semantic contents in real life. This mismatching often occurs in community detection when integrating network topology and semantic contents [10], [24]. Two-level topics are most often sufficient to express the rich semantics of each community [45], and provide a good matching between topology and contents at the same time. Also, the overfitting issue will become more serious if the semantic-level is too deep, making the model too complicated to fit. On the other hand, the higher level of semantics is of course also significant since it essentially can be taken as a refinement of the specialized topics. An ideal way may be that one determines the best number of levels of semantics from the networked data. For example, one can utilize the idea of Bayesian model selection, i.e. to add some appropriate shrinkage prior on the multi-level of topics, so that the unrelated topic levels can be automatically filtered out in training of the model. However, model selection itself is a bigger problem than network community detection, and is also not the main focus of this work. We will leave it as our main future work.

REFERENCES

[1] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang, "Spectral Redemption in Clustering Sparse Networks," *Proc. Natl. Acad. Sci. USA*, vol. 110, no. 52, pp. 20935-20940, 2013.

[2] Y. Li, K. He, K. Kloster, D. Bindel and J. Hopcrpft, "Local Spectral Clustering for Overlapping Community Detection", *ACM Transactions on Knowledge Discovery from Data*, vol.12, no.17, 2018.

[3] J.C. Delvenne, S.N. Yaliraki, and M. Barahona, "Stability of Graph Communities Across Time Scales," *Proc. Natl. Acad. Sci. USA*, vol. 107, no. 29, pp. 12755-12760, 2010.

[4] M. Rosvall, A.V. Esquivel, A. Lancichinetti, J.D. West, and R. Lambiotte, "Memory in Network Flows and its Effects on Spreading Dynamics and Community Detection," *Nature Communications*, vol. 5, no. 1, pp. 4630, Aug. 2014.

[5] V.D. Blondel, J.L. Guillaume, R. Lambiotte and E. Lefebvre, "Fast Unfolding of Communities in Large Networks," *J. Stat. Mech.*, vol. 2008, no. 10, pp. P10008, 2008.

[6] L. Ma, M. Gong, J. Liu, Q. Cai and L. Jiao, "Multi-level Learning Based Memetic Algorithm for Community Detection," *Appl. Soft. Comput.*, vol.19, pp.121-133, 2014.

[7] B. Karrer and M. Newman, "Stochastic Block Models and Community Structure in Networks," *Phys. Rev. E*, vol. 83, no. 2, pp. 211-222, 2011.

[8] J. Yang and J. Leskovec. "Overlapping Community Detection at scale: A Nonnegative Matrix Factorization Approach," *Proc. Sixth ACM international conf. Web search and data mining*, pp. 587-596, 2013.

[9] J. Yang, J. McAuley, and J. Leskovec, "Community Detection in Networks with Node Attributes," *Proc. 13th IEEE Int. Conf. Data Mining*, pp.1151-1156, 2013.

[10] D. He, Z. Feng, D. Jin, X. Wang and X. Zhang, "Joint identification of network communities and semantics via integrative modeling of network topologies and node contents," *Proc. Thirty-First AAAI Conference on Artificial Intelligence*, no.9, pp.116-124, 2017.

[11] M. Girvan and M.E.J Newman, "Community Structure in Social and Biological Networks," *Proc. Natl. Acad. Sci.* USA, vol. 99, no. 12, pp. 7821-7826, 2002.

[12] S. Jia, L. Gao, Y. Gao, J. Nastos, Y. Wang, X. Zhang and H. Wang, "Defining and Identifying Cograph Communities in Complex Networks," *New Journal of Physics*, vol. 17, no. 1, pp. 013044, 2015.

[13] M. Rezvani, W. Liang, C. Liu and J. X. Yu, "Efficient Detection of Overlapping Communities Using Asymmetric Triangle Cuts," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 11, pp. 2093-2105, 2018.

[14] S. Qiao, N. Han, Y. Guo, R. Li, J. Huang, J. Guo, L. A. Gutierrez and X. Wu, "A Fast Parallel Community Discovery Model on Complex Networks Through Approximate Optimization," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1638-1651, 2018.

[15] F. Huang, X. Li, S. Zhang, J. Zhang, J. Chen and Z. Zhai, "Overlapping Community Detection for Multimedia Social Networks," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1881-1893, 2017.

[16] X. Wen, W. Chen, Y. Lin, T. Gu, H. Zhang, Y. Li, Y. Yin and J. Zhang, "A Maximal Clique Based Multiobjective Evolutionary Algorithm for Overlapping Community Detection," *IEEE Trans. Evol. Comp.*, vol. 21, no. 3, pp. 363-377, 2017.

[17] K. Nowicki and T.A.B. Snijders, "Estimation and prediction for stochastic blockstructures," *Journal of the American Statistical Association* vol.96, no.455, pp.1077-1087, 2001

[18] T. Herlai, M. N. Schmidt and M. Mørup, "Infinite-degree-corrected stochastic block model," *Phys. Rev.* E, vol.90, no.3, pp.032819, 2014.

[19] V. Lyzinski, M. Tang, A. Athreya, Y. Park and C. E. Priebe, "Community Detection and Classification in Hierarchical Stochastic Blockmodels," *IEEE Transactions on Network Science and Engineering*, vol. 4, no. 1, pp. 13-26, 2017.

[20] B. David, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol.3, no. Jan, pp. 993-1022, 2003.

[21] Y. Sun, C.C. Aggarwal and J. Han, "Relation Strength-aware Clustering of Heterogeneous Information Networks with Incomplete Attributes," *Proc. Thirty-seventh VLDB Conf. Database*, pp. 394-405, 2012.

[22] Z. Wu, Z. Lu, and S.Y. Ho. "Community Detection with Topological Structure and Attributes in Information Networks," *ACM Trans. Intell. Syst. Tech.* vol.8, no. 33, pp. 1-17. 2016.

[23] T. Yang, R. Jin, Y. Chi and S. Zhu, "Combining Link and Content for Community Detection: A Discriminative Approach," *Proc. Twenty-fifth SIGKDD Conf. Knowledge Discovery and Data Mining*, pp. 927-936, 2009.

[24] B. Zhan, H. Li, J. Cao, Z. Wang and G. Gao, "Dynamic Cluster

Formation Game for Attributed Graph Clustering", *IEEE Transactions on Cybernetics*, vol.49, no.1 pp.1-14, 2019.

[25] Z. Li, J. Liu and K. Wu, "A Multiobjective Evolutionary Algorithm Based on Structural and Attribute Similarities for Community Detection in Attributed Networks," *IEEE Transactions on Cybernetics*, vol.48, no.7, pp.1963-1976, 2018.

[26] Y. Pei, N. Chakraborty and K. Sycara, "Nonnegative Matrix Trifactorization with Graph Regularization for Community Detection in social networks," *Proc. twenty-fourth IEEE international conf. Data Mining*, pp. 2083-2089, 2015.

[27] X. Wang, D. Jin, X. Cao, L. Yang, and W. Zhang, "Semantic Community Identification in Large Attribute Networks," *Proc. Thirtieth AAAI Conf. Artificial Intelligence*, pp. 256-271, 2016.

[28] L. Liu, L. Xu, Z. Wang, and E. Chen, "Community Detection Based on Structure and Content: A Content Propagation Perspective," *Proc. IEEE International Conf. Data Mining*, pp. 271-280, 2015.

[29] Z. Xu, Y. Ke, Y. Wang, H. Cheng and J. Cheng, "A Model-based Approach to Attributed Graph Clustering," *Proc. thirty-third SIGMOD conf. Management of Data*, pp. 505-516, 2012.

[30] L. Hu, K. C. C. Chan, X. Yuan and S. Xiong, "A variational Bayesian framework for cluster analysis in a complex network," *IEEE Trans. Knowl. Data Eng.*, 2019

[31] G. Zhang, D. Jin, P. Jiao, F. Fogelman-Soulie and X. Huang, "Finding Communities with Hierarchical Semantics by Distinguishing General and Specialized Topics," *Proc. Twenty-seventh IJCAI conf. Artificial Intelligence*, pp. 3648-3654, 2018.

[32] N.M. Nasrabadi, "Pattern Recognition and Machine Learning," *J. of electronic imaging*, vol.16, no.4, pp. 049901, 2007.

[33] M.D. Hoffman, D.M. Blei, C. Wang and J. Paisley, "Stochastic Variational Inference," *J. Mach Learn Res*, pp. 1303-1347, 2013.

[34] I. Cantador. *The 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, 2016. http://ir.ii.uam.es/hetrec2011.

[35] J. Leskovec, Stanford Network Analysis Project. http://snap.standford.edu.

[36] P. Sen, G. Namata, M. Bilgic and L. Getoor," Collective classification in the network data," *AI Magazine*, vol. 29, no. 3, pp. 93, 2008

[37] H. Liu, Z. Wu, X. Li, D. Cai and T. S. Huang, "Constrained Nonnegative Matrix Factorization for Image Representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 34, no. 7, pp. 1299-1311, 2012.

[38] D. Stenley. "Properties of the Hubert-Arable Adjusted Rand Index", *Psychological methods*, vol.9, no.3, pp.386, 2004.

[39] L. Lovasz and M. Plummer, *Matching Theory*. North Holland 1986.

[40] Collins, Linda M., and Clyde W. Dent. "Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions." *Multivariate Behavioral Research*, vol. 23, no. 2, pp. 231-242, 1988.

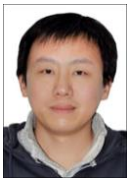
[41] H. Hu, Z. Lin, J. Feng and J. Zhou, "Smooth Representation Clustering," *Proc. twenty-seventh IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3834-3841, 2014.

[42] M. Qin, D. Jin, K. Lei, B. Gabrys and K. Musial, "Adaptive Community Detection Incorporating Topology and Content in Social Networks," *Knowledge-Based Systems*, vol.161, pp. 342-356, 2018.

[43] D. Jin, Z. Liu, D. He, B. Gabrys and K. Musial, "Robust Detection of Communities with Multi-semantics in Large Attributed Network," *Proc. International conf. on Knowledge Science, Engineering and Management*, pp.362-376, 2018.

[44] S. Pool, F. Bonchi and M. Leeuwen, "Description-driven Community Detection," *ACM Trans. Intell. Syst. Technol.*, vol.5, no.2, pp. 28, 2014.

[45] P. Xie and E.P. Xing. "Integrating document clustering and topic modeling," *Proc. Twenty-Ninth Conf. Uncertainty in Artificial Intelligence (UAI'13)*, pp.694-703, 2013



Di Jin received his Ph.D. degree in computer science from Jilin University, Changchun, China, in 2012, He was a Post-Doctoral Research Fellow at the School of Design, Engineering, and Computing, Bournemouth University, Poole, U.K., from 2013 to 2014. He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. He has published more than 50 papers in international journals and conferences in the areas of community detection, social network analysis, and machine learning. He serves as invited reviewers for journals including TKDE, and senior program committees for conferences incuding AAAI.

Kunzeng Wang is currently a master student of College of Intelligence and Computing from Tianjin University, Tianjin, China. His research interests mainly related to community detection and social network analysis.

Ge Zhang received the M.S. degree in College of Intelligence and Computing, Tianjin University, China. Her research interests are mainly related to community detection, social network analysis and machine learning.

Pengfei Jiao received the Ph.D. degree in computer science from Tianjin University, Tianjin, China, in 2018. He is currently an Assistant Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. He has published over 20 international journal articles and conference papers. His current research interests include data mining, complex network analysis and machine learning.

Dongxiao He received her Ph.D. degrees in computer science from Jilin University, Changchun, China, in 2014. She was a Post-Doctoral Research Fellow in Department of Computer Science, Dresden University of Technology, Germany, from 2014 to 2015. She is an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. She has published over 40 international journal and conference papers. Her current research interests include data mining and analysis of complex networks.

Françoise Fogelman-Soulié is a professor at the College of Intelligence and Computing, Tianjin University, China. She received her PhD from University of Grenoble, France in 1985. She was Professor at the University of Paris 11-Orsay, and then moved to industry, where she held various positions until KXEN, where she was Vice President Innovation until the company was bought out by SAP. She has co-authored more than 150 scientific publications and 13 books, on social networks, data mining and neural networks/artificial intelligence.

Xin Huang is currently an assistant professor in the department of computer science at the Hong Kong Baptist University. He received his BEng degree in computer science from the Xiamen University in 2010, and PhD degree in systems engineering and engineering management from the Chinese University of Hong Kong in 2014. His research interests mainly focus on graph data management and mining. He serves as invited reviewers for journals including VLDBJ and TKDE, and program committees for conferences including VLDB, KDD, ICDE, WWW, EDBT, AAAI, and SDM.