The graph embedded topic model

Dingge Liang, Marco Corneli, Charles Bouveyron, Pierre Latouche

The graph embedded topic model

Dingge Liang[a,*], Marco Corneli[a,b], Charles Bouveyron[a], Pierre Latouche[c,d]

[a] *Université Côte d'Azur, INRIA, CNRS, Laboratoire J.A.Dieudonné, Maasai team, Nice, France*
[b] *Center of modeling, Simulation and Interactions (MSI), Nice, France*
[c] *Université Clermont Auvergne, CNRS, LMBP UMR 6620, Aubière, France*
[d] *Université Paris Cité, CNRS, Laboratoire MAP5, UMR 8145, Paris, France*

Abstract

Most of existing graph neural networks (GNNs) developed for the prevalent text-rich networks typically treat texts as node attributes. This kind of approach unavoidably results in the loss of important semantic structures and restricts the representational power of GNNs. In this work, we introduce a document similarity-based graph convolutional network (DS-GCN) encoder to combine graph convolutional networks and embedded topic models for text-rich network representation. Then, a latent position-based decoder is used to reconstruct the graph while preserving the network topology. Similarly, the document matrix is rebuilt using a decoder that takes both topic and word embeddings into account. By including a cluster membership variable for each node in the network, we thus develop an end-to-end clustering technique relying on a new deep probabilistic model called the graph embedded topic model (GETM). Numerical experiments on three simulated scenarios emphasize the ability of GETM in fusing the graph topology structure and the document embeddings, and highlight its node clustering performance. Moreover, an application on the Cora-enrich citation network is conducted to demonstrate the effectiveness and interest of GETM in practice.

Keywords: Graph neural networks, Topic modeling, Deep latent variable models, Clustering, Network analysis

*Corresponding author
Email address: dingge.liang@inria.fr (Dingge Liang)

1. Introduction and related work

Heterogeneous mixed-type data is a common component of real-world networks. In a scientific citation network, for example, textual information such as paper titles, abstracts, and sometimes the papers themselves, is included, as well as the graph characterizing the citation relationships as links between papers represented as nodes. The way to incorporate these two valuable sources of information under the graph structure is crucial and would affect the quality of network representations through latent embeddings. Currently available graph neural networks (GNNs) for such heterogeneous networks typically treat the texts as node attributes and the similarity between the texts is lost when aggregating the sources of information (Kipf and Welling, 2016b; Mehta et al., 2019; Wang et al., 2019). This inevitably restricts the representation ability on graph topological structure (Wang et al., 2020) and results in the loss of topic and word semantics in embedding spaces, as demonstrated in Section 2.

This work focuses on the modeling and clustering of ubiquitous text-rich networks, where each node in a network is associated with a document that contains the textual information about that node, and where the connection relationships are represented by links between each pair of nodes. Recently, numerous efforts have been made to combine graph embeddings learning with text analysis techniques like word embeddings and topic modeling. To date, numerous models have been developed based on the standard graph convolutional networks (GCNs) (Kipf and Welling, 2016a), in which node representation is produced through a convolution operation between a graph adjacency matrix encoding node interactions and a node feature matrix encoding texts as attributes.

In this line of methods, an adaptive multi-channel graph convolutional networks (AM-GCN) was proposed by Wang et al. (2020), where the authors showed that the fusion capability of GCNs on network topological structures and node attributes is inadequate. The fundamental idea behind AM-GCN is that node embeddings are simultaneously learned on topology and feature spaces, and that the final representation is a combination of embeddings from various spaces. An alternative approach based on the GCN architecture was introduced in BiTe-GCN (Jin et al., 2021) to investigate the word semantic structures. This model initially converts the original text-rich network into a bipartite network with two sorts of nodes, namely the real nodes (document nodes in the original network) and the entity nodes (words extracted from

documents), as well as three types of edges between the nodes. Three separate GCNs are used with different kinds of nodes and edges to enable message passing within the three kinds of sub-networks. Then, the final graph representation is obtained by combining the embeddings learned from each type of sub-network. Lately, as an extension of BiTe-GCN, AS-GCN (Yu et al., 2021) introduced an augmented tri-typed (document, topic and word nodes) networks. AS-GCN combines a GCN module with a neural topic model that extracts word and topic semantics from raw text. Finally, the network learning part and the topic model are jointly trained so that they can benefit from one another.

Even though these well-established techniques yield satisfactory results, they merely discover various ways to mix document and graph embeddings without modifying the basic structure of GCNs. Conversely, in this paper, we address the key limit at the core of GCNs, which results in such an important loss of information as shown in Section 2. In addition, most of the aforementioned approaches concentrate on the node classification task, whereas true labels for nodes are frequently absent in real-world networks, making the need for a robust unsupervised clustering algorithm critical.

In this work, we propose a graph embedded topic model (GETM) to integrate graph embeddings, topic modeling and node clustering in an end-to-end manner. We also examine the limitation of GCNs in fusing graph topological structure and node features, and further introduce a new document similarity-based GCN to better account for these two aspects and to improve the performance of node clustering in networks. Additionally, the word and topic embeddings are jointly learned using the embedded topic model (ETM) (Dieng et al., 2020).

1.1. Main contributions

The GETM that we propose has the following key-features:

- a document similarity-based GCN (DS-GCN) encoding approach is presented to address the semantic information loss that occurs when considering documents as node attributes and using the convolution operation;

- a latent position-based decoder is employed to preserve the graph topology and to reconstruct the graph adjacency matrix more accurately;

- another ETM-based decoder is proposed to combine topic modeling and word embeddings for more efficient reconstruction of the document-term matrix;
- a joint optimization is carried out for both document embedding and graph topology learning based on a variational auto-encoder (VAE) (Kingma and Welling, 2014b) architecture;
- an *end-to-end* node clustering approach is performed by estimating the posterior probabilities for cluster memberships. Thus, the inference procedure can automatically assign each node to its group without using any additional out-of-the-box clustering algorithms.

1.2. Organization of the paper

In Section 2, we review two traditional strategies that served as the foundation for our new architecture and point out the key limit in GCNs with an introductory example. Then, the generative model behind GETM is introduced in Section 3. A variational inference strategy and a joint optimization algorithm are discussed in Section 4, highlighting the novel structure of DS-GCN. Numerical experiments are reported in Section 5, which emphasize the main features of our methodology and validate its ability in exploiting both the graph topology and topic modeling, as well as performing node clustering in simulated networks. An application on a real-world network Cora-enrich is presented in Section 6. Finally, Section 7 provides some concluding remarks and future work.

2. Background

In this section, we briefly review two classical methods: the graph convolutional network (GCN) (Kipf and Welling, 2016a) and the embedded topic model (ETM) (Dieng et al., 2020), which serve as the basis of our new architecture in GETM. We also point out a key limit in GCNs, which motivates us to propose a new graph neural network structure.

Graph convolutional network. On the one hand, graph convolutional networks (GCNs) aim at learning latent representations of nodes of the graph by iteratively aggregating feature information from its neighbors. A multi-layer GCN can be defined with the following layer-wise propagation rule

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

where $\hat{A} = A + I_N$ is the adjacency matrix of the undirected graph G with N nodes and I_N is the $N \times N$ identity matrix in $\mathbb{R}^N$ representing self-connections. Moreover, $\hat{D}$ denotes the degree matrix such that $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$ and $\hat{D}_{ij} = 0$ if $i \neq j$. $\hat{W}^{(l)}$ is the learnable weight matrix of l-th layer. Then, $\sigma(\cdot)$ denotes an activation function such as ReLU or Sigmoid. $H^{(l)}$ denotes the learned representation in layer l, and $H^{(0)} = X$ at the top layer accounts for node feature matrix X. The node attributes in this case might be any node characteristics or some textual information. It is not, however, intended primarily for text analysis in networks.

An introductory example. In order to illustrate the rationale behind our approach, we begin with a straightforward introductory example in which the network has three nodes, each node representing a document. In addition, suppose that the documents use words from a vocabulary of size 5 words, divided into two topics. Such a data set could be characterized by the following matrices

$$\hat{A} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad W = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix},$$

where $\hat{A}$ shows two connection structures: the first two nodes/documents strongly interact (for instance they might have one or more authors in common) and the third node is disconnected. Moreover, there are two groups of texts in W: the first and the third documents use similar words while words used by the second document completely differ. Therefore, when considering both the network topology and the text information, it is natural to conclude that there are three distinct clusters in this instance. The first layer in GCN is obtained as

$$\tilde{A}H^{(0)} = \begin{pmatrix} 0.5 & 0.5 & 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 0.5 & 0.5 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}, \qquad \text{with} \quad \hat{D} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where $\tilde{A} = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}$, $H^{(0)} = W$ and $\hat{D}$ is the diagonal matrix. As it can be seen, this operation ignores the semantic content of texts and only retains the network structure. As a result, it is unable to cluster the first and the second node into two different groups.

Embedded topic model. On the other hand, the embedded topic model (ETM) is a document generative model that combines topic modeling and word embeddings. In ETM, each document from a corpus $\{w_1, \cdots, w_D\}$ is generated in terms of T latent topics, with each topic $t = \{1, \cdots, T\}$ represented by a latent embedding $\alpha_t \in \mathbb{R}^L$ in the word semantic space. Each word $v \in \{1, \cdots, V\}$ in the vocabulary is then embedded in the same space through a latent word embedding ρ_v, and ρ denotes a $L \times V$ word embedding matrix. Besides, for the d-th document, the topic proportions θ_d are assumed to be drawn from a logistic-normal distribution

$$\delta_d \sim \mathcal{N}(0, I_T), \qquad \theta_d = \text{softmax}(\delta_d). \tag{2}$$

In addition, $\beta_t = (\beta_{tv})_v$ is a vector of V probabilities where β_{tv} represents the probability that word v occurs in topic t, obtained as

$$\beta_t = \text{softmax}(\rho^\top \alpha_t), \qquad \text{with} \quad \sum_{v=1}^{V} \beta_{tv} = 1, \forall t. \tag{3}$$

A matrix β with T rows and V columns is then constructed by stacking the vectors β_t as row vectors. The likelihood of the word $m \in \{1, ...M_d\}$ in the d-th document is then given by

$$p(w_{dm}|\delta_d, \alpha, \rho) = \sum_{t=1}^{T} \theta_{dt} \beta_{tw_{dm}}. \tag{4}$$

ETM has demonstrated good performance in learning meaningful word and topic patterns. Nevertheless, it only considers the content of documents and is unable to model the possible connection structure between the documents.

3. The graph embedded topic model

In our proposed GETM, we combine the representations learned by a document similarity-based GCN and the document analysis capability of ETM to obtain a joint embedding that takes both the graph topology and document semantics into account, and to further perform an end-to-end clustering of the nodes.

3.1. Notations

In this work, each network is modeled as an undirected, unweighted, graph G with N nodes. We introduce an $N \times N$ adjacency matrix A to encode the network topology, where $A_{ij} = 1$ if there is a link between node i and node j, 0 otherwise. In addition, each node is associated with a specific document. We introduce a corpus of N documents with a vocabulary that contains V unique terms. For clarity, we set the number of documents denoted by D in ETM to $D = N$ (the number of nodes) since each node is assumed to be associated with a single document. W is a document-term matrix where each vector W_i, $i \in \{1, \cdots, N\}$ encodes the document of node i containing a collection of M_i words. W_{iv}, where $v \in \{1, \cdots, V\}$; counts the number of times that the vocable v in the dictionary appears in the i-th document.

We aim at learning latent, joint, node/document embeddings Z in a lower dimension P. Then, using this learned embedding Z, our goal is to convert it into a graph embedding in dimension F and a document embedding in dimension T, which allow us to in turn reconstruct the graph adjacency and the document-term matrices, as well as to partition the nodes of the network into K clusters. We emphasize that GETM is capable of simultaneously performing node clustering and embedding construction.

3.2. Generative model

The generative process for GETM is now detailed. First, each node is assumed to be assigned to a cluster via a random variable c_i encoding its cluster membership

$$c_i \overset{\text{i.i.d}}{\sim} \mathcal{M}(1, \pi), \quad \text{with} \quad \pi \in [0,1]^K, \ \sum_{k=1}^{K} \pi_k = 1. \tag{5}$$

Then, conditionally to its cluster membership, a latent, joint embedding z_i is generated as

$$z_i | (c_{ik} = 1) \sim \mathcal{N}(\mu_k, \sigma_k^2 I_P), \quad \text{with} \quad \sigma_k^2 \in \mathbb{R}^{+*} \text{ and } \mu_k \in \mathbb{R}^P, \tag{6}$$

independently for each node $i = \{1, \cdots, N\}$.

Based on this joint embedding, a graph topology embedding is generated as

$$\eta_i = h_\iota^{(G)}(z_i), \tag{7}$$

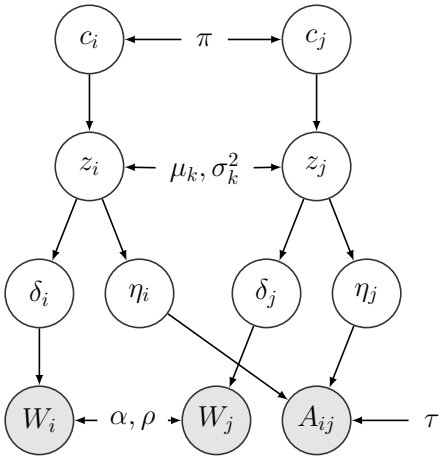

Figure 1: Graphical representation of GETM (variational parameters are not included).

where $h_\iota^{(G)}(\cdot)$ is a neural network with parameters ι to map the P-dimensional vector z_i into dimension F.

Next, the probability of a connection between nodes i and j is modeled by a distance function between two graph topology embeddings

$$A_{ij} = 1|\eta_i, \eta_j \sim \mathcal{B}(f_\tau(\eta_i, \eta_j)), \tag{8}$$

with

$$f_\tau(\eta_i, \eta_j) = \sigma(\tau - ||\eta_i - \eta_j||^2), \tag{9}$$

where $\sigma(\cdot)$ denotes a logistic sigmoid function. Here $f_\tau(\cdot)$ can be seen as the *graph decoder* parametrized by τ, which encodes the prior probability to connect.

Similarly, a document embedding is assumed to be generated based on z_i

$$\delta_i = h_\nu^{(T)}(z_i), \tag{10}$$

where $h_\nu^{(T)}(\cdot)$ is a neural network with parameter ν to map the P-dimensional vector z_i into dimension T. The topic proportions of each document i are then obtained as

$$\theta_i = \text{softmax}(\delta_i). \tag{11}$$

Finally, each document is assumed to be drawn from

$$W_i|\theta_i \sim \mathcal{M}(M_i; \theta_i^\top \beta), \quad \text{with} \quad \beta = \text{softmax}(\alpha^\top \rho). \tag{12}$$

As in ETM, α is a topic embedding representing topics in an L-dimensional space and ρ is a $L \times V$ word embedding matrix obtained typically via word2vec (Mikolov et al., 2013) or any other word embedding approach. Moreover, the product $\theta^\top \beta$ can be viewed as a *document decoder* to map the topic and word embeddings into a reconstructed document-term matrix. A graphical representation of the generative model described so far can be seen in Figure 1.

4. Inference and estimation

In this section, we detail the developed variational inference and the optimization algorithm, and introduce the proposed document similarity-based graph convolutional network (DS-GCN).

4.1. Variational inference

Before getting into the details of the inference, we first denote by $\Theta = \{\pi, (\mu_k, \sigma_k^2)_k, \iota, \nu, \tau, \alpha, \rho\}$ the set of the model parameters introduced so far. A natural procedure would consist in maximizing the integrated log-likelihood of the observed data A and W with respect to Θ

$$\log p(A, W|\Theta) = \log \int_Z \sum_C p(A, W, Z, C|\Theta) dZ. \tag{13}$$

Unfortunately, Eq. (13) is not tractable and we rely on a variational approach to approximate it

$$\log p(A, W|\Theta) = \mathcal{L}(q(Z, C); \Theta) + D_{KL}(q(Z, C)||p(Z, C|A, \Theta)), \tag{14}$$

where D_{KL} denotes the Kullback-Leibler divergence between the true and approximate posterior distributions of (Z, C) given the data and model parameters. Then, in order to deal with a tractable family of distributions, $q(Z, C)$ is assumed to fully factorize (*mean-field* assumption)

$$q(Z, C) = q(Z)q(C) = \prod_{i=1}^{N} q(z_i)q(c_i). \tag{15}$$

Moreover, to benefit from the representational learning capabilities of graph neural networks, we use a two-layer DS-GCN to encode the graph adjacency matrix and the document-term matrix into a joint embedding

$$q(z_i|A, W) = \mathcal{N}(z_i; \mu_i, \sigma_i^2 I_P), \tag{16}$$

where $\mu_i : \mathbb{R}^{N\times N} \mapsto \mathbb{R}^{N\times P}$ (respectively $\sigma_i^2 : \mathbb{R}^{N\times N} \mapsto \mathbb{R}^{+*}$) is the function mapping the normalized adjacency matrix $\tilde{A} = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}$ into the matrix of variational means (and standard deviations), parametrized by a two-layer DS-GCN defined as g_ϕ:

$$H^{(1)} = \sigma(\tilde{A} \odot (\tilde{W}\tilde{W}^\top)\hat{W}^{(0)}),$$
$$H^{(2)} = \tilde{A} \odot (H^{(1)}{H^{(1)}}^\top)\hat{W}^{(1)},$$

where $\tilde{W}$ is the normalized document-term matrix, obtained via $\tilde{W} = \frac{W}{|W|}$. $\hat{W}^{(\cdot)}$ are learnable weight matrices and $\sigma(\cdot)$ denotes a ReLU activation function. Here g_ϕ can be seen as the *encoder* that transforms two input matrices into latent, joint embeddings. The details of our proposal for such a structure are described in the following.

Finally, a standard assumption is made for variational cluster probabilities

$$q(C) = \prod_{i=1}^{N} \mathcal{M}(c_i; 1, \gamma_i), \quad \text{with} \quad \sum_{k=1}^{K} \gamma_{ik} = 1, \tag{17}$$

where γ_{ik} represents the variational probability that node i is in cluster k.

Details of DS-GCN. In contrast to standard GCNs, we here assume a new GNN architecture named DS-GCN, with the following per-layer propagation rule

$$H^{(l+1)} = \sigma(\tilde{A} \odot (H^{(l)}{H^{(l)}}^\top)\hat{W}^{(l)}), \tag{18}$$

where $\tilde{A} = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}$, $\sigma(\cdot)$ denotes the ReLU activation, and $\odot$ is an element-wise multiplication. $H^{(l)}$ encodes the learned representation in layer l, with $H^{(0)} = \tilde{W}$ the normalized document-term matrix, and $\hat{W}^{(l)}$ is the learnable weight matrix of the l-th layer. Returning to the illustration example in Section 2, we have in the top layer

$$\tilde{A} \odot (\tilde{W}\tilde{W}^\top) = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where the dot product $\tilde{W}\tilde{W}^\top$ is proportional to the cosine similarities between documents, typically used in data analysis. The product $\tilde{A} \odot (\tilde{W}\tilde{W}^\top)$ both accounts for the graph topology and document similarities. As a result, GETM is able to detect three clusters, as expected. Experiments are conducted in Section 5 to show the validity of DS-GCN.

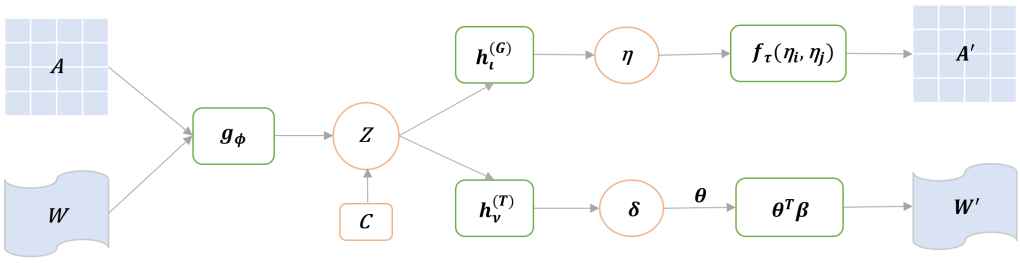

Figure 2: Model architecture of GTEM.

Model architecture. From a deep learning view, the model architecture of GETM can be seen in Figure 2. GETM takes the graph adjacency matrix A and the document-term matrix W as model inputs. Through the *DS-GCN encoder* g_ϕ, we obtain a combined embedding Z containing information about both graph topology and latent topics. Then, a latent position-based *graph decoder* f_τ is developed to map the joint embedding into a reconstructed graph matrix, and another *document decoder* $\theta^\top\beta$ is used to rebuild the document-term matrix. Additionally, by including latent cluster variables C, we are also able to explicitly optimize and eventually output a matrix $\hat{\gamma}$ that represents the clustering probabilities, and, as a result, achieve end-to-end clustering.

4.2. Optimization

In this part, we focus on maximizing the evidence lower bound (ELBO)

$$\mathcal{L}(q(Z,C);\Theta) = \int_Z \sum_C q(Z,C) \log \frac{p(A,W,Z,C|\Theta)dZ}{q(Z,C)} \tag{19}$$

with respect to the model parameters Θ and the variational parameters. Thanks to Equations (15)-(16)-(17), the ELBO denoted by $\mathcal{L}$ can be further developed as

$$\begin{aligned}\mathcal{L} &= \int_Z \sum_C q(Z,C) \log \frac{p(A,W|Z,\iota,\nu,\tau,\alpha,\rho)p(Z|C,\mu_k,\sigma_k^2)p(C|\pi)dZ}{q(Z,C)} \\ &= \sum_{i\neq j} \mathbb{E}_{q(Z|A,W)}\left[\log p(A|\iota,\tau,\eta_i,\eta_j)\right] + \sum_{i=1}^{N}\sum_{m=1}^{M_i} \mathbb{E}_{q(Z|A,W)}\left[\log p(W_{im}|\nu,\delta_i,\alpha,\rho)\right]\end{aligned}$$

$$
\begin{aligned}
&+ \mathbb{E}\left[\log \frac{p(Z|C,\mu_k,\sigma_k^2)}{q(Z|A,W)}\right] + \mathbb{E}\left[\log \frac{p(C|\pi)}{q(C)}\right] \\
&= \sum_{i\neq j} \mathbb{E}_{q(Z|A,W)}\left[\log p(A|\iota,\tau,\eta_i,\eta_j)\right] + \sum_{i=1}^{N}\sum_{m=1}^{M_i} \mathbb{E}_{q(Z|A,W)}\left[\log p(W_{im}|\nu,\delta_i,\alpha,\rho)\right] \\
&\quad - \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik} D_{KL}(\mathcal{N}(\mu_i,\sigma_i^2 I_P)||\mathcal{N}(\mu_k,\sigma_k^2 I_P)) + \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}\log(\frac{\pi_k}{\gamma_{ik}}).
\end{aligned} \tag{20}
$$

The first term of the ELBO calculates the difference between the reconstructed and original graph adjacency matrices. The second term accounts for the reconstruction error between the document-term matrices of the input and output. The third term considers the Kullback-Leibler divergence (denoted by $D_{KL}(\cdot)$) between the approximate posterior distribution of node i, obtained with the encoder, and the prior distribution of component k. Finally, the last term takes into consideration the clustering probabilities.

On the one hand, an *explicit* optimization of the ELBO with respect to the parameters $\gamma_{ik}, \pi_k, \mu_k$ and σ_k can be performed via Proposition 1.

Proposition 1. *The following variational updates can be obtained (all proofs are given in Appendix A)*

$$
\hat{\gamma}_{ik} = \frac{\pi_k \mathrm{e}^{-D_{KL}^{ik}}}{\sum_{l=1}^{K} \pi_l \mathrm{e}^{-D_{KL}^{il}}}, \tag{21}
$$

where $D_{KL}^{ik} = \frac{1}{2}\left\{\log \frac{(\sigma_k^2)^P}{(\tilde{\sigma}_\phi^2(\tilde{A})_i)^P} - P + \frac{\tilde{\sigma}_\phi^2(\tilde{A})_i}{\sigma_k^2} + \frac{1}{\sigma_k^2}||\mu_k - \tilde{\mu}_\phi(\tilde{A})_i||^2\right\}$.

Then

$$
\hat{\pi}_k = \sum_{i=1}^{N} \gamma_{ik}/N, \tag{22}
$$

$$
\hat{\mu}_k = \sum_{i=1}^{N} \tilde{\mu}_\phi(\tilde{A})_i \gamma_{ik} / \sum_{i=1}^{N} \gamma_{ik}, \tag{23}
$$

and

$$\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \gamma_{ik}(P\sigma_\phi^2(\tilde{A})_i + ||\mu_k - \tilde{\mu}_\phi(\tilde{A})_i||^2)}{P\sum_{i=1}^{N}\gamma_{ik}}. \tag{24}$$

On the other hand, the *implicit* optimization of the encoder parameter ϕ, the two neural networks parameters ι and ν, the graph decoder parameter τ as well as the document decoder parameters α and ρ, is automatically performed via stochastic gradient descent. In this work, the implicit optimization is implemented using the Adam optimizer (Kingma and Ba, 2014). We also point out that during the estimation, a reparameterization trick as in Kingma and Welling (2014a) is used for the terms $\mathbb{E}_{q(Z|A,W)}[\log p(A|\iota, \tau, \eta_i, \eta_j)]$ and $\mathbb{E}_{q(Z|A,W)}[\log p(W_{im}|\nu, \delta_i, \alpha, \rho)]$.

5. Numerical experiments

This section aims at testing the effectiveness of GETM, including a DS-GCN encoder, a latent position-based graph decoder and a ETM-based document decoder, on three types of synthetic networks, and to demonstrate the validity of the estimation algorithm proposed in the previous section.

5.1. Simulation setup

We first generate three different types of synthetic networks, each of which having three groups of nodes. The connections between nodes are obtained by a stochastic block model (SBM, Nowicki and Snijders, 2001). Each node in the network is then associated with a document, where each word is picked at random from three articles from BBC news, denoted by D_1, D_2 and D_3, respectively. The first document discusses the birth of Princess Charlotte. The second text is about black holes in astrophysics. The last article focuses on UK politics. Three scenarios are described in detail as follows, with an illustration provided in Figure 3.

- Scenario A simulates a graph according to SBM, where edges between two nodes are drawn from independent Bernoulli distributions

$$A_{ij}|(Z_{ik}Z_{jl} = 1) \sim \mathcal{B}(\Pi_{kl}),$$

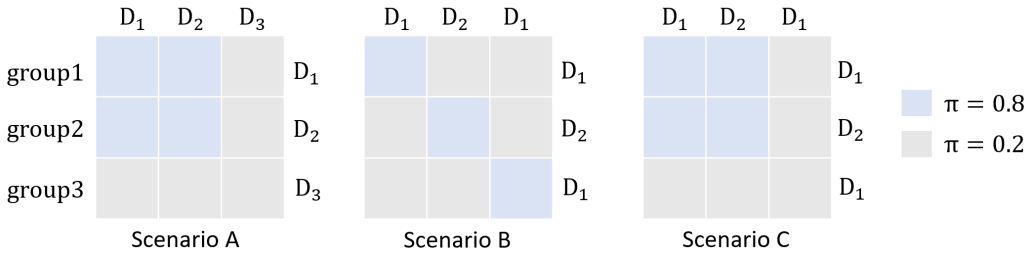

Figure 3: Three scenarios to simulate synthetic networks.

with Z_{ik} being 1 if node i is in cluster k, 0 otherwise. The connection probabilities are defined as

$$\Pi = \begin{pmatrix} 0.8 & 0.8 & 0.2 \\ 0.8 & 0.8 & 0.2 \\ 0.2 & 0.2 & 0.2 \end{pmatrix},$$

where each entry Π_{kl} denotes the probability that a node in group k and a node in group l connect. Thus, the third group has low inter and intra connection probabilities and differs from the other two groups, which are more likely to connect. Additionally, we assign a document to each node and the words used in the three document groups are randomly extracted from text D_1, D_2 and D_3, respectively (see Figure 3). If we solely analyze the network topology structure, the first two groups are likely to be clustered together. However, if we look at the text information that each node in this network is concerned with, the actual number of clusters is three.

- In Scenario B, networks are also simulated according to SBM, with connection probabilities given by

$$\Pi = \begin{pmatrix} 0.8 & 0.2 & 0.2 \\ 0.2 & 0.8 & 0.2 \\ 0.2 & 0.2 & 0.8 \end{pmatrix},$$

corresponding to a clear community structure. It is simple to detect three clusters by just considering the network topology. However, group 1 and group 3 adopt words from D_1, whereas words in group 2 are extracted from D_2 (see Figure 3). Therefore, there are two clusters if we only take the document information into account.

- Scenario C has the same connection probabilities as in scenario A and the texts assignments are the same as in scenario B. In this situation, focusing solely on the graph topology or the textual data would lead to the discovery of two clusters, whereas the actual group numbers is three when the two types of information are considered simultaneously.

5.2. Benchmark study

We now aim at benchmarking the clustering performance of GETM with the following competitors in the three simulated scenarios.

- ETM (Dieng et al., 2020) is a document generative model that combines traditional topic models with word embeddings and is intended only for textual data.

- SBM (Nowicki and Snijders, 2001) is a widely used generative model in network analysis for clustering of nodes and is designed for graph data only.

- VGAE (Kipf and Welling, 2016b) encodes the adjacency matrix and the node/document feature matrix by a GCN, and adopts an inner-product decoder for graph reconstruction.

- AM-GCN (Wang et al., 2020) is a GCN-based method which performs graph convolution both accounting for the network topology and the node features space.

For each scenario, we randomly generated 15 networks with 900 nodes, 3 clusters, and calculated the averaged adjusted rand index (ARI, Hubert and Arabie, 1985) for node clustering comparisons. The results are reported in Table 1.

As can be observed, ETM showed excellent results in Scenario A due to the fact that each group is associated with a distinct topic. However, in Scenario B and C, only two types of documents are considered, ETM has a poor ARI since it cannot exploit the connectivity between nodes/documents. SBM only achieved great results in Scenario B, which contains three groups in the graph topological structure. However, SBM failed to detect three clusters in Scenario A and C since it cannot exploit the textual interaction. VGAE treats text data as node attributes to perform clustering. Due to the constraints in fusing the graph topology and the word semantics in GCN,

Table 1: Experimental clustering results on 3 simulated scenarios.

	Scenario A	Scenario B	Scenario C
ETM	**1.000**±0.00	0.552±0.02	0.540±0.02
SBM	0.630±0.05	**1.000**±0.00	0.608±0.05
VGAE	0.459±0.00	0.773±0.07	0.460±0.00
AM-GCN	**1.000**±0.00	0.892±0.03	0.961±0.02
GETM	**0.990**±0.01	**1.000**±0.00	**0.998**±0.00

VGAE shows the worst performance in all situations. AM-GCN constructs a k-nearest neighbor graph based on the node feature matrix to capture the underlying document semantics. As a result, AM-GCN was able to identify three clusters in different scenarios. Nevertheless, its performance in Scenario B was unsatisfactory in comparison to Scenario A and C. Finally, GETM consistently displayed strong clustering performance with high ARI values in all situations, which demonstrates its capability in both representation learning and node clustering.

5.3. A more detailed example

We now focus on Scenario C, which is the primary emphasis of this work. In the initial configuration, both the graph topology structure and topic subjects are divided into two categories, making it difficult to find the actual number of clusters.

We first run GETM on a synthetic network with 900 nodes, generated according to Scenario C, and then visualize the learned joint embeddings, graph embeddings and document embeddings, respectively. After training for 600 epochs, we plot the reconstruction loss for graph and text, the total loss (negative ELBO), and display the evolution of the ARI during training (Figure 4). All types of loss have converged and the ARI value (equal to 1.0) highlights the clustering capability of GETM as well.

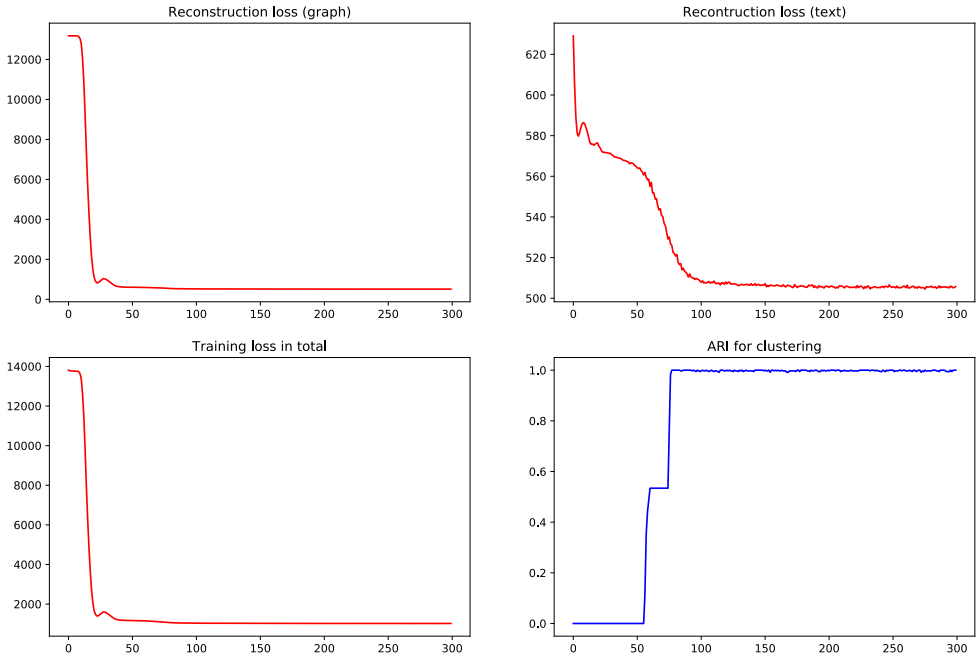

Figure 4: The training loss for graph and texts reconstruction, the overall loss and the evolution of ARI during training.

Then, we visualize the latent embeddings learned by GETM. First, Figure 5 displays the combined embeddings Z using PCA. Three groups with distinct colors can be distinguished by GETM, which demonstrates the ability of DS-GCN to exploit the network topology and the textual information. Then, regarding the graph topological embeddings η in Figure 6, two groups with different topologies are well preserved. As we can see, there are two clusters with a high probability of intra connections (positions are relatively closed) and one cluster is separated, which is coherent with our initial configuration. In addition, two latent topics are successfully detected based on the document embeddings θ, as shown in Figure 7. One document cluster is far away, whereas the other two clusters are very close, recovering the simulation setup in Scenario C.

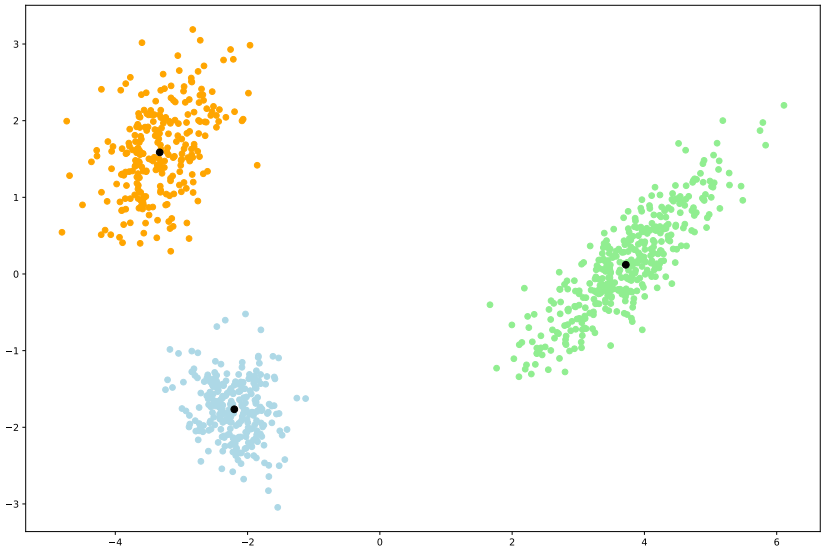

Figure 5: PCA visualisation of the joint embedding Z ($P = 128$) in Scenario C. Each cluster is represented by a distinct color. Three black points are the estimated cluster centers μ_k.

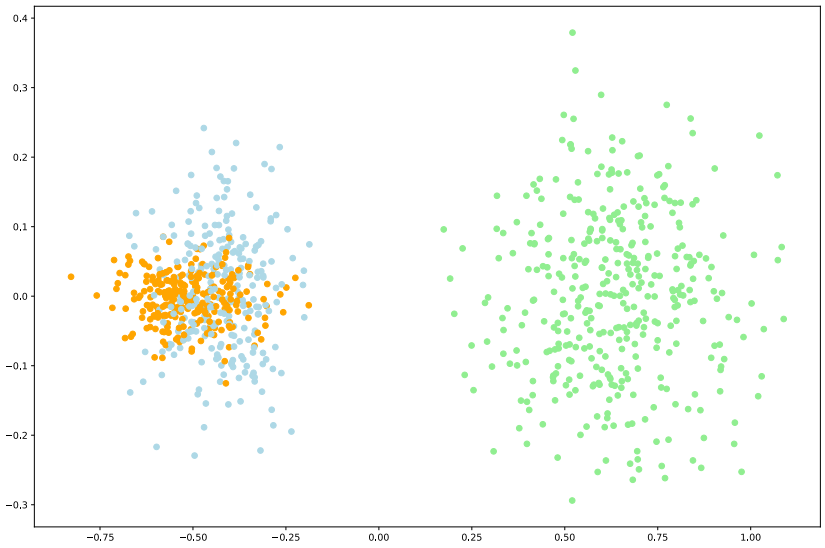

Figure 6: PCA visualisation of η ($F = 128$) in Scenario C. The green group is far away, whereas the orange and blue groups are relatively close, showing two different topologies.

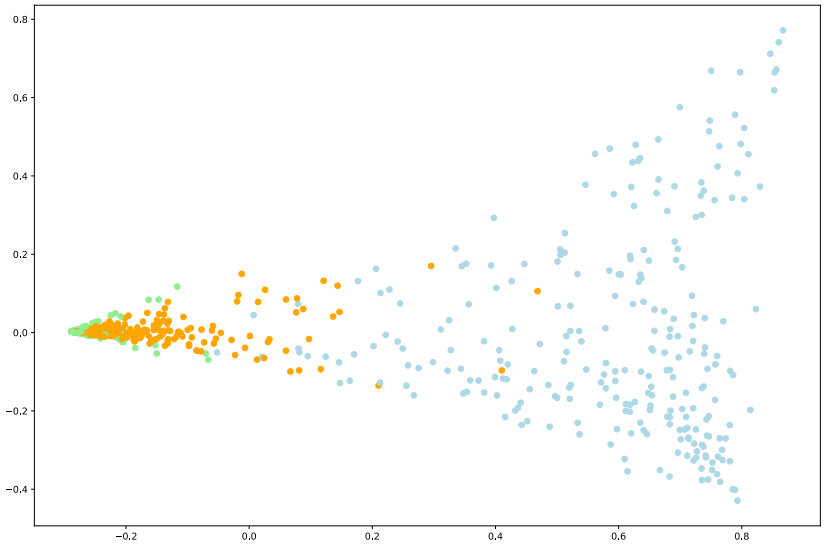

Figure 7: PCA visualisation of θ ($T = 16$) in Scenario C. The blue group is far away, whereas the green and orange groups are relatively close, showing two different topics.

Table 2: Top-10 words obtained from two topics.

Topic 1	”princess”	”charlotte”	”birth”	”queen”	”duchess”
	”duke”	”cambridge”	”granddaughter”	”great”	”palace”
Topic 2	”hole”	”black”	”see”	”gravity”	”light”
	”one”	”event”	”horizon”	”around”	”disc”

Moreover, we also illustrate the top-10 words selected from two topics in Table 2. As we can see, words associated with the first topic are related to the birth of Princess Charlotte, and vocables in the second topic are about black holes in astrophysics, which recovers the simulation setup for texts.

To conclude, all previous results indicate that GETM is capable of learning representations and performing node clustering in heterogeneous information networks.

5.4. Model selection

A key element of an unsupervised learning technique such as GETM is to be able to automatically determine the number of clusters (K). We highlight here the ability of our methodology to auto-penalize the ELBO for selecting the number of groups appropriately, which is made possible by the self-regularization ability of variational auto-encoders, also reported in Kingma et al. (2016); Dai et al. (2017).

Number of clusters. Letting the number of clusters vary from 2 to 7, Figure 8 illustrates how the training loss (negative ELBO) can be used to estimate the number of clusters. In this experiment, for each value of the number of clusters, we generated five synthetic networks from Scenario C and trained GETM with the latent, joint embedding dimension $P = 16$, the graph embedding dimension $F = 16$ as well, and the dimension of document embedding equal to the number of topics $T = 2$. It can be seen that when $K = 3$, the training loss (negative ELBO) is minimal, thus recovering the actual value of K for the simulation setting.

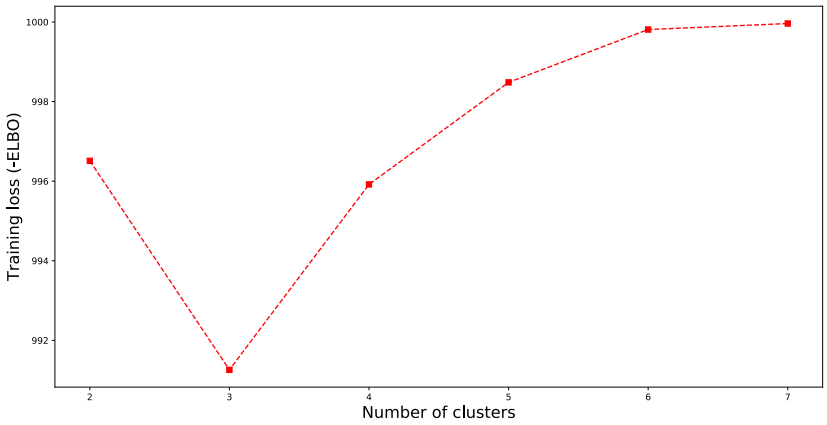

Figure 8: Averaged training loss (negative ELBO) with different number of clusters on 30 synthetic networks in Scenario C. GETM was able to estimate $K = 3$ by displaying a clear minimum of the negative ELBO.

Dimension for latent topics. We further evaluated the model selection ability with different number of latent topics. In this experiment, we generated five synthetic networks from Scenario C and trained GETM with the latent, joint embedding dimension P of 128, the graph embedding dimension F of 128,

and the document embedding dimension T of $\{2, 16, 128\}$, respectively. In Table 3, we examine the training loss (negative ELBO) for various number of clusters $K \in \{1, \cdots, 6\}$. The capacity of GETM to choose the proper cluster numbers is highlighted by the fact that the minimal training loss (negative ELBO) is discovered when $K = 3$, with varying dimensions of latent topics.

Table 3: Averaged training loss (negative ELBO) with different number of clusters and latent topic dimension in Scenario C.

	K=2	K=3	K=4	K=5	K=6
T=2	1048.11	**1028.74**	1038.37	1037.69	1037.17
T=16	1040.65	**1026.58**	1027.36	1027.66	1028.89
T=128	1027.88	**1026.60**	1028.44	1027.27	1028.14

6. Application on real-world network

In this section, GETM is fitted on a text-rich citation network Cora-enrich[1] as an illustration of its practical use. The original Cora[2] dataset contains 2,708 scientific publications classified in seven categories: *case based*, *genetic algorithms*, *neural networks*, *probabilistic methods*, *reinforcement learning*, *rule learning* and *theory*. It consists of 5,429 links and 1,433 vocabulary. Each publication is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from a dictionary. Recently, Ganguly and Pudi (2017) enriched the text information by collecting the titles, abstracts and all sentences from a paper containing citations, which leads to 25,955 vocables in Cora-enrich network. This dataset shares the same papers, categories and citation relationships with Cora.

Most related works (Pan et al., 2018; Mehta et al., 2019; Jin et al., 2021; Yu et al., 2021) assume that the number of clusters is equal to the number of classes used in supervised node classification tasks, whereas we argue that the class labels (thematic categories) might not be in a one-to-one relation with the detected clusters in unsupervised node clustering. Instead, an appropriate cluster number should be obtained through model selection.

[1] `http://zhang18f.myweb.cs.uwindsor.ca/datasets/`

[2] `https://relational.fit.cvut.cz/dataset/CORA`

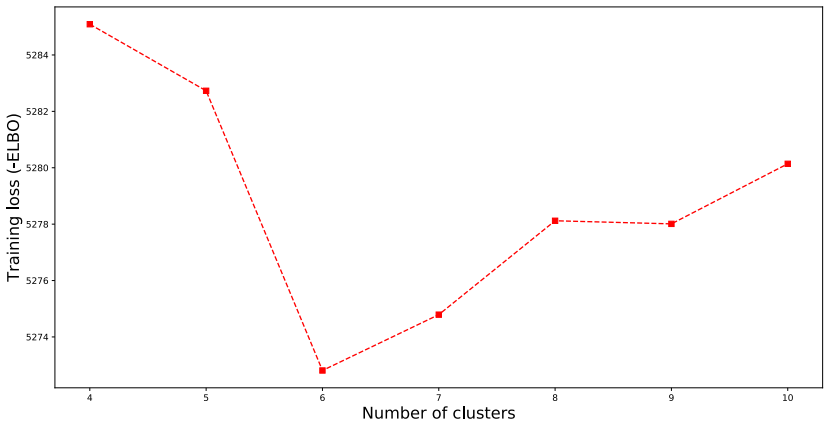

Figure 9: Training loss (negative ELBO) with different number of clusters on Cora-enrich. GETM estimates $K = 6$ by clearly showing the minimum of the negative ELBO.

6.1. Model selection

As the model selection ability of GETM is demonstrated in Section 5.4, GETM is also fitted to the Cora-enrich network for different numbers of clusters, ranging between 4 and 10, with fixed dimensions $(P, F, T = 128)$ for three latent spaces. The evolution of the training loss (negative ELBO) with various cluster numbers is shown in Figure 9. The reported result is the lowest value obtained after running GETM 10 times for each cluster number. Finally, the estimated number of clusters is $K = 6$ by displaying a clear minimum of the negative ELBO.

Confusion matrix. We also plot the confusion matrix between six estimated cluster partitions and seven thematic categories to investigate the fusion or dispersion between multiple classes, as shown in Figure 10. As we can see, the majority of publications on reinforcement learning (T3), rule learning (T5) and genetic algorithms (T6) are extracted into clusters $C3$, $C5$ and $C1$, respectively. The two clusters $C2$ and $C6$ constitute the primary division between articles about neural networks (T7). Theoretical publications (T4) are mainly grouped into $C4$ and $C5$ clusters. Case based papers (T2) are separated into clusters $C3$ and $C5$. The articles that address probabilistic approaches (T1) are divided into the clusters $C2$, $C3$, $C4$, and $C5$.

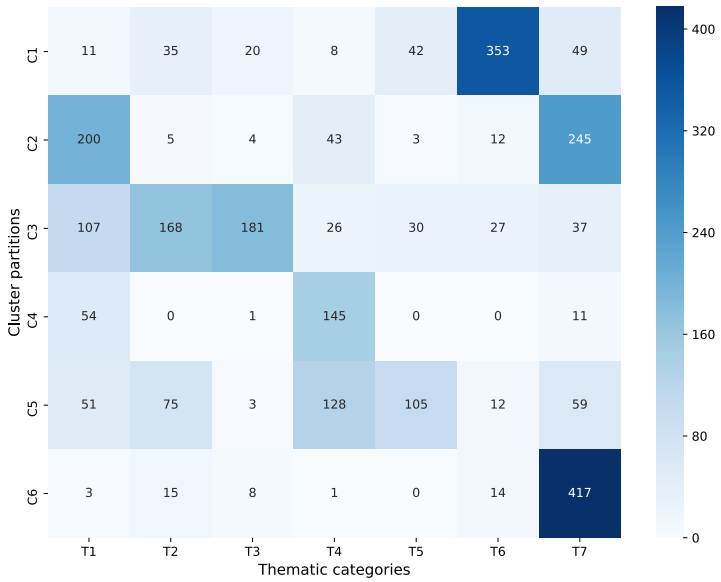

Figure 10: Confusion matrix between six estimated clusters and seven thematic categories (T1: *probabilistic methods*, T2: *case based*, T3: *reinforcement learning*, T4: *theory*, T5: *rule learning*, T6: *genetic algorithms*, T7: *neural networks*).

Based on these results, we stress that the number of clusters cannot be determined solely based on the number of the thematic classes. Conversely, when selecting the number of clusters via model selection, we are able to discover interesting new similarities between the nodes of a graph.

6.2. Visualisation and analysis

We further visualize the latent embeddings discovered by GETM in Figure 11. The standard network visualization tool *gplot* within the *sna* library in R is used. Six clusters are represented in distinct colors with a layout using a variant of Fruchterman and Reingold force-directed placement algorithm by default. It can be seen that, GETM was able to detect different communities on Cora-enrich, where nodes from various clusters are gathered. Moreover, Figure 12 shows the paper distributions in six groups when seven thematic categories are taken into account.

- The Grp 1 in red, in particular, collects the majority of publications on genetic algorithms. It makes sense because genetic techniques is a specialized topic of research that is distinct from other themes. Since

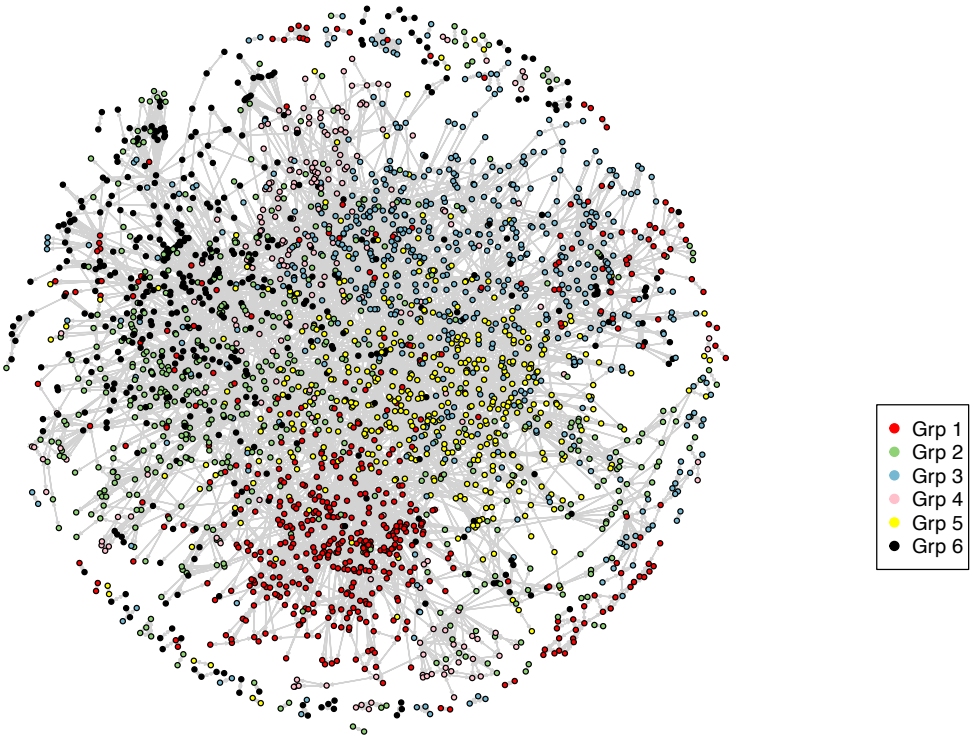

Figure 11: Visualization of Cora-enrich. Six clusters are represented in distinct colors.

many neural network models are constructed using probabilistic principles and it is common for them to strengthen proposed models based on theory background.

- Grp 2 in green extracts many publications from neural networks, probabilistic approaches, and some theoretical articles. This is expected given that theory papers or probabilistic methods can serve as the foundation for neural network models.

- Because many reinforcement learning models are created for a particular case study and may rely on some probabilistic basis, the Grp 3 (blue) includes practically almost all of the reinforcement learning publications, a significant number of case-based articles, and some probabilistic methods.

- Similarly, a number of probabilistic or neural network-based models are

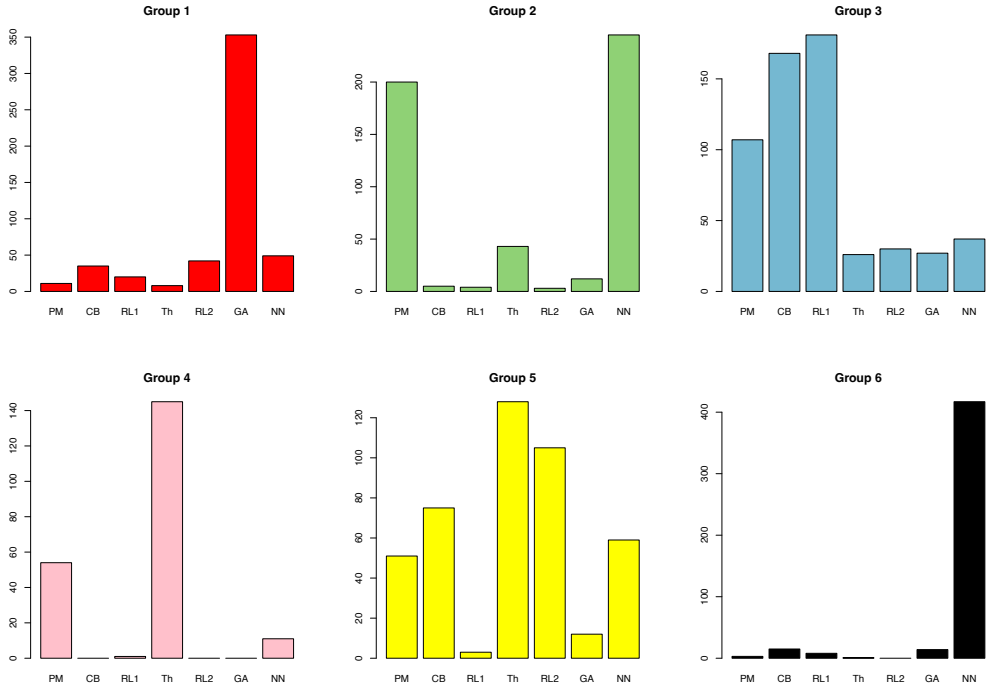

Figure 12: Partitions taking into account thematics in each cluster on Cora-enrich (PM: *probabilistic methods*, CB: *case based*, RL1: *reinforcement learning*, Th: *theory*, RL2: *rule learning*, GA: *genetic algorithms*, NN: *neural networks*).

gathered specifically in the Grp 4 (pink) together with the majority of theory articles.

- Since rule learning can be used in a variety of domains, Grp 5 in yellow contains the majority of rule learning publications along with many other types of papers.

- Finally, the Grp 6 in black captures the other neural network publications that are solely informatics-related.

Additionally, we also visualize the graph topological embedding and analyze the link connections between nodes. Figure 13 illustrates the latent embedding η of graph topology. As we can see, there is a small community of nodes in Grp 1 (red) that are exclusively interconnected, these could represent publications on specific genetic algorithms. The relationships for

the other groups are not very evident from this figure, so we go further into them by computing the community memberships quantitatively for each cluster (number of links within each cluster). The results are reported in Table 4. It is clear that all of the diagonal values are quite large, highlighting the community structure: nodes are more likely to interact within their communities.

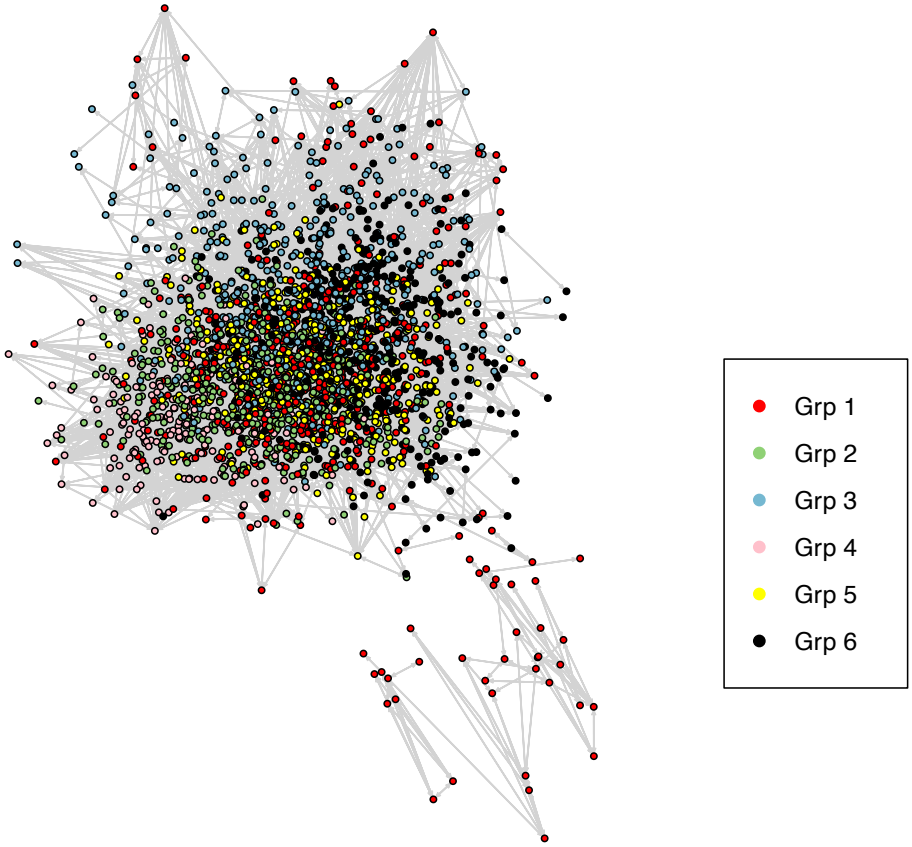

Figure 13: Visualization of the graph topology embedding η.

Table 4: Number of links within each cluster.

Grp \ Grp	1	2	3	4	5	6
1	**1794**	207	393	74	111	218
2	207	**1032**	79	106	117	119
3	393	79	**1480**	7	114	51
4	74	106	7	**568**	46	0
5	111	117	114	46	**1312**	29
6	218	119	51	0	29	**1030**

We close this section emphasizing once more that, in unsupervised problems, we cannot determine the number of clusters solely based on the number of the classes that are used in supervised tasks. These classes are simply assigned based on the article thematics, without taking into account the citation relationships. Conversely, when selecting the number of clusters via model selection (that VAEs seem to perform intrinsically), we are able to discover interesting new similarities between the nodes of a graph by combining the graph topology structure with latent thematics in textual information.

7. Conclusion

In this work, we propose the document similarity-based graph convolutional network (DS-GCN) to account for both the network topology structure as well as word and topic semantics from the textual information. Then, two different decoding networks are introduced to reconstruct both the graph adjacency matrix and the document-term matrix. In addition, an end-to-end node clustering is performed using the graph embedded topic model (GETM) by estimating the posterior probabilities for cluster memberships. Numerical experiments on simulated scenarios demonstrate that GETM is capable of learning representations in heterogeneous information network. Moreover, the performance of GETM in node clustering is highlighted by the benchmark study with other competitors based on three different simulations. We further conduct a model selection to test the ability of our methodology to auto-penalize the ELBO for choosing the number of clusters appropriately. Finally, an unsupervised network analysis is conducted on the Cora-enrich

network to emphasize the model selection ability and the interest to discover hidden patterns behind the thematics.

Here we considered networks that include mixed-type information, but each node and link have the same kind of properties and relationships. For instance, each node is associated with a textual document and links are all undirected and unweighted. For future works, we could deal with more complex real-world networks, including nodes with various characteristics.

References

Dai, B., Wang, Y., Aston, J., Hua, G., Wipf, D., 2017. Hidden talents of the variational autoencoder. arXiv preprint arXiv:1706.05148 .

Dieng, A.B., Ruiz, F.J., Blei, D.M., 2020. Topic modeling in embedding spaces. Transactions of the Association for Computational Linguistics 8, 439–453.

Ganguly, S., Pudi, V., 2017. Paper2vec: Combining graph and text information for scientific paper representation, in: European conference on information retrieval, Springer. pp. 383–395.

Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of classification 2, 193–218.

Jin, D., Song, X., Yu, Z., Liu, Z., Zhang, H., Cheng, Z., Han, J., 2021. Bite-gcn: A new gcn architecture via bidirectional convolution of topology and features on text-rich networks, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 157–165.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Kingma, D.P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., Welling, M., 2016. Improved variational inference with inverse autoregressive flow. Advances in neural information processing systems 29, 4743–4751.

Kingma, D.P., Welling, M., 2014a. Auto-encoding variational bayes. International Conference on Learning Representations (ICLR) .

Kingma, D.P., Welling, M., 2014b. Stochastic gradient vb and the variational auto-encoder, in: Second International Conference on Learning Representations, ICLR, p. 121.

Kipf, T.N., Welling, M., 2016a. Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations (ICLR-17).

Kipf, T.N., Welling, M., 2016b. Variational graph auto-encoders, in: NeurIPS Workshop on Bayesian Deep Learning (NeurIPS-16 BDL).

Mehta, N., Duke, L.C., Rai, P., 2019. Stochastic blockmodels meet graph neural networks, in: International Conference on Machine Learning, PMLR. pp. 4466–4474.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. International Conference on Learning Representations .

Nowicki, K., Snijders, T.A.B., 2001. Estimation and prediction for stochastic blockstructures. Journal of the American statistical association 96, 1077–1087.

Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., Zhang, C., 2018. Adversarially regularized graph autoencoder for graph embedding, in: International Joint Conference on Artificial Intelligence (IJCAI-18), pp. 2609–2615.

Wang, C., Pan, S., Hu, R., Long, G., Jiang, J., Zhang, C., 2019. Attributed graph clustering: A deep attentional embedding approach. arXiv preprint arXiv:1906.06532 .

Wang, X., Zhu, M., Bo, D., Cui, P., Shi, C., Pei, J., 2020. Am-gcn: Adaptive multi-channel graph convolutional networks, in: Proceedings of the 26th ACM SIGKDD International conference on knowledge discovery & data mining, pp. 1243–1253.

Yu, Z., Jin, D., Liu, Z., He, D., Wang, X., Tong, H., Han, J., 2021. As-gcn: Adaptive semantic architecture of graph convolutional networks for text-rich networks, in: 2021 IEEE International Conference on Data Mining (ICDM), IEEE. pp. 837–846.

Appendix A. Derivatives of the ELBO

Proof. Detailed derivations are obtained as follows. In order to take into account the equality constraint $\sum_{k=1}^{K} \gamma_{ik} = 1, \forall k$, we introduce the Lagrange multipliers λ_i

$$\tilde{\mathcal{L}} := \mathcal{L} - \sum_{i=1}^{N} \lambda_i \left(\sum_{k=1}^{K} \gamma_{ik} - 1 \right),$$

then, we derive $\tilde{\mathcal{L}}$ according to γ_{ik} and set the derivative equal to zero

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \gamma_{ik}} = \log \pi_k - \log \gamma_{ik} - \frac{\gamma_{ik}}{\gamma_{ik}} - D_{KL}^{ik} - \lambda_i = 0,$$

thus, we have

$$\begin{aligned} \log \gamma_{ik} &= \log \pi_k - 1 - D_{KL}^{ik} - \lambda_i, \\ \gamma_{ik} &= \mathrm{e}^{(\log \pi_k - 1 - D_{KL}^{ik} - \lambda_i)} = \frac{\mathrm{e}^{(\log \pi_k - D_{KL}^{ik})}}{\mathrm{e}^{(1+\lambda_i)}}. \end{aligned} \tag{A.1}$$

By using the constraint on $\sum_{k=1}^{K} \gamma_{ik}$, we get

$$\begin{aligned} \sum_{k=1}^{K} \gamma_{ik} &= \frac{\sum_{k=1}^{K} \mathrm{e}^{(\log \pi_k - D_{KL}^{ik})}}{\mathrm{e}^{(1+\lambda_i)}} = 1 \\ \log \sum_{k=1}^{K} \mathrm{e}^{(\log \pi_k - D_{KL}^{ik})} &= \log \mathrm{e}^{(1+\lambda_i)} \\ \lambda_i &= \log \sum_{k=1}^{K} \mathrm{e}^{(\log \pi_k - D_{KL}^{ik})} - 1. \end{aligned}$$

After putting λ_i into Eq. (A.1)

$$\gamma_{ik} = \frac{\mathrm{e}^{(\log \pi_k - D_{KL}^{ik})}}{\mathrm{e}^{(1+\log \sum_{k=1}^{K} \mathrm{e}^{(\log \pi_k - D_{KL}^{ik})} - 1)}} = \frac{\mathrm{e}^{(\log \pi_k - D_{KL}^{ik})}}{\sum_{k=1}^{K} \mathrm{e}^{(\log \pi_k - D_{KL}^{ik})}}.$$

Finally, we obtain

$$\hat{\gamma}_{ik} = \frac{\pi_k \mathrm{e}^{-D_{KL}^{ik}}}{\sum_{l=1}^{K} \pi_l \mathrm{e}^{-D_{KL}^{il}}}. \tag{A.2}$$

Similarly, since $\sum_{k=1}^{K} \pi_k = 1, \forall k$, we introduce another Lagrange multiplier, say ζ

$$\tilde{\mathcal{L}} := \mathcal{L} - \zeta \left(\sum_{k=1}^{K} \pi_k - 1 \right),$$

then, we derive $\tilde{\mathcal{L}}$ according to π_k and impose the derivative equal to zero

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \pi_k} = \sum_{i=1}^{N} \frac{\gamma_{ik}}{\pi_k} - \zeta = 0,$$

next, we use the equality constraint to find the value of ζ

$$\begin{aligned} \sum_{k=1}^{K} \sum_{i=1}^{N} \gamma_{ik} &= \sum_{k=1}^{K} \pi_k \zeta \\ \zeta &= N, \end{aligned}$$

and finally, we have

$$\hat{\pi}_k = \sum_{i=1}^{N} \gamma_{ik} / N. \tag{A.3}$$

Last, we need to calculate the derivatives of the lower bound with respect to μ_k and σ_k^2. We start by deriving $\tilde{\mathcal{L}}$ according to μ_k

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \mu_k} = -\frac{1}{2} \sum_{i=1}^{N} \gamma_{ik} \left(\frac{1}{\sigma_k^2} (2\mu_k - 2\tilde{\mu}_\phi(\tilde{A})_i \right) = 0,$$

then, we obtain

$$\begin{aligned} \mu_k \sum_{i=1}^{N} \gamma_{ik} &= \sum_{i=1}^{N} \tilde{\mu}_\phi(\tilde{A})_i \gamma_{ik}, \\ \hat{\mu}_k &= \frac{\sum_{i=1}^{N} \tilde{\mu}_\phi(\tilde{A})_i \gamma_{ik}}{\sum_{i=1}^{N} \gamma_{ik}}, \end{aligned} \tag{A.4}$$

and finally for σ_k^2, we have

$$\begin{aligned} \frac{\partial \tilde{\mathcal{L}}}{\partial \sigma_k^2} = -\frac{1}{2} \sum_{i=1}^{N} \gamma_{ik} \left(\frac{P}{\sigma_k^2} - \frac{1}{\sigma_k^4} (P\sigma_\phi^2(\tilde{A})_i + ||\mu_k - \tilde{\mu}_\phi(\tilde{A})_i||^2) \right) &= 0 \\ P \sum_{i=1}^{N} \frac{\gamma_{ik}}{\sigma_k^2} &= \sum_{i=1}^{N} \frac{\gamma_{ik}}{\sigma_k^4} (P\sigma_\phi^2(\tilde{A})_i + ||\mu_k - \tilde{\mu}_\phi(\tilde{A})_i||^2) \\ P \sum_{i=1}^{N} \gamma_{ik} \sigma_k^2 &= \sum_{i=1}^{N} \gamma_{ik} (P\sigma_\phi^2(\tilde{A})_i + ||\mu_k - \tilde{\mu}_\phi(\tilde{A})_i||^2) \\ \hat{\sigma}_k^2 &= \frac{\sum_{i=1}^{N} \gamma_{ik} (P\sigma_\phi^2(\tilde{A})_i + ||\mu_k - \tilde{\mu}_\phi(\tilde{A})_i||^2)}{P \sum_{i=1}^{N} \gamma_{ik}}. \end{aligned} \tag{A.5}$$

□

Appendix B. Implementation details and computation time

In GETM, the DS-GCN encoder g_ϕ has 512 neurons in the first hidden layer and 128 neurons in the second hidden layer, respectively, equipped with a Relu activation for the first layer. The neural networks $h_\iota^{(G)}$ and $h_\nu^{(T)}$ are one-layer linear networks with 128 and 16 neurons, respectively. The graph decoder f_τ is a one-layer neural network, following with a sigmoid function, which maps the latent graph topology embeddings η into a reconstructed graph. The document decoder $\theta^\top \beta$ maps the topic and word embeddings in dimension $L = 300$ into a reconstructed document-term matrix.

Adam optimizer is used to update network weights. On the simulated networks, the learning rate for the graph part is $5e^{-3}$, and 0.02 for the document part. On the Cora-enrich network, the learning rate for two parts are $5e^{-3}$, and 0.01, respectively.

The computation time on the simulated network with 900 nodes, 558 (Scenario B and C) or 721 words (Scenario A) is about 0.03s/epoch, for a total of 9.18s for 300 epochs on a GeForce RTX 2070 GPU. On the same GPU, training on the citation network Cora-enrich with 2,708 nodes and 25,955 words takes about 0.10s/epoch and a total of 28.50s for 300 epochs.

For more details, our code is available in: `https://github.com/ldggggg/GraphETM`. In this paper, SBM is implemented by *sparsebm* package in Python, ETM, VGAE and AM-GCN are conducted using the available Python code in github: `https://github.com/lffloyd/embedded-topic-model`, `https://github.com/DaehanKim/vgae_pytorch` and `https://github.com/zhumeiqiBUPT/AM-GCN`, respectively.