Variational Graph Embedding and Clustering with Laplacian Eigenmaps

Zitai Chen[1,2] , **Chuan Chen**[1,2*] , **Zong Zhang**[1] , **Zibin Zheng**[1,2] and **Qingsong Zou**[1,3]
[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[2]Guangdong Key Laboratory for Big Data Analysis and Simulation of Public Opinion, School of Communication and Design, Sun Yat-sen University, Guangzhou, China
[3]Guangdong Province Key Laboratory of Computational Science, Sun Yat-sen University, Guangzhou, China
{chenzt25,zhangz7}@mail2.sysu.edu.cn, {chenchuan,zhzibin,mcszqs}@mail.sysu.edu.cn

Abstract

As a fundamental machine learning problem, graph clustering has facilitated various real-world applications, and tremendous efforts had been devoted to it in the past few decades. However, most of the existing methods like spectral clustering suffer from the sparsity, scalability, robustness and handling high dimensional raw information in clustering. To address this issue, we propose a deep probabilistic model, called Variational Graph Embedding and Clustering with Laplacian Eigenmaps (VGECLE), which learns node embeddings and assigns node clusters simultaneously. It represents each node as a Gaussian distribution to disentangle the true embedding position and the uncertainty from the graph. With a Mixture of Gaussian (MoG) prior, VGECLE is capable of learning an interpretable clustering by the variational inference and generative process. In order to learn the pairwise relationships better, we propose a Teacher-Student mechanism encouraging node to learn a better Gaussian from its instant neighbors in the stochastic gradient descent (SGD) training fashion. By optimizing the graph embedding and the graph clustering problem as a whole, our model can fully take the advantages in their correlation. To our best knowledge, we are the first to tackle graph clustering in a deep probabilistic viewpoint. We perform extensive experiments on both synthetic and real-world networks to corroborate the effectiveness and efficiency of the proposed framework.

1 Introduction

Graphs are natural expressions to characterize the complex interactions between entities, such as social networks, citation networks, and gene interactions. Revealing the underlying structure is of significance to understanding these data. To this end, graph clustering is proposed to identify groups of nodes having a higher probability of being connected than to members of other groups. The graph clustering methods have shown its potential in many real-world applications, such as global air transportation network analysis and protein interaction analysis.

As a fundamental unsupervised machine learning problem, graph clustering, also called community detection, has attracted considerable research attention in the past few decades. Traditional algorithms seek to find the cluster structure based on specific criteria, e.g., modularity, normalized cut, permanence, and conductance. Among these methods, nonnegative matrix factorization (NMF) [Kuang *et al.*, ; Li *et al.*, 2019] and eigenvalue decomposition (EVD) are widely adopted and various approaches have been derived from them. They first map a network to embeddings in a low-dimensional latent space and then identify the structure. However, these types of embedding are linear [Yang *et al.*, 2016] which is not only in sharp contrast to the complex relationship among nodes but also limits the embedding capacity of the model. What's more, they represent nodes with deterministic embeddings ignoring the uncertainty of the embeddings which is inherent in the relationship. For example, nodes connecting to multiple clusters, also called hubs, would be confronted with cluster contradiction between their neighboring nodes. Such discrepancy should be considered in the uncertainty of its embedding and clustering. More importantly, the pairwise relationship is the building block of the graph data. Preserving such property in the embedding would benefit the clustering task.

On the other hand, deep learning models in numerous machine learning tasks have achieved state-of-the-art performance resulted from learning effective representations in a non-linear way. As a result, some recent works try to learn a more powerful embedding by the non-linear mappings. Instead of learning the mapping in the spectral clustering, GraphEncoder [Tian *et al.*, 2014] first introduces a sparse autoencoder to encoder the similarity matrix, and then does the k-means in embedding space to learn the clusters. Similarly, [Sun *et al.*, 2017; Ye *et al.*, 2018] proposes a nonnegative symmetric encoder-decoder approach to preserve the nonnegative property of embeddings like NMF. Deep nonlinear reconstruction (DNR) [Yang *et al.*, 2016] extends the traditional modularity based clustering by reconstructing the modularity matrix with deep autoencoder, which introduces non-linear mapping in graph clustering. The better performance of these deep learning model in graph clustering demonstrates the potential of non-linear mappings. However, all of these ap-

[*] Corresponding Author

proaches could not learn the uncertainty in the graph and well preserve the relationships in the embedding space. More importantly, the processing of indicating clusters is not inherent in the learning model, which leads to a loss of information.

To adequately capture the uncertainty and learn the implicit relationships in embedding space, the integration of probabilistic graphical models and deep clustering models is taking into consideration. Probabilistic graphical models are widely adopted to model the user-item relationships in the recommendation and generate samples in computer vision. By modeling each sample as a Gaussian distribution, probabilistic models can not only describe the sample's noise by dividing the representation into mean and covariance (i.e., the true position and the uncertainty), but also capture the implicit relationship between samples well. Very recently, [Jiang *et al.*, 2017; Dilokthanakul *et al.*, 2016] and [Hsu *et al.*, 2019] explore image generation and text topic modeling by combining the Gaussian mixture model (GMM) with a variational autoencoder (VAE). They propose these generative clustering models to generate highly realistic samples from a latent cluster. And the disentanglement of covariance is to preserve the diversity in generating samples. Despite the state-of-the-art performance they gain, they are feature-based approaches dealing with samples in Euclidean space and assume i.i.d. inputs. There is not the case in graph data in which the non-i.i.d. nature arises from the complex interactions between the nodes. And directly applying these methods to the graph data in non-Euclidean space will inevitably impact the performance ignoring the pairwise characteristic of the graph. Since the structure and influence of neighborhood vary widely by the node, preserving this information from the graph space to embedding space make a better performance on clustering task possible.

To address the above challenges, we propose a neural variational model which can compress the node representation as a Gaussian distribution preserving the pairwise relationship and cluster the nodes in their generative process simultaneously. The model, called variational graph embedding and clustering with Laplacian Eigenmaps (VGECLE), is formulated as a generative model based on the variational autoencoder (VAE) framework with a Mixture-of-Gaussian prior and a Teacher-Student (T-S) like regularization. In order to model the pairwise relationship in embedding, under the Stochastic Gradient Variational Bayes (SGVB) estimator, this T-S regularization term forces the node (student) to learn a distribution closer to its neighbors' one (teacher) with respect to the similarity. By learning the embedding from each other at different train batch, we show that it is also the Laplacian Eigenmaps (LE) [Belkin and Niyogi, 2003] to some extent. Specifically, our contributions can be summarized as follows:

- We propose a variational autoencoder for graph embedding and clustering, representing each node in the graph as a Gaussian distribution to disentangle the uncertainty and the true position of embedding.
- We propose a Teacher-Student mechanism in the stochastic gradient descent training process, which preserves the pairwise relationship and makes full use of the graph information.
- Extensive experiment on real-world datasets has shown that VGECLE can significantly outperform the state-of-the-art models.

2 Related Work

In this section, we briefly review the representative works in graph clustering and the graph embedding technique. The related works on probabilistic modeling are also included.

Since graph data are ubiquitous and almost feature-based data can be described by the similarity, a great deal of effort has been devoted to graph clustering over the past few decades [Fortunato, 2010]. However, there is no consensus on the formalization of the graph clustering and a variety of criteria are proposed to evaluate the quality of a network partition, such as modularity [Newman, 2006], normalized/ratio cut [Shi and Malik, 2000], permanence [Chakraborty *et al.*, 2014] and conductance [Leskovec *et al.*, 2010]. Spectral Clustering (SC) [von Luxburg, 2007] is one of the most widely adopted approach in graph clustering and various spectral-based approaches have been developed for graph clustering [Craddock *et al.*, 2012; Bühler and Hein, 2009]. Readers can refer to [Fortunato and Hric, 2016; Javed *et al.*, 2018] for more traditional methods and shallow models. For the deep graph clustering, GraphEncoder [Tian *et al.*, 2014] first investigated the connection between spectral clustering and autoencoder in terms of reconstructing the normalized graph similarity matrix. And it runs k-means on the graph embeddings learned by a sparse autoencoder (SAE) to partition the graph. SAE tries to compress the information of the node neighbors or a row of the adjacency matrix into an embedding and reconstruct the original vector. [Yang *et al.*, 2016] tried to encode the modularity matrix instead of the adjacency matrix to do clustering. [Sun *et al.*, 2017] proposed a nonnegative encoder-decoder architecture using the same nonnegative matrix to compress and reconstruct the neighborhood and cluster nodes with the nonnegative embeddings. However, all these graph clustering methods only learn a vector for each node entangled with uncertainty and restricts the performance in the clustering task.

Graph embedding aims to compress the node representations to a low-dimensional space meanwhile preserve some specific properties of the graph, such as the proximity and node degree. Typical network embedding methods include DeepWalk [Perozzi *et al.*, 2014], Node2vec [Grover and Leskovec, 2016] and SDNE [Wang *et al.*, 2016]. DeepWalk learns embeddings by exploiting the co-occurrence of nodes in a random walk. SDNE learns embeddings by combining GraphEncoder with a Laplacian Eigenmaps term to preserve the pairwise relationships. Very recently, Graph2Gauss [Bojchevski and Günnemann, 2018] and DVNE [Zhu *et al.*, 2018] attempt to use Gaussian distributions to represent a node to integrate uncertainty and position to embeddings.

Although graph embedding offers an effective way to obtain node representations, it still left the clustering problem unsolved. Consequently, we have to conduct clustering on the obtained embedding which does not correspond to the cluster structure of the graph. In this paper, we directly learn the clusters and the representation is the by-product, without requiring a clustering post-processing step.

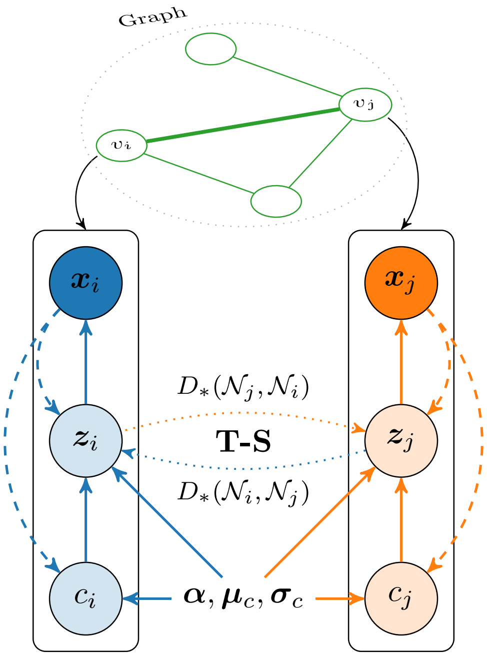

Graph
v_i
v_j
$\boldsymbol{x}_i$
$\boldsymbol{x}_j$
$D_*(\mathcal{N}_j, \mathcal{N}_i)$
$\boldsymbol{z}_i$
T-S
$\boldsymbol{z}_j$
$D_*(\mathcal{N}_i, \mathcal{N}_j)$
c_i
$\boldsymbol{\alpha}, \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c$
c_j

Figure 1: The graphical illustration of the proposed framework VGECLE. The graph is colored in green, and below is the graphical model of the variables. The generative process of observation $\boldsymbol{x}_i$ is in solid arrow, while the inference of latent variables z_i and c_i are in dashed arrow. The middle panel with $\boldsymbol{\alpha}, \{\boldsymbol{\mu}_c, \boldsymbol{\sigma}_{\boldsymbol{c}}\}_{c=1}^K$ is the global latent learnable parameters of MoG. For the link between v_i and v_j, the Teacher-Student (T-S) mechanism is also shown in dotted arrow: when training the Gaussian embedding for v_i colored in blue, $\mathcal{N}_i$ is encouraged to learn from its neighbor $\mathcal{N}_j$ colored in orange. And $\mathcal{N}_j$ also learn from $\mathcal{N}_i$ when training node v_j.

3 Proposed Framework

In this section, we describe the proposed variational graph embedding and clustering with Laplacian Eigenmap (VGECLE) model. Before diving into the details of the proposed method, we first summarize the notations used in this paper. Then we present the details of VGECLE. The graphical model of the VGECLE is shown in Figure 1.

Throughout this paper, we denote scalars, vectors and matrices by lowercase letters, bold lowercase letters and bold uppercase letters respectively. A graph is denoted as $G = (V, E)$, where $V = \{v_1, v_2, \cdots, v_n\}$ is a set of n nodes and $E = \{e_{ij}\}_{i,j=1}^n$ is a set of edges. Each edge e_{ij} is associated with a weight $s_{ij} \geq 0$. Only when v_i and v_j are not linked by an edge, $s_{ij} = 0$. $S = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_n\}$ provides the information of the neighborhood structure of each node. The neighborhood of node v_i is denoted as $N(v_i)$. We aim to partition the graph into K parts, where K is given. We denote $\mathcal{N}_i = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\boldsymbol{I})$ as a low-dimensional Gaussian embedding for node v_i, where $\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i \in \mathbb{R}^D, D \ll n$ and $\boldsymbol{I}$ is the identity matrix.

Although we focus on the clustering task instead of the generative task in this model, for better understanding of the model, we follow the literature in introducing the variants of VAE with the generative process first and then the inference.

3.1 Generative Model with Mixture of Gaussian Prior

Two latent variables c and $\boldsymbol{z}$ are introduced to model the MoG prior in VAE. Specifically, c is a K-way categorical discrete variable, named latent mixture-component indicator, and $\boldsymbol{z}$ is a D-dimensional continuous variable, named latent sample. Given an observed node v_i and neighborhood structure $\boldsymbol{x} = \boldsymbol{s}_i \in \mathbb{R}^n$, we aim to reconstruct the neighborhood from the latent variables. The latent variable c first sampled from its prior $p(c)$, choosing the mixture-component/cluster. And then the latent sample $\boldsymbol{z}$ is sampled from a conditional distribution $p(\boldsymbol{z}|c)$, choosing a sample from the given component. Finally, the neighborhood is drawn from $p(\boldsymbol{x}|\boldsymbol{z})$. Assuming $p(\boldsymbol{x}|\boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z}, c)$, the joint probability can be factorized as:

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{z}, c) &= p(c)p(\boldsymbol{z}|c)p(\boldsymbol{x}|\boldsymbol{z}), \\ p(c) &= Cat(c|\boldsymbol{\alpha}), \\ p(\boldsymbol{z}|c) &= \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2\boldsymbol{I}), \\ p(\boldsymbol{x}|\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x^2\boldsymbol{I}), \end{aligned} \tag{1}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^K$, $\sum_{k=1}^K \alpha_k = 1$ and $Cat(\boldsymbol{\alpha})$ is a categorical distribution. $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2\boldsymbol{I})$ is a Gaussian component with learnable means $\boldsymbol{\mu}_c$ and variances $\boldsymbol{\sigma}_c^2$. As a result, the marginal distribution of latent variable $\boldsymbol{z}$ is a MoG:

$$p(\boldsymbol{z}) = \sum_{c=1}^K p(c)p(\boldsymbol{z}|c) = \sum_{c=1}^K \alpha_c \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2\boldsymbol{I}). \tag{2}$$

As a result, the latent variable c points out the cluster that generates the targeted neighborhood structure. In other words, the assignment of the node clustering is done according to the cluster information accompanied by itself. This good interpretability is inherent in the latent variables of the MoG model and passed to the nodes in the graph. The distribution of $\boldsymbol{x}$ is learned by a deep neural network (DNN) $f_\theta(\boldsymbol{z})$ which is parametrized by θ and $f_\theta(\boldsymbol{z}) = [\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x^2]$.

3.2 Variational Inference

Following the VAE framework [Kingma and Welling, 2013], to maximize the likelihood of the given data points is equal to maximizing its evidence lower bound (ELBO) with an inference model:

$$\begin{aligned} \mathcal{L}_{ELBO}(\boldsymbol{x}) :=& \log p(\boldsymbol{x}) - D_{KL}[q(\boldsymbol{z}, c|\boldsymbol{x})||p(\boldsymbol{z}, c|\boldsymbol{x})] \\ =& \mathbb{E}_{q(\boldsymbol{z},c|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}[q(\boldsymbol{z}, c|\boldsymbol{x})||p(\boldsymbol{z}, c)] \end{aligned} \tag{3}$$

where $q(\boldsymbol{z}, c|\boldsymbol{x})$ is the inference model for approximating the intractable posterior $p(\boldsymbol{z}, c|\boldsymbol{x})$. We assume the mean-field variational family $q(\boldsymbol{z}, c|\boldsymbol{x})$ can be factorized as $q(\boldsymbol{z}, c|\boldsymbol{x}) = q(\boldsymbol{z}|\boldsymbol{x})q(c|\boldsymbol{x})$. The approximated posterior is a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z^2\boldsymbol{I})$, where the mean $\boldsymbol{\mu}_z$ and variance $\boldsymbol{\sigma}_z$ are also learned by a DNN $g_\phi(\boldsymbol{x}) = [\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z^2]$. The Gaussian embedding of node v_i is the learned approximated posterior $N_i = q(\boldsymbol{z}|\boldsymbol{x}_i) = \mathcal{N}(g_\phi(\boldsymbol{x}_i))$. As for inferencing the mixture-component/clustering assignment, we can approximate it in a more elegant way instead of introducing another DNN. Since $q(c|\boldsymbol{x})$ is an approximation of the true posterior $p(c|\boldsymbol{x})$, we can achieve it by assuming $p(c|\boldsymbol{z}) = p(c|\boldsymbol{z}, \boldsymbol{x})$:

$$\begin{aligned} p(c|\boldsymbol{x}) &= \int_{\boldsymbol{z}} p(c|\boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} \\ &\approx \int_{\boldsymbol{z}} p(c|\boldsymbol{z}) q(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} := q(c|\boldsymbol{x}) \end{aligned} \tag{4}$$

where $p(c|\boldsymbol{z})$ is the membership-weights of $\boldsymbol{z}$ in MoG and has a closed-form solution:

$$\begin{aligned} p(c|\boldsymbol{z}) &= \frac{p(c)p(\boldsymbol{z}|c)}{\sum_{c'=1}^{K} p(c')p(\boldsymbol{z}|c')} \\ p(\boldsymbol{z}|c) &= \frac{\exp\left(-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu}_c)^T \boldsymbol{\sigma}_c^{-1/2} \boldsymbol{I} (\boldsymbol{z}-\boldsymbol{\mu}_c)\right)}{(2\pi)^{D/2} |\boldsymbol{\sigma}_c^2|^{1/2}}. \end{aligned} \tag{5}$$

To sum up, we compress the graph nodes to the latent cluster and the latent distribution by variational inference DNN $g_\phi(\cdot)$, and reconstruct the graph nodes from this latent information by generative DNN $f_\theta(\cdot)$. We also call $g_\phi(\cdot)$ as an *Encoder* transforming the neighborhood structure $\boldsymbol{s}_i$ into a low-dimensional code $\mathcal{N}_i$, which is a Gaussian distribution. Accordingly, $f_\theta(\cdot)$ is the *Decoder* interpreting the latent code/distribution $\mathcal{N}_i$ to reconstruct the neighborhood structure $\boldsymbol{x} = \boldsymbol{s}_i$.

Similar to VAE, the ELBO can be optimized by using SGVB estimator and reparameterization trick [Kingma and Welling, 2013].

$$\begin{aligned} L_{ELBO}(\boldsymbol{x}_i) =& \mathbb{E}_{q(\boldsymbol{z},c|\boldsymbol{x}_i)}[\log p(\boldsymbol{x}_i|\boldsymbol{z})] - D_{KL}(q(c|\boldsymbol{x}_i)||p(c)) \\ & - \mathbb{E}_{q(c|\boldsymbol{x}_i)}[D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{x}_i)||p(\boldsymbol{z}|c)\right)] \\ =& \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\boldsymbol{x}_i|\boldsymbol{z}_{i,l}) - D_{KL}(q(c|\boldsymbol{x}_i)||p(c)) \\ & - \mathbb{E}_{q(c|\boldsymbol{x}_i)}[D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{x}_i)||p(\boldsymbol{z}|c)\right)], \end{aligned} \tag{6}$$

where L is the number of Monte Carlo samples in evaluation, $\boldsymbol{z}_{i,l} = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}_l$ and $\boldsymbol{\epsilon}_l \sim \mathcal{N}(0, \boldsymbol{I})$.

3.3 Teacher-Student Mechanism

To improve the clustering performance of Gaussian embeddings and learn the pairwise relationships in the graph, we propose a Teacher-Student mechanism which can effectively distill information from graph space to embedding space. Intuitively, nodes in the same cluster would be more likely to have stronger relationships than that in other clusters. It is natural to get nodes closer in the embedding space if they are linked by an edge. However, in a stochastic gradient descent optimization scenario, the goal of updating all the non-i.i.d. node embeddings at the same time is hard to reach. Therefore, from the point of view of a single node, a node ought to seek the guidance information from its neighbors to learn a better Gaussian embedding for itself. We recruit the neighbors' Gaussian embeddings as teachers to provide information for learning. Specifically, the loss function for node v_i to reach this goal is defined as follows:

$$\sum_{v_j \in N(v_i)} D_*(\mathcal{N}_i, \mathcal{N}_j) \tag{7}$$

where $D_*(\mathcal{N}_i, \mathcal{N}_j)$ measures the distance between two distributions and here we adopt KL divergence from $\mathcal{N}_j$ to $\mathcal{N}_i$ as the VAEs usually do. For the weighted graph, it is reasonable for student to pay more attentions to the teachers whom have a tighter connection to the student in its learning process.

$$L_{T-S}(\boldsymbol{x}_i) = \sum_{v_j \in N(v_i)} s_{ij} D_{KL}(\mathcal{N}_i, \mathcal{N}_j). \tag{8}$$

All in all, we formulate the objective function of VGECLE for a single node v_i as:

$$L(\boldsymbol{x}_i) = L_{ELBO}(\boldsymbol{x}_i) - \beta L_{T-S}(\boldsymbol{x}_i) \tag{9}$$

where β is for balancing the trade-off between reconstruction loss and the pairwise relationship.

Next, we will show the relationship between the Teacher-Student mechanism and the Laplacian Eigenmaps (LE) [Belkin and Niyogi, 2003]. In training with stochastic gradient descent, the gradient is back-propagate to optimize the loss of a single node. While training with gradient, the model will update the loss of the whole graph. In that case, the Teacher-Student term will be formulated as follows:

$$\sum_{i,j|e_{ij} \in E} s_{ij} \cdot D_{KL}(\mathcal{N}_i, \mathcal{N}_j). \tag{10}$$

While the objective function in LE adopts ℓ-2 Norm to measure the difference between vectors instead of distributions:

$$\sum_{i,j|e_{ij} \in E} s_{ij} \cdot ||\boldsymbol{y}_i - \boldsymbol{y}_j||^2. \tag{11}$$

However, it is unpractical for VGECLE to optimize the whole graph with gradient descent, especially when dealing with the large-scale graphs.

4 Experiments

In this section, we use four benchmark real-world datasets to demonstrate the effectiveness of VGECLE. We provide quantitative comparisons of VGECLE with other state-of-the-art clustering methods in two categories: shallow models and deep learning models. The experimental results show significant improvements with respect to the baselines.

4.1 Baseline Methods

To evaluate the performance of our proposed VGECLE, we compared it with the following five baseline methods. To validate the effectiveness of deep models in graph clustering, two shallow approaches are used for comparison, i.e., k-means and Spectral Clustering. VGECLE also compares to a deep graph clustering method GraphEncoder, a representative embedding method based on autoencoder SDNE, and a variational feature-based clustering method VaDE. The detailed descriptions of these methods are listed as follows.

- **k-means**: This algorithm aims to partition n observations into k clusters. It assigns each observation to the cluster with the minimal distance to the others. We run the k-means algorithm taking the neighborhood structure as representation.

Datasets	Nodes	Edges	Classes
Cora	2708	5429	7
BlogCatalog	5196	171743	6
Flickr1	7564	239365	9
Flickr2	80513	5899882	195

Table 1: Datasets statistics

- **Spectral Clustering** [von Luxburg, 2007]: This algorithm makes use of the spectrum of the similarity matrix of the data to perform dimensionality reduction. It performs k-means on the learned eigenvector-based solutions.
- **GraphEncoder** [Tian *et al.*, 2014]: This method learns a nonlinear embedding of the original graph by sparse autoencoder, and then runs the k-means algorithm on the embedding to obtain a clustering result.
- **SDNE** [Wang *et al.*, 2016]: SDNE try to preserve the first and second order proximity to the embeddings. It learns a point-vector for each node using a deep autoencoder with LE regularization. We run k-means on the deterministic embeddings to acquire the clustering assignments.
- **VaDE** [Jiang *et al.*, 2017]: VaDE is one of the representative feature-based variational clustering methods adopting a Mixture of Gaussian as the prior in VAE. We evaluate the effectiveness of the proposed Teacher-Student mechanism.

4.2 Datasets

To evaluate the effectiveness and efficiency of the proposed framework, we employ three networked datasets: Cora, BlogCatalog, and Flickr. All the networks are publicly available, and also undirected. The statistics of the datasets are summarized in Table 1. The detailed information is shown as follows:

- **Cora**: This network represents the citation relationships between scientific publications, which consists of 2708 scientific publications classified into one of seven classes.
- **BlogCatalog**: It is a blogger community. The network is formed according to the interaction between users. The labels represent the topic categories provided by the authors. Those users without a predefined category have been removed. There are 6 different categories.
- **Flickr**: It is an online platform where people can share photos. Photographers can follow each other and form a network. The labels represent the interest groups of the users. There are overall 9 different categories for Flickr1 and 195 categories for Flickr2.

4.3 Evaluation Metric

In our experiment, we perform the task of clustering. We use the unsupervised clustering accuracy (ACC) to measure the performance of VGECLE. This metric is widely used in unsupervised learning scenario.

Method	Cora	Blog	Flickr1	Flickr2
k-means	31.57	26.96	13.03	\
Spectral Clustering	32.23	27.13	14.37	\
GraphEncoder	32.90	27.68	17.14	15.23
SDNE + k-means	32.92	30.12	20.33	15.95
VaDE	33.22	33.34	26.41	25.41
VGECLE	**34.67**	**35.61**	**28.39**	**27.53**

Table 2: Clustering results (ACC%)

- **ACC**: For the node v_i, c_i is the clustering result from the algorithm, and l_i is the ground-truth label. Then the ACC is defined as:

$$ACC = \max_{m \in \mathcal{M}} \frac{\sum_{i=1}^{n} \mathbf{1}(l_i = m(c_i))}{n} \tag{12}$$

where n is the total number of nodes, and m is the optimal mapping function in mapping set $\mathcal{M}$ that can be computed by employing the KuhnMunkres algorithm. The higher the ACC value, the better the clustering performance.

4.4 Implementation

In the unsupervised clustering scenario, we are not capable of determining network structure by cross-validation on a validation set. And for a fair comparison, we use the same network architectures in all the deep learning models. So we set the network dimensions to $input - 500 - 100 - D$, where $input$ is the dimension of adjacency vector aka n. As mentioned in [Kingma and Salimans, 2016], VAE-based models suffer from the reconstructed problem. The reconstruction term in ELBO is too weak to play a part at the beginning of training. Thus, pretraining is necessary. We use the autoencoder to pretrain the network and initialize the parameters for the deep models. The learning rate for Cora, BlogCatalog and Flickr2 is 0.01 and decreases every 100 epochs with a decay rate of 0.9. And The learning rate for Flickr1 is 0.001 and decreases every 100 epochs with a decay rate of 0.9. For all baseline algorithms, we simply run them 10 times and obtain the average performance.

4.5 Experiment Results

The clustering results on four datasets are summarized in Table 2. For an overview, our proposed method achieves much better clustering performances than others on each dataset.

From the results, all the deep learning methods outperform the shallow models in all datasets. Especially, the difference between Spectral Cluster and GraphEncoder in experiment and methodology analysis show that deep structures can help to obtain better representation and improve the performance in clustering. Furthermore, the deep probabilistic models, i.e. VaDE and VGECLE, also give a better results in almost all comparisons, which demonstrates the advantage of disentangling the position and the uncertainty in learning the embeddings. Taking the embedding and clustering as an integral task in learning is also superior to the two-stage learning e.g.,

Inner	Cross	SNR	k-means	GraphEncoder	VaDE	VGECLE
250	0	2.0214	39.42	30.92	44.14	**45.70**
150	0	1.2352	28.32	29.80	40.44	**42.68**
0	0	0.6794	26.96	27.68	33.34	**35.61**
0	60	0.4540	25.84	26.42	31.09	**32.92**
0	100	0.2957	24.00	24.04	30.88	**32.24**

Table 3: Accuracy(%) under different SNRs by appending edges inner clusters and cross clusters

SDNE first, then k-means. Compared with VaDE, the improvement of VGECLE validates the effectiveness of the proposed Teacher-Student mechanism and the importance of the pairwise relationship in graph clustering.

Our proposed method compresses the node representation as a Gaussian distribution while preserving the proximity of the neighborhood. It can be seen that VGECLE achieves ACC of 34.67%, 35.61%, 28.39% and 27.53% on Cora, BlogCatalog, Flickr1, Flickr2 respectively.

4.6 Effectiveness Analysis

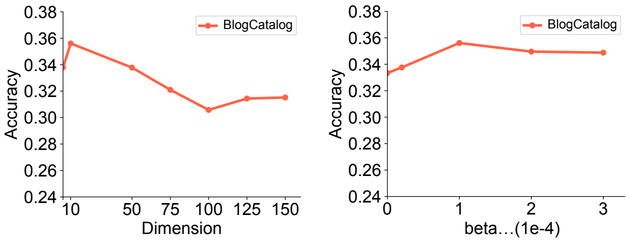
In order to demonstrate the effectiveness of our proposed VGECLE, we design an experiment running on the BlogCatalog dataset. Specifically, we conducted five sets of experiments on graphs under different signal-to-noise ratios. We add edges inner clusters and cross clusters to adjust the ratios. The result is presented in Table 3. From the results, we have the following observations and analysis.

The results show that our proposed method achieves better performances than others. It demonstrates that our method is able to learn the complicated distribution of the dataset better. Although VGECLE and VaDE both adopt a Mixture of Gaussian as the prior in VAE, VGECLE outperforms the VaDE by 2% on average. This result testifies the effectiveness of the proposed Teacher-Student mechanism. And the pairwise relationship plays an important role in the graph clustering. And then we can see that our proposed VGECLE outperforms the GraphEncoder(GE) by 8% on average. Without taking the graph uncertainty into account, GraphEncoder is inferior to both VGECLE and VaDE in the performance

4.7 Parameter Sensitivity

In this part, we investigate how the different dimensions D of the Gaussian embedding and the different hyper-parameter β values affect the performances. Specifically, we run our approach on the BlogCatalog dataset.

As shown in Figure 2(a), we can see that the dimension of the Gaussian embedding affects the performance in ACC. With the increase of D value from 5 to 10, the performance raises. However, when the number of dimensions continuously increases, the performance is in a declining trend and fluctuates within a range and less than 0.34. The reason is that when the D value is too low, it could not capture enough structure information of the graph. While the number of dimensions is too large, it may contain more noise, leading to poor performance. It is essential to determine the appropriate number of Gaussian embedding dimensions. Therefore, we set the dimension of the embeddings to 10 in all experiments.



(a) D

(b) β

Figure 2: The dimensions D of embedding and the value of β.

As for the parameter β, it is for balancing the trade-off between reconstruction loss and the pairwise relationship. Then we show how the value of β affects the clustering results in Figure 2(b) with fixed embedding dimension $D = 10$. When $\beta = 0$, the pairwise relationship in the graph is ignored and VGECLE is replaced by VaDE. The larger the β, the better the performance of VGECLE. Therefore, the pairwise relationship is important for VGECLE. We can see that the performance of $\beta = 0.0001$ is better than the others.

5 Conclusion

In this paper, we propose a novel probabilistic deep graph clustering framework VGECLE. Unlike the other methods in deep graph clustering, we disentangle the position and uncertainty of the graph node by representing each node as a Gaussian distribution. And we propose a Teacher-Student mechanism to preserve the pairwise relationship in the graph space to the Gaussian embeddings space. It offers an interpretable clustering assignment which is learned by the variational inference with a Mixture of Gaussian prior. We compared the clustering performance of VGECLE with baselines on 4 real-world datasets, and the experiment results show that VGECLE outperforms the other methods. Also, we conduct 5 sets of experiments on graphs under different signal-to-noise ratios to testify the effectiveness of VGECLE.

Acknowledgements

The work described in this paper was supported by the National Key R&D Program of China (2016YFB0200604), the National Natural Science Foundation of China (61722214, 11801595, 11571384), Natural Science Foundation of Guangdong (2018A030310076, 2017B030311001) and CCF Opening Project of Information System. Chuan Chen is corresponding author.

References

[Belkin and Niyogi, 2003] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[Bojchevski and Günnemann, 2018] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *ICLR*, 2018.

[Bühler and Hein, 2009] Thomas Bühler and Matthias Hein. Spectral clustering based on the graph p-laplacian. In *ICML*, pages 81–88. ACM, 2009.

[Chakraborty *et al.*, 2014] Tanmoy Chakraborty, Sriram Srinivasan, Niloy Ganguly, Animesh Mukherjee, and Sanjukta Bhowmick. On the permanence of vertices in network communities. In *KDD*, pages 1396–1405. ACM, 2014.

[Craddock *et al.*, 2012] R. Cameron Craddock, G.Andrew James, Paul E. Holtzheimer III, Xiaoping P. Hu, and Helen S. Mayberg. A whole brain fmri atlas generated via spatially constrained spectral clustering. *Human Brain Mapping*, 33(8):1914–1928, 2012.

[Dilokthanakul *et al.*, 2016] Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *CoRR*, abs/1611.02648, 2016.

[Fortunato and Hric, 2016] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1 – 44, 2016.

[Fortunato, 2010] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75 – 174, 2010.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864, 2016.

[Hsu *et al.*, 2019] Wei-Ning Hsu, Yu Zhang, Ron Weiss, Heiga Zen, Yonghui Wu, Yuan Cao, and Yuxuan Wang. Hierarchical generative modeling for controllable speech synthesis. In *ICLR*, 2019.

[Javed *et al.*, 2018] Muhammad Aqib Javed, Muhammad Shahzad Younis, Siddique Latif, Junaid Qadir, and Adeel Baig. Community detection in networks: A multidisciplinary review. *Journal of Network and Computer Applications*, 108:87 – 111, 2018.

[Jiang *et al.*, 2017] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. In *IJCAI*, pages 1965–1972. AAAI Press, 2017.

[Kingma and Salimans, 2016] Diederik P Kingma and Tim Salimans. Improving variational autoencoders with inverse autoregressive flow. In *NeurIPS*, 2016.

[Kingma and Welling, 2013] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[Kuang *et al.*,] Da Kuang, Chris Ding, and Haesun Park. *Symmetric Nonnegative Matrix Factorization for Graph Clustering*, pages 106–117.

[Leskovec *et al.*, 2010] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *WWW*, pages 631–640. ACM, 2010.

[Li *et al.*, 2019] Rui Li, Fanghua Ye, Shaoan Xie, Chuan Chen, and Zibin Zheng. Digging into it: Community detection via hidden attributes analysis. *Neurocomputing*, 331:97 – 107, 2019.

[Newman, 2006] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *KDD*, pages 701–710, 2014.

[Shi and Malik, 2000] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[Sun *et al.*, 2017] Bing-Jie Sun, Huawei Shen, Jinhua Gao, Wentao Ouyang, and Xueqi Cheng. A non-negative symmetric encoder-decoder approach for community detection. In *CIKM*, pages 597–606. ACM, 2017.

[Tian *et al.*, 2014] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *AAAI*, pages 1293–1299, 2014.

[von Luxburg, 2007] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[Wang *et al.*, 2016] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *KDD*, pages 1225–1234, 2016.

[Yang *et al.*, 2016] Liang Yang, Xiaochun Cao, Dongxiao He, Chuan Wang, Xiao Wang, and Weixiong Zhang. Modularity based community detection with deep learning. In *IJCAI*, pages 2252–2258. AAAI Press, 2016.

[Ye *et al.*, 2018] Fanghua Ye, Chuan Chen, and Zibin Zheng. Deep autoencoder-like nonnegative matrix factorization for community detection. In *CIKM*, CIKM '18, pages 1393–1402, New York, NY, USA, 2018. ACM.

[Zhu *et al.*, 2018] Dingyuan Zhu, Peng Cui, Daixin Wang, and Wenwu Zhu. Deep variational network embedding in wasserstein space. In *KDD*, pages 2827–2836. ACM, 2018.