Graph Attention Topic Modeling Network

Liang Yang[*]
School of Artificial Intelligence,
Hebei University of Technology
Tianjin, China
SKLOIS, IIE, CAS, Beijing, China
yangliang@vip.qq.com

Fan Wu
School of Artificial Intelligence,
Hebei University of Technology
Tianjin, China
wufanslient@outlook.com

Junhua Gu
School of Artificial Intelligence,
Hebei University of Technology
Tianjin, China
jhgu@hebut.edu.cn

Chuan Wang, Xiaochun Cao
SKLOIS, Institute of Information
Engineering, Chinese Academy of
Sciences, Beijing, China
{wangchuan,caoxiaochun}@iie.ac.cn

Di Jin[†]
College of Intelligence and
Computing, Tianjin University
Tianjin, China
jindi@tju.edu.cn

Yuanfang Guo[*]
School of Computer Science and
Engineering, Beihang University
Beijing, China
andyguo@buaa.edu.cn

ABSTRACT

Existing topic modeling approaches possess several issues, including the overfitting issue of Probablistic Latent Semantic Indexing (pLSI), the failure of capturing the rich topical correlations among topics in Latent Dirichlet Allocation (LDA), and high inference complexity. In this paper, we provide a new method to overcome the overfitting issue of pLSI by using the amortized inference with word embedding as input, instead of the Dirichlet prior in LDA. For generative topic model, the large number of free latent variables is the root of overfitting. To reduce the number of parameters, the amortized inference replaces the inference of latent variable with a function which possesses the shared (amortized) learnable parameters. The number of the shared parameters is fixed and independent of the scale of the corpus. To overcome the limited application of amortized inference to independent and identically distributed (i.i.d) data, a novel graph neural network, Graph Attention TOpic Network (GATON), is proposed to model the topic structure of non-i.i.d documents according to the following two observations. First, pLSI can be interpreted as stochastic block model (SBM) on a specific bi-partite graph. Second, graph attention network (GAT) can be explained as the semi-amortized inference of SBM, which relaxes the i.i.d data assumption of vanilla amortized inference. GATON provides a novel scheme, i.e. graph convolution operation based scheme, to integrate word similarity and word co-occurrence structure. Specifically, the bag-of-words document representation is modeled as a bi-partite graph topology. Meanwhile, word embedding, which captures the word similarity, is modeled as attribute of the word node and the term frequency vector is adopted as the attribute of the document node. Based on the weighted (attention) graph convolution operation, the word co-occurrence structure and word similarity patterns are seamlessly integrated for topic identification. Extensive experiments demonstrate that the effectiveness of GATON on topic identification not only benefits the document classification, but also significantly refines the input word embedding.

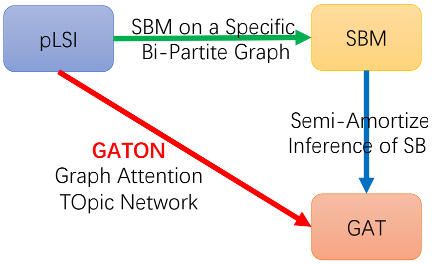


Figure 1: The intuition of this paper. First, Graph Attention Network (GAT) is interpreted as the semi-amortized inference of Stochastic Block Model (SBM) in Section 4.4. Second, probabilistic latent semantic indexing (pLSI) is interpreted as SBM on a specific bi-partite graph in Section 5.1. Finally, a novel graph neural network, Graph Attention TOpic Network (GATON), is proposed for topic modeling based on the above two interpretations in Section 5.2.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**;

KEYWORDS

Graph Neural Network, Stochastic Block Model, Graph Attention Network, Topic Modeling, Bipartite Network

ACM Reference Format:
Liang Yang, Fan Wu, Junhua Gu, Chuan Wang, Xiaochun Cao, Di Jin, and Yuanfang Guo. 2020. Graph Attention Topic Modeling Network. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380102

[*] Both authors contributed equally to this research.
[†] Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WWW '20, April 20–24, 2020, Taipei, Taiwan
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380102

1 INTRODUCTION

Topic modeling aims to discover the latent semantics of the documents. Most state-of-the-art topic modeling approaches, including Probabilistic Latent Semantic Indexing (pLSI), simply represent document as bag-of-words and construct generative models for text corpora. By fitting the model to the observed document collection, latent topics can be revealed via certain inference algorithms. Unfortunately, the large number of latent variables in these generative models, which grows exponentially with the number of documents and topics, makes the inferences inefficient and induces overfittings to the training documents.

Although Latent Dirichlet Allocation (LDA) [5] alleviates the overfitting issue by introducing Dirichlet priors for latent variables, it fails to capture the rich topical correlations among topics, because the introduced Dirichlet priors assume that topics occur independently. Many subsequent literatures attempt to relax the independence assumption of the Dirichlet prior. Correlated Topic Model (CTM) [4] explicitly models the correlation patterns by adopting a logistic-normal prior with covariance matrix. Despite of its impressive representation power, its high inference complexity, which is induced by the non-conjugacy of logistic-normal prior, prevents it from applying in practice. Correlated Topic Modeling with Topic Embedding (CTMTE) [16] extracts the correlation structures of the latent topics by introducing Gaussian distribution based topic embeddings. However, its complicated hybrid inference algorithm, which leverages the reparameterization trick in variational inference, makes it difficult to implement.

In parallel with the generative topic modeling came the research of word embedding. Word embedding aims at learning distributed word representation, where words with similar meanings tend to be close in a lower-dimensional embedding space, instead of the traditional one-hot representation. Most of the word embedding algorithms, including Skip-gram and CBOW [37], are neural language models, which measure the words similarity based on their co-occurrences within a local context window. However, most of the generative topic models often ignore this word similarity, which is a supplement to the bag-of-words document representation. Therefore, to incorporate word embedding into topic modeling, existing approaches usually adopt topic embedding into neural language model and model the relationships between words and topics by jointly modeling their embeddings [13, 16, 29, 30]. Unfortunately, these approach are often incapable to model high-order correlation between documents.

In this paper, we attempt to overcome the overfitting issue of pLSI by exploiting amortized inference with the word embedding as input, instead of the Dirichlet prior in LDA. The intrinsic reason of overfitting is the large number of parameters to learn. For generative topic model, the large number of free latent variables, which is much more than can be justified by the data, is the root of overfitting. To regularize the inference process and reduce the number of parameters, amortized inference is adopted. The amortized inference replaces the inference of the latent variables with a function which shares (amortizes) the learnable parameters. The number of the learnable parameters is then fixed and independent of the scale of the corpus. Typically, amortized inference can be used to process the independent and identically distributed (i.i.d.) data. For example, VAE adopts amortized inference to model image data. Unfortunately, the documents in topic modeling is not i.i.d., because the words in documents are often semantically correlated.

To integrate word embedding into generative topic modeling with Semi-Amortized inference, we have obtained the following two observations. First, pLSI can be interpreted as stochastic block model (SBM) on a bi-partite graph by comparing the generative processes of pLSI and SBM as shown in Figure 1. Second, graph attention network (GAT) is equivalent to the semi-amortized inference of SBM as shown in Figure 1, via certain mathematical deductions. The vanilla amortized inference [22] converts the inference problem of a large number of latent variables to a learning problem with shared (amortize) parameters, and thus significantly reduces the number of parameters. The semi-amortized inference, which combines the amortized inference [22] and traditional EM inference [11], relaxes the i.i.d data assumption of amortized inference. According to these two observations, a novel bi-partite graph attention network, named Graph Attention TOpic Network (GATON), is proposed for topic modeling. Our GATON integrates word similarity and word co-occurrence structure via a novel approach, i.e., graph convolution operation. Specifically, the bag-of-words document representation is modeled as a bi-partite graph topology. Meanwhile, the word embedding representation, which captures the word similarity, is modeled as the attribute of the word node and the term frequency is considered as the attribute of document node. Via the weighted (attention) graph convolution operation, the bag-of-words structure and word similarity patterns are seamlessly integrated for topic identification.

The main contributions are summarized as follows.

- We propose a novel approach to overcome the overfitting issue in topic modeling. Instead of directly imposing the Dirichlet prior, which prevents from mining topic correlations, we adopt amortized inference, with the word embedding as input, to significantly reduce the number of to-be-estimated parameters. This novel approach inherently explores topic correlation, and benefits from both the probabilistic topic models and word embedding approaches.
- We reveal the connections between the generative stochastic block model (SBM) and graph neural networks (GNNs), especially graph attention network (GAT). According to our mathematical deductions, GAT is equivalent to the Semi-Amortized inference algorithm of SBM. This observation may facilitate the development of novel GNNs for different kinds of graphs.
- We observe that the probabilistic latent semantic indexing (pLSI), which is a probabilistic topic model, can be seen as SBM on a specific bi-partite graph, where the documents and the words are the two kinds of the nodes, respectively.
- To relax the i.i.d. data assumption of vanilla amortized inference, we pioneer to propose a novel graph neural network model, named Graph Attention TOpic Network (GATON), for correlated topic modeling. GATON, which constructs the graph topology with the bi-partite graph of documents and words, explores the topic structure by convolving the node attributes over the graph with an attention mechanism.

2 RELATED WORKS

Topic model originates from the dimension reduction in information retrieval. Latent semantic indexing (LSI) adopts the singular value decomposition (SVD) of the term-by-document matrix [10]. Probabilistic latent semantic indexing (pLSI) develops a generative probabilistic model of the text corpora by assuming each document being a mixture of the topics [18]. To alleviate the overfitting to the training data, Latent Dirichlet Allocation (LDA) is introduced by imposing latent variables with Dirichlet prior. To relax the topic independence assumption induced by the Dirichlet prior, Correlated Topic Model (CTM) adopts a logistic-normal prior to explicitly model the correlations with a Gaussian covariance matrix [4]. Unfortunately, non-conjugacy of logistic-normal prior requires huge computations, which can be slightly reduced by introducing independent factor models or efficient sampling [7, 28].

Neural Variational Topic Modeling. Recently, the development of deep generative networks and stochastic variational inference enable the variational inference via neural network, i.e., neural variational inference. Auto-encoding variational Bayes provides a general framework for deep generative model, and its example variational auto-encoder (VAE), which consists of a generative network (decoder) and a inference network (encoder), is designed for the generation of i.i.d. data, such as images. Neural variational document model (NVDM) [36] applies VAE to unsupervised document modeling by assuming document with bag-of-words representation as i.i.d data. Neural variational latent Dirichlet allocation (NVLDA) [46] alleviates the difficulty of non-location scale family of the Dirichlet prior by replacing the Dirichlet prior with Laplace approximation, which is also a Gaussian distribution. Gaussian Softmax Model (GSM) [35] extends NVDM [36] by constructing the topic distribution with a softmax function, which is applied to the projection of the Gaussian random vectors. Neural Variational Correlated Topic Modeling (NVCTM) [31] overcomes the common drawback of above NVI approaches, which are incapable of modeling topic correlations due to the isotropic Gaussian topic distribution, by proposing Centralized Transformation Flow to reshape the topic distribution. The main drawback of existing Neural Variational Topic Modeling is the assumption that the documents should be i.i.d to adopt VAE. In fact, the documents are composed of words, which tend to be correlated instead of completely independent. Therefore, the correlations between documents are critical for topic modeling and under-fitted by NVI.

Topic Modeling with Word Embedding. Recently, some methods are proposed to explore topic correlations in embedding space [13, 16, 29, 30]. With the help of the similarity information contained in word embedding, topic modeling can be significantly improved. Usually, the approaches of topic modeling with word embedding can be divided into two categories. Methods in the first category directly introduce topic embedding into the word embedding approaches. Collaborative Language Model [51] collaboratively models the topics and learns the word embeddings by considering complementary global and local context information based on matrix factorization. Skip-gram Topical word Embedding (STE) [45] learns the word embeddings and latent topics in a unified skip-gram framework to obtain the topic-specific word embeddings, and thus addresses the issue of polysemy. Although this kind of methods seamlessly integrates topic modeling into word embedding framework, the statistical characteristics of documents are lost.

Methods in the second category tend to modify the generative process of the topic model by exploiting word embedding [9, 19, 39, 42, 52, 56, 57]. Word Featured LDA (WF-LDA) [42] treats the word information as features rather than an explicit constraint and relaxes the single global hyper-parameter for topic's word distribution to multiple ones according to the word similarity. To reduce the out-of-vocabulary (OOV) words caused by the fixed vocabulary of word types, Gaussian LDA [9] replaces topic's Multinomial distribution over word with multivariate Gaussian distributions in the word embedding space. Latent feature LDA (LF-LDA) [39] incorporates the inner product of topic and word embeddings in modeling the topic-word distributions to relax the assumption that topics are unimodal in the embedding space in Gaussian LDA [9]. Latent Concept Topic Model (LCTM) [19] introduces a latent concept between topic and word, and models each topic as a distribution over the latent concepts, each of which is a localized Gaussian distribution in the word embedding space. Correlated Gaussian Topic Model (CGTM) [52] replaces words in documents with meaningful word embeddings, and models topics as multivariate Gaussian distributions in the word embeddings.

Although statistical characteristics of documents are retained in the methods of the second category, word embedding is only exploited to constrain the generation process of the documents, thus it has not been fully explored.

3 PRELIMINARIES

3.1 Notations

The corpora consists of M documents $O = \{o_1, o_2, ..., o_M\}$. A document o is a sequence of N_o words, $o = \{w_1, w_2, ..., w_{N_o}\}$, where w_n is the n^{th} token in the sequence and drawn from a U-words vocabulary. For simplicity, each token w is represented as a U-dimensional unit-basis vector $w = \{w^1, w^2, ..., w^U\}$, where only a single element equals to one and all the others equal to zero. If $w^u = 1$, the token w is occupied by the u^{th} word in the vocabulary. For topic modeling problem, the number of topics, T, is given.

An attributed network can be modeled as an attributed graph $G = (V, E, X)$. $V = \{v_i | i = 1, ..., N\}$ is a set of N vertices, each of which, v_i, is associated with an attribute $x_i \in \mathbb{R}^F$. Network topology is composed by a set of edges, $E = \{e = (v_i, v_j)\}$, each of which connects two vertices in V. $X = [x_{ij}] \in \mathbb{R}^{N \times F}$ represents the collection of the attribute features. Each row of X, i.e., x_i^T, corresponds to the attributes of a node. For convenience, $x_i \in \mathbb{R}^F$ and $x_{\cdot,j} \in \mathbb{R}^N$ are utilized to denote the i^{th} row (all the attributes of vertex v_i) and j^{th} column (the j^{th} attribute of all the vertices) of X in vector form, respectively. Besides, the adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{N \times N}$ represents the network topology, where $a_{ij} = 1$ if an edge connects the vertices v_i and v_j, and vice versa. $d_n = \sum_j a_{nj}$ stands for the degree of v_n and $D = \text{diag}(d_1, d_2, ..., d_N)$ is the degree matrix of A. The graph Laplacian and its normalized form are defined as $L = D - A$ and $\hat{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$, respectively. Generally, the number of communities, K, is given.

3.2 Amortized Inference

In probabilistic generative models, the key task is to estimate the posterior distributions of the latent variables and parameters given the observed variables. Since this evaluation is infeasible in many practical cases, variational inference (VI), which is a deterministic approximation scheme, analytical approximates $q_i(z_i|\lambda_i)$ to the posterior distribution by making some assumptions about the form of posterior distribution, such as Gaussian distribution. Variational inference reformulates the estimation problem as the optimization of parameter λ_i [20]. This optimization, however, is challenging for large datasets and non-conjugate models, because it separately updates each latent variable with a conjugate posterior distribution

$$\lambda_i = \lambda_i + \epsilon\nabla\text{ELBO}(\lambda_i, x), \tag{1}$$

where ELBO is the evidence lower bound. To alleviate this issue, amortized variational inference (AVI) is developed to reformulate the variational inference as a prediction neural network which is shared (amortized) across all the data in the dataset [21, 22, 34]. The prediction neural network maps the data to the parameters of its posterior distribution or latent representation as

$$\lambda_i = f(x_i, \phi), \tag{2}$$

with the shared parameter ϕ. AVI can alleviate the overfitting issue caused by the large amount of parameters λ_i via sharing the parameter ϕ. Variational auto-encoder (VAE) is an example which utilizes AVI to jointly train the generative and inference network [22, 44].

4 SBM VS. GAT

In this section, a well-known generative community detection method, stochastic block model (SBM) with EM optimization algorithm, and a well-behaved graph neural network, graph attention network (GAT), are accordingly reviewed. Then, the comparisons are given from the perspective of propagation. At last, from the perspective of inference, GAT is interpreted as Semi-Amortized inference of SBM which possesses the advantages of both the traditional VI and amortized VI.

4.1 Stochastic Block Model

Overlapping community detection divides the network into overlapping sub-networks by considering that each node may share certain properties with many nodes which may belong to different groups. A common strategy in overlapping community detection is to estimate the categorical distribution of each node instead of estimating all of its communities. In [2], a generative model for overlapping community detection is introduced. Given N nodes, to generate the topology, the model is parametrized by a set of parameters θ_{ik}, which represent the propensity of node v_i belonging to community k. Then, $\theta_{ik}\theta_{jk}$ is the expected number of edges in community k between the nodes v_i and thus v_j, and $\sum_k \theta_{ik}\theta_{jk}$ is the expected number of edges between the nodes v_i and v_j. The observed edge can be modeled by Poisson distribution with the mean value as the expected number of edges. As shown in [2], the adoption of Poisson distribution allows the existences of multi-edges and self-edges, which are common in most of the real-world networks. Thus, the probability of generating the observed graph G with adjacency matrix A is

$$\begin{aligned} P(G|\Theta) &= \prod_{i<j} \frac{\left(\sum_k \theta_{ik}\theta_{jk}\right)^{a_{ij}}}{a_{ij}!} \exp\left(-\sum_k \theta_{ik}\theta_{jk}\right) \qquad (3)\\ &\quad \prod_i \frac{(\sum_k \theta_{ik}\theta_{ik})^{a_{ii}/2}}{(a_{ii}/2)!} \exp\left(-\frac{1}{2}\sum_k \theta_{ik}\theta_{ik}\right). \end{aligned}$$

Note that $a_{ii} = 2$ denotes a self-edge. Taking the logarithm of Eq. (3) and omitting the constants, the formula can be transformed to

$$\log P(G|\Theta) = \sum_{i<j} a_{ij} \log\left(\sum_k \theta_{ik}\theta_{jk}\right) - \sum_{ijk} \theta_{ik}\theta_{jk}. \tag{4}$$

Directly maximizing Eq. (4) with respect to θ_{ik} cannot obtain any analytical solution. By applying Jensen's inequality

$$\log\left(\sum_k x_k\right) \geq \sum_k q_k \log \frac{x_k}{q_k},$$

where q_k is any probability satisfying $\sum_k q_k = 1$, and the exact equality can be achieved if $q_k = x_k / \sum_r x_r$. Eq. (4) can be transformed to

$$\log P(G|\Theta) \geq \sum_{ijk} \left[a_{ij} q_{ij}(k) \log \frac{\theta_{ik}\theta_{jk}}{q_{ij}(k)} - \theta_{ik}\theta_{jk}\right], \tag{5}$$

where $q_{ij}(k)$ is any probability satisfying $\sum_k q_{ij}(k) = 1$, and the exact equality can be achieved if

$$q_{ij}(k) = \frac{\theta_{ik}\theta_{jk}}{\sum_k \theta_{ik}\theta_{jk}} = g_k\left(\theta_{ik}\theta_{jk}\right), \tag{6}$$

where $g_k(.)$ denotes the normalization over the dimension of community membership. Here, $q_{ij}(k)$ is only defined for observed edges i.e. $a_{ij} = 1$. By concatenating $q_{ij}(k)$, where $k = 1, 2, ..., K$, into $q_{ij} \in \mathbb{R}^K$, the vector-form of Eq. (6) is

$$q_{ij} = \frac{\theta_i \odot \theta_j}{\theta_i^T \theta_j} = \left(\frac{\theta_i}{\theta_i^T \theta_j}\right) \odot \theta_j, \tag{7}$$

where $\theta_i \in \mathbb{R}^K$ is the concatenation of θ_{ij} and $\odot$ represents the element-wise product between two vectors. When $q_{ij}(k)$ is fixed, the optimal θ_{ik} can be obtained by maximizing Eq (5) with respect to θ_{ik} as

$$\theta_{ik} = \frac{\sum_j a_{ij} q_{ij}(k)}{\sum_i \theta_{ik}}. \tag{8}$$

By summing Eq (8) over i and multiplying with $\sum_i \theta_{ik}$, we can obtain $(\sum_i \theta_{ik})^2 = \sum_{ij} a_{ij} q_{ij}(k)$ and Eq. (8) can be reformulated as

$$\theta_{ik} = \frac{\sum_j a_{ij} q_{ij}(k)}{\sqrt{\sum_{ij} a_{ij} q_{ij}(k)}} = g_i\left(\sum_j a_{ij} q_{ij}(k)\right), \tag{9}$$

where $g_i(.)$ denotes the normalization over all the nodes. Eq. (9)'s vector-form can be represented as

$$\theta_i = g_i\left(\sum_j a_{ij} q_{ij}\right). \tag{10}$$

By alternately optimizing between Eqs. (6) and (9), the log-likelihood can be maximized. Note that the value of θ_i is randomly initialized.

This iterative optimization is equivalent to the E-step and M-step in expectation maximization (EM) algorithm and it can be proved that the log-likelihood increases monotonically as the number of iterations increases.

4.2 Graph Attention Network

Motivated by the successful applications of deep learning to the regular grid data (e.g. images and videos), developing deep learning techniques to process the irregular graph data, i.e. graph neural networks (GNNs), became popular in both theory and practice [40, 50, 53–55, 58, 59]. GNNs can be categoried into spectral ones and spatial ones. Originated from spectral graph theory, spectral GNNs apply deep operations, e.g. convolution, in spectral domain and progressively overcome the high computation complexity caused by the spectral decomposition. Motivated from a first-order approximation of spectral graph convolutions, graph convolutional network (GCN) [23] employs a layer-wise propagation rule for neural network models, which operate directly on graphs as

$$H^{(l+1)} = \sigma\left(W^{(l)}H^{(l)}\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\right), \tag{11}$$

where $\tilde{A} = A + I_N$, $\tilde{D}$ represents the degree matrix of $\tilde{A}$, i.e., the i^{th} diagonal elements $\tilde{d_i} = \sum_j \tilde{A}_{ij}$ and $W^{(l)}$ stands for the trainable weight matrix of a fully-connected layer. $H^{(l)} \in \mathbb{R}^{D \times N}$ is the representations of the l^{th} layer after propagation and $H^{(0)} = X$ contains the original node attributes. $\sigma(.)$ denotes the nonlinear activation function, such as ReLU. Eq. (11) can be formulated in a node-wise form

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in N(i) \cup i} \frac{1}{\sqrt{(d_i+1)(d_j+1)}} W^{(l)} h_j^{(l)}\right), \tag{12}$$

where $h_i^{(l)}$, the i^{th} column of $H^{(l)}$, is the representation of node v_i in the l^{th} layer. $N(i)$ represents the neighbourhood of vector v_i. Although GCN significantly improves the performance, its main drawback is the fixed propagation weight $\frac{1}{\sqrt{(d_i+1)(d_j+1)}}$, which is completely determined by the degrees of the two connected nodes.

To overcome that drawback, graph attention network (GAT) [48] attempts to learn the propagation weight by leveraging the self-attention mechanism [1]. It alternately proceeds between the weight learning and the attribute propagation as

$$\begin{aligned}
\alpha_{ij} &= \text{softmax}_j(a(Wh_i^{(l)}, Wh_j^{(l)})) \qquad (13)\\
&= \frac{\exp\left(\text{LeakyReLU}(b^T[Wh_i||Wh_j])\right)}{\sum_{k \in N(i)} \exp\left(\text{LeakyReLU}(b^T[Wh_i||Wh_k])\right)},\\
h_i^{(l+1)} &= \sigma\left(\sum_{j \in N(i)} \alpha_{ij} W h_j^{(l)}\right),
\end{aligned}$$

where $a(.,.)$ stands for a neural network, $||$ denotes the concatenate operator and $b \in \mathbb{R}^{2D'}$ is the learnable weight vector. To facilitate the discussion of connections between GAT and Community Detection, the above two steps can be reformulated as the following three steps:

$$\begin{aligned}
h_i' &= Wh_i^{(l)} \qquad (14)\\
h_{ij}'' &= \alpha_{ij} h_j' \qquad (15)\\
&= \text{softmax}_j(\text{LeakyReLU}(b^T[h_i'||h_j'])h_j'\\
h_i^{(l+1)} &= \sigma\left(\sum_j a_{ij} h_{ij}''\right), \qquad (16)
\end{aligned}$$

where W and b are two learnable parameters. This iterative optimization is initialized with node attributes as

$$h_i^{(0)} = x_i. \tag{17}$$

Eq. (14) can be considered as a mapping from h_i to h_i' through a fully-connected neural network parameterized by W. Eq. (15) can be regarded as the propagated information from v_j to v_i, whose weight α_{ij} is obtained from a mapping parameterized by b, which takes the concatenation of h_i' and h_j' as input. After the above two steps, Eq. (16) propagates the information h_{ij}'' from v_j to v_i. The predicted labels $Y = [y_{ik}] \in \mathbb{R}^{N \times K}$ are the output of the last layer, i.e. $h_i^{(L)}$, where $i = 1, 2, ..., N$. For simplicity, $Y_l = [y_{ik}] \in \mathbb{R}^{|V_l| \times K}$ contains the predicted labels of the nodes with given labels. The parameters W and b can be learned by minimizing the cross-entropy between the predicted labels and ground-truth labels on the labelled nodes

$$\mathcal{L} = -\sum_{v_i \in V_l} \sum_{k=1}^{K} z_{ik} \ln y_{ik}. \tag{18}$$

This attention mechanism has also been extend to model meta-paths in heterogeneous information network [49] and signed edges in signed network [12].

4.3 Comparisons

The complete comparisons between SBM (Eqs. (7) and (10)) and GAT (Eqs. (15) and (16)) are shown in Table 1. It can be observed that both of them exploit a similar weighted propagation principle (Eqs. (10) and (16)). Their detailed differences are given as follows.

- **Latent Variable:** In SBM, the latent variable is initialized randomly without any learning strategy. On the contrary, GAT learns the latent variables from the corresponding node attributes via a fully-connected (FC) layer which is parameterized by W. Similar to the amortized inference in Eq. (2), this FC layer is shared across all the nodes.
- **Propagation Weight:** In SBM, the element-wise propagation weight is optimized via gradient descent without any learnable parameters. On the other hand, GAT learns the edge-wise weights from the concatenated latent variables of the connected nodes with a regression model which is parameterized by b. The regression model is also shared across all the edges, which is also equivalent to that of the amortized inference in Eq. (2).

In general, SBM *infers* each latent variable and propagation weight separately, while GAT *learns* the latent variables and propagation weights based on an amortized (shared) scheme. According to their similarities and differences, the connections between GAT and SBM are summarized in the next subsection.

Table 1: Comparisons between Stochastic Block Model and Graph Attention Network.

	Stochastic Block Model	Graph Attention Network
Latent Variable	θ_i (community membership)	h_i (node representation)
Initialization	random initialization	x_i (node attributes)
Amortized Mapping	without mapping	$h'_i = Wh_i^{(l)}$ with learnable parameter W
Propagation Weight	$\frac{\theta_i}{\theta_i^T \theta_j}$	$\text{softmax}_j(\text{LeakyReLU}(b^T [h'_i \Vert h'_j])$
Propagation Weight Granularity	element-wise	edge-wise
Propagation Weight Learnability	without learnable parameters	with learnable parameter b
Propagated Information	θ_i (original latent variable)	h'_i (latent variable after mapping)
Weighted Information	$q_{ij} = \left(\frac{\theta_i}{\theta_i^T \theta_j}\right) \odot \theta_j$	$h''_{ij} = \text{softmax}_j(\text{LeakyReLU}(b^T [h'_i \Vert h'_j])h'_j$
Propagation Rule	$\theta_i = g_i\Big(\sum_j a_{ij} q_{ij}\Big)$	$h_i^{(l+1)} = \sigma\Big(\sum_j a_{ij} h''_{ij}\Big)$

4.4 GAT as Semi-Amortized Inference of SBM

Recall that the amortized inference, such as VAE [22], directly maps the inputs to the representations or latent variable parameters as shown in Eq. (2) with an i.i.d. assumption of the data. However, the nodes in networks are usually connected and correlated. Therefore, the direct adoption of the amortized inference to the network data may cause under-fitting of the correlations among nodes.

GAT provides a novel scheme to incorporate the Amortized Inference in modeling the network. This novel scheme, named as Semi-Amortized Inference, combines the amortized inference with the traditional variational inference. As shown in Eqs. (14), (15) and (16), GAT performs latent variable propagations (Eq. (16)) as well as the amortized inference step (Eqs. (14) and (15)). According to the analysis in Section 4.3 and Table 1, the propagation rules of SBM and GAT are identical. Note that the propagations in SBM are performed in the M-step in EM algorithm, which possesses the same philosophy as variational inference [3]. Therefore, the propagations (Eq. (16)) in GAT can be considered as the traditional variational inference, which alleviates the difficulty of the amortized inference on modeling the node correlations in the network. In summary, **GAT can be regarded as the Semi-Amortized Inference (SAI) of SBM, which alternately performs the amortized inference (Eqs. (14), (15)) and traditional inference (Eq. (16)).** SAI can simultaneously provides a fast inference speed and certain flexibility of modeling correlations.

5 PROPOSED MODELS

In this section, the probabilistic latent semantic indexing (pLSI) is firstly reviewed and reformulated as stochastic block model (SBM) of a specific bi-partite graph. Then, a novel graph neural network for topic modeling, Graph Attention TOpic Network, is presented according to the observations in Section 4.4.

5.1 Topic Modeling as SBM on Bi-partite Graph

Probabilistic latent semantic indexing (pLSI) is a well-known generative topic model and the basis of many topic modeling approaches. For example, Latent Dirichlet Allocation introduces the Dirichlet priors to latent variables of pLSI to alleviate its overfitting issue. pLSI assumes the following generative process for each document o in a corpus O:

(1) Choose the number of word $N_o \sim \text{Poisson}(\eta_o)$ for document o;
(2) For each of the N_o words w_{on} in document o;
  (a) Choose a topic $z_{on} \sim \text{Multinomial}(\theta_o)$;
  (b) Choose a word $w_{on} \sim \text{Multinomial}(\beta_{z_{on}})$.

The probabilistic graphical model of this generative process is represented in Figure 2(a). In the generative process, $\eta_o \in \mathbb{R}$ is the Poisson parameter to determine the length of the document. In many models, such as LDA, this process is omitted by assuming all the documents possessing the same length. $\theta_o \in \mathbb{R}^T$ is the topic distribution of document o. $\beta_t \in \mathbb{R}^U$ denotes the word distribution of topic t. η, Θ and B represent the collection of η_o, θ_o and β_t, respectively. $z_{on} \in 1, 2, ..., T$ stands for the topic assignment of n^{th} word in document o.

The likelihood of the above model on an observed corpora O is

$$\begin{aligned} P(O|\eta, \Theta, B) &= \prod_{o=1}^{M} p(N_o|\eta_o) \prod_{n=1}^{N_o} \sum_{z_{on}=1}^{T} p(z_{on}|\theta_o) p(w_{on}|z_{on}, B) \\ &\propto \prod_{o=1}^{M} \eta_o^{N_o} \exp(-\eta_o) \prod_{n=1}^{N_o} \sum_{z=1}^{T} \prod_{u=1}^{U} (\theta_{oz}\beta_{zu})^{w_{on}^u}. \quad (19) \end{aligned}$$

By letting $n_{ou} = \sum_{n=1}^{N_o} w_{on}^u$ be the frequency of word u which appears in document o, and normalizing the multinomial distribution, Eq. (19) can be rewritten as

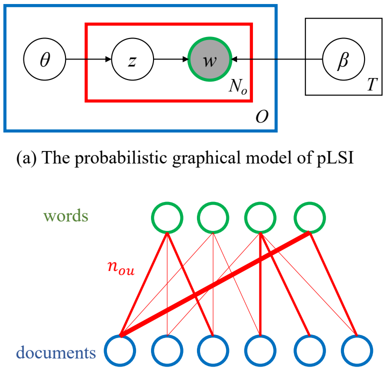
$$P(O|\eta, \Theta, B) = \prod_{o=1}^{M} \eta_o^{N_o} \exp(-\eta_o) \prod_{u=1}^{U} \frac{(\sum_{z=1}^{T} \theta_{oz}\beta_{zu})^{n_{ou}}}{n_{ou}!}. \quad (20)$$

Since $\sum_{u=1}^{U} n_{ou} = N_o$ is the number of word in document o and

$$\eta_o^{N_o} \sum_{u=1}^{U} \sum_{z=1}^{T} \theta_{oz}\beta_{zu} = \eta_o^{N_o} \sum_{z=1}^{T} \theta_{oz} = \eta_o^{N_o},$$

Eq. (20) can be reformed as

$$P(O|\eta, \Theta, B) = \prod_{o=1}^{M} \prod_{u=1}^{U} \exp\left(-(\eta_o \sum_{z=1}^{T} \theta_{oz}\beta_{zu})\right) \frac{(\eta_o \sum_{z=1}^{T} \theta_{oz}\beta_{zu})_{ou}^{n}}{n_{ou}!}.$$



(a) The probabilistic graphical model of pLSI

(b) The bi-partite graph of pLSI

Figure 2: Two representations of pLSI. The blue and green objects denote the documents and words, respectively. The red objects (both lines and boxes) represent the words contained in the documents. (a) The probabilistic graphical model representation of pLSI. The red box represents the words (green circles) contained in the document (blue box). (b) The bi-partite graph representation of pLSI. The red lines (edges), which connect document nodes and word nodes in bi-partite graph, indicate that documents contain words. The weight of edge, which connects document o and word u, is the frequency of word u in document o.

By absorbing η_o into parameters θ_{oz} as $\theta'_{oz} = \eta_o\theta_{oz}$, Eq. (20) can be revised as

$$P(O|\Theta, B) = \prod_{o=1}^{M}\prod_{u=1}^{U} \exp\left(-\sum_{z=1}^{T}\theta'_{oz}\beta_{zu}\right)\frac{(\sum_{z=1}^{T}\theta'_{oz}\beta_{zu})^{n_{ou}}}{n_{ou}!}. \quad (21)$$

By comparing the likelihood of pLSI in Eq. (21) with that of SBM in Eq. (3), pLSI and SBM can be connected [15]. The latent variable θ_{ik} in SBM is the propensity of node v_i belonging to community k, while the latent variable θ_{oz} and normalized $\beta'_{uz} = \beta_{zu}/\sum_t \beta_{tu}$ in pLSI are the propensities of document o and word u belonging to topic z, respectively. The community in SBM is similar to the topic in pLSI. The edge A_{ij} in SBM corresponds to the n_{ou}, which is the frequency of word u in document o, in pLSI. Thus, pLSI can be regarded as the SBM of a specific graph, i.e., bi-partite graph, where documents o and words u are the two kinds of nodes, respectively. As shown in Figure 2(b), only the edges between words and documents exist in this bi-partite graph, and the edge weights is n_{ou}, i.e., the frequency of word u in document o. Therefore, the topic modeling problem can be regarded as a specific bi-partite graph modeling problem, and many network modeling approaches can be then adopted to identify different topics.

5.2 Graph Attention TOpic Network

In this subsection, a novel graph neural network, Graph Attention TOpic Network (GATON), is proposed for topic modeling. The motivation of GATON bases on the above two interpretations, which are also shown in Figure 1: interpretation of pLSI as the SBM of a specific bi-partite graph in Section 5.1 and interpretation of GAT as semi-amortized inference of SBM in Section 4.4. Therefore, GATON is designed to follow the semi-amortized inference of SBM on a bipartite network.

5.2.1 Node Attribute. As reviewed in Section 3.2 and interpreted in Section 4.4, semi-amortized inference significantly reduces the number of parameters (latent variables) by learning the shared (amortized) function which maps the node attributes to latent variables. SBM divides network only based on its topology, while GAT leverages node attributes to help the inference of SBM. Similarly, the bipartite network, which pLSI is equivalent to, only possesses topology structure yet lacks the node attributes. Therefore, the adoption of semi-amortized inference on the bi-partite graph, which consists of document nodes and word nodes, assigns the attributes to both the document and word nodes. Note that the assigned attributes should reflect the individual properties of each node. Besides, the information contained in node attributes may better be different from that contained in network topology. The overall configuration of GATON is shown in the left subfigure of Figure 3.

Word node attribute. Since the bi-partite graph represents the inclusion relationship between documents and words, word node attributes should reflect the semantics of the word itself i.e., word with similar semantics should possess similar attributes, and vice versa. Therefore, the one-hot word representation is less appropriate, because the differences of the one-hot representations of any two different words are identical. A more suitable choice for word represention is the word embedding, which embeds the word similarity into real-valued vector as

$$x_u^{word} = \text{embedding}(u), \quad (22)$$

where embedding(.) represents the embedding function such as CBOW, Skig-gram [37] and GloVe [41]. Note that word embedding is independent of the co-occurrence of words in the documents, which is reflected by the bi-partite graph topology, thus the information contained in word embedding is different from that contained in the bipartite network.

Document node attribute. The document is composed of words, thus, it is natural to model the document attributes as the bag-of-words representation of the document. Unfortunately, the bag-of-words representation has already being contained in the bi-partite graph topology, because edge reveals the inclusion relationship between document and word, and each edge weight represents the number of the corresponding words contained in the document. Note that the edge weights in GAT are not fixed. They are learned from a shared (amortized) normalized regression function, which takes the attributes of its two corresponding nodes as input, as shown in Eq. (13). Therefore, the bi-partite graph topology can be simplified to an unweighted one, and the document attributes can be modeled as term frequency vector

$$x_o^{document} = (n_{o1}, n_{o2}, ..., n_{oU}), \quad (23)$$

where U is the number of words in the vocabulary, and n_{ou} is the frequency of word u in document o. Therefore, the weighted bi-partite graph in Figure 2(b) is represented as the unweighted bi-partite graph with attributes in the left subfigure of Figure 3.

5.2.2 Amortized Inference. Similar to GAT, GATON also introduces two amortized inference steps to infer the node and edge

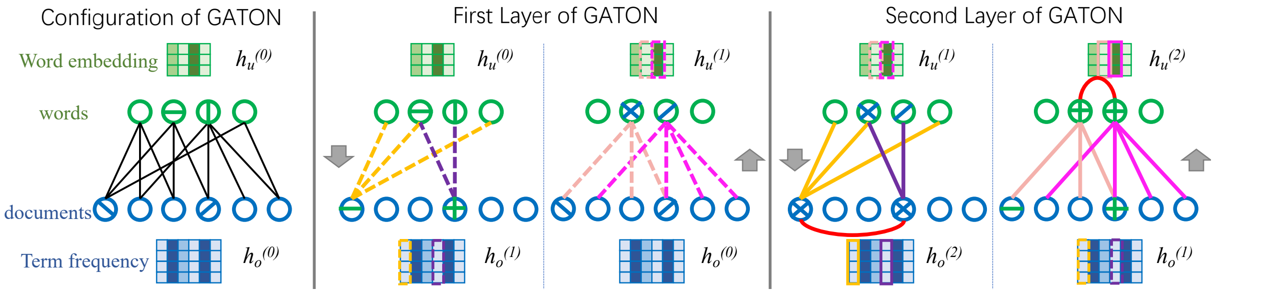


Figure 3: The propagation scheme of GATON. The left subfigure is the setup of the GATON, where the initialized representations of words and documents are word embedding (in green) and term frequency (in blue) vectors, respectively. Horizontal "-" and vertical "|" lines denote the information contained in two different word nodes, while slash "/" and backslash "\" stand for information contained in two different document nodes. The crosses ("+" or "×") represent the combinations of information from two different (word or document) nodes. The middle and right subfigures are the first and second layers of GATON, which consist of two components: propagations from word to document (top-down arrow) and propagations from document to word (bottom-up arrow). The one-layer propagation explores the relationships between words and documents, while two-layer propagations explore the relationships (word-word and document-document) between the same kinds of nodes as indicated by the red lines, which connect two nodes that have exchanged their information (indicated by crosses in nodes).

latent variables with shared mapping functions. Note that this specific bi-partite graph possesses two significant differences from homogeneous networks. 1) There are two kinds of nodes, i.e., document nodes and word nodes, in the bi-partite graph, and their dimensions are different. 2) The impact between two kinds of nodes is not symmetric i.e., the impact of words to documents may be different from that of documents to words.

Since there exists two types of nodes, two different mapping functions are defined as

$$\hat{h}_u^{word} = W^{word} x_u^{word}, \quad (24)$$
$$\hat{h}_o^{document} = W^{document} x_o^{document}. \quad (25)$$

Note that W^{word} and $W^{document}$ are the parameters for two mapping functions, respectively, and they cannot be shared due to different embedding semantics and embedding dimensions.

Due to the unsymmetrical impacts, two different normalized regression functions are introduced as

$$\alpha_{o\to u} = \frac{\exp\left(\mathrm{LeakyReLU}(b_{o\to u}^T[\hat{h}_o||\hat{h}_u])\right)}{\sum_{t\in N(o)}\exp\left(\mathrm{LeakyReLU}(b_{o\to u}^T[\hat{h}_o||\hat{h}_t])\right)}, \quad (26)$$

$$\alpha_{u\to o} = \frac{\exp\left(\mathrm{LeakyReLU}(b_{u\to o}^T[\hat{h}_u||\hat{h}_o])\right)}{\sum_{z\in N(u)}\exp\left(\mathrm{LeakyReLU}(b_{u\to o}^T[\hat{h}_u||\hat{h}_z])\right)}, \quad (27)$$

where $\alpha_{o\to u}$ and $\alpha_{u\to o}$ denote the attention from document o to word u and that from word u to document o, respectively. Note that $\hat{h}_o$ and $\hat{h}_z$ are the representations of documents as in Eq. (24), and $\hat{h}_u$ and $\hat{h}_t$ are the representations of words as in Eq. (25). For simplicity, superscripts are omitted and o and u represent document and word, respectively. $N(o)$ and $N(u)$ are the neighbours of document o and word u, respectively. Note that all the neighbours of a document are the words belonging to it, while all the neighbours of a word are the documents containing this word. $b_{o\to u}^T$ and $b_{u\to o}^T$ stand for the parameters of the impact of document to word and that of word to document, respectively, and they cannot be shared due to the different roles.

5.2.3 *Propagation as Semi-Amortized Inference.*

So far, the representations of two kinds of nodes and the edge weights of two directions have been obtained. As in GAT, the propagations are carried out via semi-amortized inference. In GATON, the propagations are bidirectional as

$$h_o = \sigma\Big(\sum_{t\in N(o)} \alpha_{u\to o}\hat{h}_t\Big), \quad h_u = \sigma\Big(\sum_{z\in N(u)} \alpha_{o\to u}\hat{h}_z\Big), \quad (28)$$

where $\sigma(.)$ is the nonlinear activation function, such as ReLU. This process is shown in Figure 3, which illustrates the propagations of one-layered GATON. In the propagation, the first-order inclusion relationship of word in document is explored. On one hand, by propagating information from word to document, the representations of documents can absorb the information possessed by the word embedding, which reflects the word semantic. Then, the documents, which contain similar words or words with similar semantic, can obtain similar representations. On the other hand, by propagating information from document to word, the representations of words can acquire information contained in the documents, to which the word belongs. Then, the words, which belong to similar document set or document with similar word distribution, can possess the similar representations. Similar to GAT [48], multi-head attention [47] can be employed to stabilize the learning process. The final h_o and h_u are the concatenation of the results from S independent attention mechanisms.

5.2.4 *High-order Propagation.*

Although one-layered propagation captures the inclusion relationships, the high-order relationships (word-to-word, document-to-document) cannot be exploited. Thus, multiple GATON layers are stacked to model the multi-hop relationships. Taking two-layered GATON as an example (shown in Figure 3). In the first layer, word embedding information (green short lines) is utilized to enhance the representations of documents (blue circles), while document information (blue short lines) is adopted to improve the representations of words (green circles). In the second layer, word representation, which has been improved with document information (blue short lines in green circles), is exploited to further revise the representations of documents for the second time via propagations, thus it can explore the document-document relationships (red line between the blue circles with blue crosses). Similarly, the word-word relationships can also be explored (red line between the green circles with green crosses).

To achieve the amortized inference in different layers, Eqs (24) and (25) are extended to

$$\hat{h}_u^{l+1} = W_u^{l+1} h_u^l, \tag{29}$$

$$\hat{h}_o^{l+1} = W_o^{l+1} h_o^l, \tag{30}$$

where h_u^l and h_o^l are the word and document representations of the previous layer, $h_u^0 = x_u^{word}$ and $h_o^0 = x_o^{document}$, respectively. Besides, the attentions ($\alpha_{o \to u}$ and $\alpha_{u \to o}$) in Eqs. (26) and (27), and propagations in Eq. (28) are also extended to multiple-layer.

5.3 Objective Function

The parameters of GATON are $\Omega = \{W_o^{(l)}, W_u^{(l)}, b_{u \to o}^{(l)}, b_{o \to u}^{(l)} | l \in \{1, 2, ..., L\}\}$, where L is the number of layers, and the final representations of words and documents are $H^{word} = \{h_u^{(L)} | u \in \{1, 2, ..., U\}\}$ and $H^{document} = \{h_o^{(L)} | o \in O\}$, respectively. To obtain them, the reconstruction error of the bi-partite graph from the representations of words and documents H^{word} and $H^{document}$,

$$\mathcal{L}(\Omega) = \sum_{n_{ou} \neq 0} ||n_{ou} - < h_o^{(L)}, h_u^{(L)} > ||_2^2 + \lambda ||\Omega||_2^2,$$

is minimized with respect to the parameter Ω, where $< h_o^{(L)}, h_u^{(L)} >$ is the inner product of $h_o^{(L)}$ and $h_u^{(L)}$ and λ is the parameter employed to balance the reconstruction error and regularization.

6 EVALUATIONS

Datasets. Two text datasets, 20NewsGroups[1] and Reuters-21578[2] are employed for performance evaluation. 20NewsGroups dataset, 20News for short, consists of 18,846 newsgroup documents (11,314 for training and 7,532 for testing), which are classified into 20 categories. Reuters-21578, Reuters for short, contains about 10,000 documents. Due to the high imbalanced numbers of the documents in all the categories, only the 7,674 documents in the largest 8 categories, are employed. In the preprocessing step, stop words and words with total frequency lower than 10 are removed, and all the words are converted to lowercase.

Table 2: Topic coherence performances on both datasets.

Dataset	20News			Reuters		
#Top-words	5	10	20	5	10	20
NMF	-18.05	-85.53	-417.19	-11.28	-66.41	-335.61
pLSI	-15.15	-78.59	-365.69	-13.22	-70.07	-333.57
LDA	-15.30	-80.48	-368.82	-12.09	-69.80	-352.29
Gauss-LDA	-19.45	-94.52	-435.90	-24.22	-108.45	-478.43
LF-LDA	-16.58	-78.54	-385.73	-13.26	-71.35	-369.00
CLM	-11.62	-60.30	-282.79	-11.48	-63.08	-313.45
GATON-C	**-10.17**	**-55.82**	-245.29	**-10.06**	-57.46	-285.90
GATON-S	-10.92	-55.98	**-244.73**	-10.35	**-56.75**	**-277.34**
GATON-G	-11.55	-58.13	-285.91	-11.66	-61.03	-299.35

Settings. For all the datasets, we employ a two-layered GATON model. Word embeddings from CBOW, Skip-gram [37] and GloVe [41] are adopted to model the node attributes, and the correspondingly constructed GATONs are named as, GATON-C, GATON-S and GATON-G, respectively. The dimensions of word embeddings are set to 50 for all the experiments. The first layer adopts only 4 attention heads and the exponential linear unit (ELU) [8] as the nonlinear activation function. The second layer consists of only one attention head and it employs the softmax for nonlinear mapping. For the topic discovery task, the output dimension of the second layer is set as the number of topics. For the word embedding task, the output dimension of the second layer is set as the same as that of the other word embedding methods, i.e. 50. For the document classification task, the output dimension of the second layer is set to 200. During the training process, the L2 regularization with $\lambda = 0.0005$ and the dropout with $p = 0.6$, is applied to the inputs of both layers and attention coefficients. The model is initialized by Glorot and optimized via Adam SGD whose initial learning rate is 0.002.

6.1 Topic Coherence

Metric and baselines Coherence score [24, 38] is a well-adopted metric to evaluate the coherence of topics. Intuitively, it should measure the frequency that top words in the same topic co-occur in documents. Given R topic words $U^t = \{u_1^t, u_2^t, ..., u_R^t\}$ of topic t, its coherence score with respect to U^t is defined as

$$C(t, U^t) = \sum_{r=2}^{R} \sum_{l=1}^{r} \log \frac{JJ(u_r^t, u_l^t) + 1}{J(u_l^t)},$$

where $J(u_l^t)$ is the number of documents containing word u_l^t, while $JJ(u_r^t, u_l^t)$ is the number of documents containing both u_l^t and u_r^t. Then, the overall quality can be measured by the average of the coherence score over K topics as $\tilde{C} = \frac{1}{T} \sum_{t=1}^{T} C(t, U^t)$. For fair evaluation, the number of topic words are set from 5, 10 and 20.

Six baseline methods are employed for comparison. Among them, LDA [5], Non-negative Matrix Factorization (NMF) [26] and pLSI [18] are the topic modeling approaches without word embedding, while Gauss-LDA [9], LF-LDA [39] and CLM [51] are the ones based on word embedding, which is obtained from Skip-gram [37].

Results analysis. The results are shown in Table 2. It can be observed that pLSI, LDA and LF-LDA achieve similar performances. Although Gauss-LDA and LF-LDA leverage the word embedding, their performances on topic coherence do not achieve any improvement. The performance of CLM is better than other baselines on the 20News dataset, but it only obtains similar performance as NMF on the Reuters dataset. This phenomenon may be caused by the detailed scheme of incorporating word embedding, which interferes the topic modeling. GATON consistently outperforms others on both datasets. Our outstanding performances may be induced by the approach of incorporating word embedding, which only reduces the number of parameters and facilitates the inference.

6.2 Document Classification

Metric and Baselines. Macro-averaged precision, recall and F1-score are adopted as the metrics, because macro-averaging is more informative for dataset with imbalanced categories.

[1] http://qwone.com/jason/20Newsgroups/

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

Table 3: Document classification performances on datasets.

Dataset	20News			Reuters		
Metrics	Prec.	Recall	F1	Prec.	Recall	F1
NMF	0.704	0.701	0.697	0.911	0.877	0.891
pLSI	0.722	0.712	0.709	0.919	0.896	0.906
LDA	0.727	0.722	0.719	0.888	0.870	0.879
Gauss-LDA	0.309	0.265	0.227	0.462	0.315	0.353
LF-LDA	0.716	0.714	0.709	0.893	0.591	0.661
CLM	0.825	0.818	0.816	0.944	0.916	0.929
TWE	0.525	0.466	0.437	0.794	0.512	0.626
PV-DBOW	0.510	0.491	0.459	0.755	0.505	0.549
PV-DM	0.428	0.386	0.361	0.681	0.434	0.507
TopicVec	0.713	0.713	0.712	0.925	0.921	0.922
MeanWV	0.704	0.703	0.701	0.920	0.896	0.905
TV+Mean	0.718	0.715	0.716	0.922	0.916	0.916
GATON-C	0.822	0.803	0.812	**0.975**	**0.979**	**0.977**
GATON-S	**0.859**	**0.842**	**0.850**	0.944	0.937	0.940
GATON-G	0.716	0.767	0.741	0.914	0.896	0.905

In addition to the baseline methods used in Section 6.1, other six approaches, which exploits word embedding, are employed for comparison. Topical Word Embeddings (TWE) [32] incorporates topic embedding into the Skip-gram word embedding framework. PV-DBOW and PV-DM [25] are doc2vec models. TopicVec [30] is a generative topic model which also considers the word sequence. MeanWV takes the mean of word embedding in TopicVec [30]. TV+Mean is the concatenation of the TopicVec and MeanWV.

Results Analysis. The results are shown in Table 3. It can be observed that GATON-S and GATON-C remarkably outperform the baseline methods, including TopicVec [30] and CLM [51] which are the state-of-the-arts on integrating word embedding and topic modeling. Different performances may be caused by different mechanisms on integrating word embedding and topic modeling. Most of the existing approaches only simply assume that the representations of documents and words should contain the information in both the word similarity and co-occurrence. For example, TWE, MeanWV, PV-DBOW and PV-DM all represent the documents by utilizing the embeddings of words which are contained in it. The proposed GATON, however, explicitly considers heterogenous impacts and high-order relationships between words and documents by weighting the impacts between documents and words based on attention mechanism and stacking multiple layers.

6.3 Word Embedding

Here, the performance of word embedding refinement is evaluated. The refined word embeddings in GATON are the representations of word nodes in bi-partite graph, i.e., h_u's.

Metric and Baselines. The quality of word embedding is measured based on the Spearman's rank-order correlation with the human ratings. Word pairs are ranked based on their cosine similarities in embedding space. The ranking of word pairs is compared to the human-assigned similarity scores in seven datasets including WordSim353 (W353) [14] (WordSim Relatedness (WSim) and WordSim Similarity (WRel)), Men [6], Turk [43], SimL [17], and Rare [33].

Table 4: Word embedding performances on 20News dataset.

	W353	WRel	WSim	Men	Turk	SimL	Rare
SPPMI	0.461	0.444	0.465	0.444	0.551	0.131	0.245
SPPMI+SVD	0.451	0.435	0.449	0.426	0.489	0.166	0.349
PV-DBOW	0.477	0.442	0.486	0.449	0.488	0.139	0.285
TWE	0.317	0.231	0.407	0.190	0.260	0.084	0.184
CLM	0.526	0.486	0.550	0.477	0.525	0.189	0.411
CBOW	0.488	0.451	0.494	0.432	0.529	0.151	0.407
Skip-Gram	0.492	0.479	0.473	0.456	0.512	0.155	0.407
GloVe	0.300	0.279	0.320	0.192	0.268	0.049	0.230
GATON-C	**0.563**	**0.531**	**0.579**	0.505	**0.569**	0.232	0.470
GATON-S	0.552	0.527	0.573	**0.516**	0.560	**0.242**	**0.473**
GATON-G	0.461	0.405	0.460	0.352	0.435	0.154	0.358

For comparison, five word embedding methods are employed, including CBOW and Skip-gram [37], GloVe Glove, Shifted Positive PMI (SPPMI) matrix and its dimension reduced version with SVD (SPPMI+SVD)[27]. Besides, another four word embedding refinement approaches, PV-DBOW and PV-DM [25], TWE [32] and CLM [51] are also employed.

Results Analysis. The results are shown in Table 4. Most of the word embedding approaches including CBOW, Skip-gram, SPPMI and SPPMI+SVD give the similar performance. The word embedding refinement approaches, PV-DBOW, PV-DM and TWE cannot significantly improve the performance since they intended to integrate word embedding into topic modeling instead improvement word embedding. Although CLM outperforms other baselines, its performance is limited by its essence of learning word embedding instead of refining word embedding. Our proposed GATON consistently achieves the best performance due to its refinement nature, which refines word embedding by taking it as input and augmenting it with the discovered topics.

7 CONCLUSIONS

In this paper, we provide a new approach to overcome the overfitting issue of pLSI by exploiting amortized inference with word embedding as input, instead of the problematic Dirichlet prior in LDA. Although the vanilla amortized inference can significantly reduce the number of parameters by replacing the inference of latent variables with a function which shares the (amortized) learnable parameters, it has limited ability to handle the i.i.d. data. Therefore, a novel graph neural network, Graph Attention TOpic Network, is proposed to model the topic structure of non-i.i.d documents, because graph neural networks are equivalent to the semi-amortized inference of SBM on non-i.i.d. network data and pLSI is equivalent o SBM on a specific bi-partite graph. Extensive experiments demonstrate that GATON's effectiveness on topic identification, document classification and word embedding.

8 ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 61972442, Grant 61802391, Grant 61772361, Grant 61702296, Grant U1636214 and Grant U1605252 and Key Program of the Chinese Academy of Sciences (No. QYZDB-SSW-JSC003).

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.

[2] Brian Ball, Brian Karrer, and M. E. J. Newman. 2011. Efficient and principled method for detecting communities in networks. *Physical Review E* 84 (Sep 2011), 036103.

[3] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[4] David M. Blei and John D. Lafferty. 2005. Correlated Topic Models. In *NIPS*. 147–154.

[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *JMLR* 3 (2003), 993–1022.

[6] Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional Semantics in Technicolor. In *ACL*. 136–145.

[7] Jianfei Chen, Jun Zhu, Zi Wang, Xun Zheng, and Bo Zhang. 2013. Scalable Inference for Logistic-Normal Topic Models. In *NIPS*. 2445–2453.

[8] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In *ICLR*.

[9] Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for Topic Models with Word Embeddings. In *ACL*. 795–804.

[10] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by Latent Semantic Analysis. *JASIS* 41, 6 (1990), 391–407.

[11] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)* (1977), 1–38.

[12] Tyler Derr, Yao Ma, and Jiliang Tang. 2018. Signed Graph Convolutional Networks. In *ICDM*. 929–934.

[13] Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2019. Topic Modeling in Embedding Spaces. *arXiv preprint arXiv:1907.04907* (2019).

[14] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: the concept revisited. In *WWW*. 406–414.

[15] Martin Gerlach, Tiago P. Peixoto, and Eduardo G. Altmann. 2018. A network approach to topic models. *Science Advances* 4, 7 (2018).

[16] Junxian He, Zhiting Hu, Taylor Berg-Kirkpatrick, Ying Huang, and Eric P Xing. 2017. Efficient correlated topic modeling with topic embedding. In *SIGKDD*. 225–233.

[17] Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. *Computational Linguistics* 41, 4 (2015), 665–695.

[18] Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *SIGIR*. 50–57.

[19] Weihua Hu and Jun'ichi Tsujii. 2016. A Latent Concept Topic Model for Robust Topic Inference Using Word Embeddings. In *ACL*. 380–386.

[20] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. 1999. An Introduction to Variational Methods for Graphical Models. *Machine Learning* 37, 2 (1999), 183–233.

[21] Yoon Kim, Sam Wiseman, Andrew C. Miller, David Sontag, and Alexander M. Rush. 2018. Semi-Amortized Variational Autoencoders. In *ICML*. 2683–2692.

[22] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *ICLR*.

[23] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[24] Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality. In *EACL*. 530–539.

[25] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML*. 1188–1196.

[26] Daniel D. Lee and H. Sebastian Seung. 2000. Algorithms for Non-negative Matrix Factorization. In *NIPS*. 556–562.

[27] Omer Levy and Yoav Goldberg. 2014. Neural Word Embedding as Implicit Matrix Factorization. In *NIPS*. 2177–2185.

[28] Aaron Q. Li, Amr Ahmed, Sujith Ravi, and Alexander J. Smola. 2014. Reducing the sampling complexity of topic models. In *SIGKDD*. 891–900.

[29] Dingcheng Li, Jingyuan Zhang, and Ping Li. 2019. TMSA: A Mutual Learning Model for Topic Discovery and Word Embedding. In *SDM*. 684–692.

[30] Shaohua Li, Tat-Seng Chua, Jun Zhu, and Chunyan Miao. 2016. Generative Topic Embedding: a Continuous Representation of Documents. In *ACL*. 666–675.

[31] Luyang Liu, Heyan Huang, Yang Gao, Yongfeng Zhang, and Xiaochi Wei. 2019. Neural Variational Correlated Topic Modeling. In *WWW*. 1142–1152.

[32] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical Word Embeddings. In *AAAI*. 2418–2424.

[33] Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better Word Representations with Recursive Neural Networks for Morphology. In *CoNLL*. 104–113.

[34] Joseph Marino, Yisong Yue, and Stephan Mandt. 2018. Iterative Amortized Inference. In *ICML*. 3400–3409.

[35] Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering Discrete Latent Topics with Neural Variational Inference. In *ICML*. 2410–2419.

[36] Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural Variational Inference for Text Processing. In *ICML*. 1727–1736.

[37] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*. 3111–3119.

[38] David M. Mimno, Hanna M. Wallach, Edmund M. Talley, Miriam Leenders, and Andrew McCallum. [n. d.]. Optimizing Semantic Coherence in Topic Models. In *EMNLP*. 262–272.

[39] Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving Topic Models with Latent Feature Word Representations. *TACL* 3 (2015), 299–313.

[40] Shirui Pan, Ruiqi Hu, Sai-fu Fung, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Learning graph embedding with adversarial training methods. *IEEE Transactions on Cybernetics* (2019).

[41] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*. 1532–1543.

[42] James Petterson, Alexander J. Smola, Tibério S. Caetano, Wray L. Buntine, and Shravan M. Narayanamurthy. 2010. Word Features for Latent Dirichlet Allocation. In *NIPS*. 1921–1929.

[43] Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *WWW*. 337–346.

[44] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *ICML*. 1278–1286.

[45] Bei Shi, Wai Lam, Shoaib Jameel, Steven Schockaert, and Kwun Ping Lai. 2017. Jointly Learning Word Embeddings and Latent Topics. In *SIGIR*. 375–384.

[46] Akash Srivastava and Charles A. Sutton. 2017. Autoencoding Variational Inference For Topic Models. In *ICLR*.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*. 5998–6008.

[48] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[49] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In *WWW*. 2022–2032.

[50] Man Wu, Shirui Pan, Xingquan Zhu, Chuan Zhou, and Lei Pan. 2019. Domain-Adversarial Graph Neural Networks for Text Classification. In *ICDM*. 648–657.

[51] Guangxu Xun, Yaliang Li, Jing Gao, and Aidong Zhang. 2017. Collaboratively Improving Topic Discovery and Word Embeddings by Coordinating Global and Local Contexts. In *SIGKDD*. 535–543.

[52] Guangxu Xun, Yaliang Li, Wayne Xin Zhao, Jing Gao, and Aidong Zhang. 2017. A Correlated Topic Model Using Word Embeddings. In *IJCAI*. 4207–4213.

[53] Liang Yang, Zhiyang Chen, Junhua Gu, and Yuanfang Guo. 2019. Dual Self-Paced Graph Convolutional Network: Towards Reducing Attribute Distortions Induced by Topology. In *IJCAI*. 4062–4069.

[54] Liang Yang, Zesheng Kang, Xiaochun Cao, Di Jin, Bo Yang, and Yuanfang Guo. 2019. Topology Optimization based Graph Convolutional Network. In *IJCAI*. 4054–4061.

[55] Liang Yang, Fan Wu, Yingkui Wang, Junhua Gu, and Yuanfang Guo. 2019. Masked Graph Convolutional Network. In *IJCAI*. 4070–4077.

[56] He Zhao, Lan Du, and Wray L. Buntine. 2017. A Word Embeddings Informed Focused Topic Model. In *ACML*. 423–438.

[57] He Zhao, Lan Du, Wray L. Buntine, and Gang Liu. 2017. MetaLDA: A Topic Model that Efficiently Incorporates Meta Information. In *ICDM*. 635–644.

[58] Shichao Zhu, Chuan Zhou, Shirui Pan, Xingquan Zhu, and Bin Wang. 2019. Relation Structure-Aware Heterogeneous Graph Neural Network. In *ICDM*. 1534–1539.

[59] Shichao Zhu, Lewei Zhou, Shirui Pan, Chuan Zhou, Guiying Yan, and Bin Wang. 2020. GSSNN: Graph Smoothing Splines Neural Networks. In *AAAI*.