Optimizing Variational Graph Autoencoder for Community Detection

Jun Jin Choong  
Department of Computer Science  
Tokyo Institute of Technology  
Tokyo, Japan  
junjin.choong@net.c.titech.ac.jp

Xin Liu  
National Institute of Advanced Industrial Science and Technology  
Tokyo, Japan  
xin.liu@aist.go.jp

Tsuyoshi Murata  
Department of Computer Science  
Tokyo Institute of Technology  
Tokyo, Japan  
murata@c.titech.ac.jp

Abstract—Variational Graph Autoencoders (VGAE) has recently been a popular framework of choice for learning representations on graphs. Its inception has allowed models to achieve state-of-the-art performances for challenging tasks such as link prediction, rating prediction and node clustering. However, a fundamental flaw exists in Variational Autoencoder (VAE) based approaches. Specifically, the objective function of VAE (reconstruction loss), deviates from its primary objective (i.e. clustering). In this paper, we attempt to address this issue by introducing two significant changes to Variational Graph Autoencoder for Community Detection (VGAECD). Firstly, we introduce a simplified graph convolution encoder to increase computational speed and reduce computational time. Secondly, a dual variational objective is introduced to encourage learning of the primary objective. The outcome is a faster converging model with competitive community detection performance.

Index Terms—community detection, graph neural network, variational autoencoder, network embedding

I. INTRODUCTION

Networks (graph) are often projected mappings from real-world observations. A network with nodes (vertices) and edges (link) are great simplification of complex patterns observed in real life, thus permitting studies of complex systems. For instance, the study of social interactions between two individuals can be represented in a form of social network and the number of contexts which their interaction took place can be formulated in terms of edge-weights [1]. Movies and their respective critic can be presented as a bipartite network with the edge-weight representing user-move ratings. The flexibility of networks and its vast literature on graph theory makes network science extremely appealing to researchers.

An area of interest with significant importance is community detection or widely known as graph clustering in graph theory [2]. Unlike supervised learning tasks, the definition of community detection is subject to debate. In graph representation learning, we consider structural and feature similarity to be grouped within the same community. Traditionally, network scientists has measured communities in form of partition quality, known as modularity [3]. A recovered community structure with high modularity denotes good partition quality. To this date, community detection algorithm has evolved from traditional algorithms to usage of complex deep learning algorithms like Graph Neural Network [4] (GNN). Similar to statistical base learning methods [5], GNN emphasize on aggregation of features from neighboring nodes, proving to be much more effective than traditional algorithms [6] in supervised and semi-supervised learning task where labels are scarce.

From a different perspective, GNN as representation learning [7, 8] on graphs; a general branch of machine learning that leverage on generalizing learning for downstream tasks such as link prediction, classification and clustering. Primarily driven by the performance gain in domain such as natural language processing (NLP), speech recognition and computer vision, representation learning has proven to be significantly important. However, deep learning algorithms are widely known to be a black-box learning algorithm. To rectify this issue, machine learning scientists have chosen to look at explainable artificial intelligence (XAI) algorithms. Causal inference [9] and Bayesian Deep Learning [10] are some examples of teaching machine learning algorithms to describe uncertainties and causal reasons.

Besides the aforementioned methods, generative models are equally appealing methods for explainability. For instance, Variational Autoencoder (VAE) [11] improves explainability by introducing uncertainty to an autoencoder. In graphs, Stochastic Blockmodel (SBM) [12, 13] is a popular approach to model network generation from a network science perspective. Depending on the structure of the stochastic matrix, we can observe different kinds of pattern whilst generating networks of similar structures. Recently, Kipf and Welling proposed Variational Graph Autoencoder (VGAE), which results in research variants such as VGAECD [15] and ARVGA [16].

Albeit powerful, VGAE based algorithm suffers from an optimization objective problem; deviating from its primary objective (community detection) in favor of the reconstruction of the input network. In section III-C, we discuss this problem in detail and propose a solution to rectify this issue. To summarize, the contributions are as follows,

- Simplify VGAECD using SGC [17] for computational speed up.
- Propose a dual optimization approach to alleviate the deviation of objective functions (community detection vs. network reconstruction)
II. PROBLEM DEFINITION

Formally, a network with $N$ number of nodes can be defined as $G = (V, E)$, where $V = \{v_1, \ldots, v_N\}$ denotes the set of nodes and $E = \{e_{ij}\}$ is a set of edges. Incidentally, each node may consist of features $X = \{x_1, \ldots, x_N\}$ where $x_i \in \mathbb{R}^D$ defines a vector of real-values associated with node $v_i$ with $D$-dimension. Vectorizing the notations, $A = \{a_1, \ldots, a_N\} \in \mathbb{R}^{N \times N}$ is the adjacency matrix of $G$. In this work, we consider the undirected and unweighted network $G$, such that $A_{ij} = 1$ if $e_{ij} \in E$ otherwise 0.

Given the network $G$, we aim to partition the nodes in $G$ into $K$ disjoint groups $\{c_1, c_2, \ldots, c_K\}$, such that nodes grouped within the same communities share similarities in terms of network structure or features. In graph representation learning, nodes in the same communities should share closer representation in an embedding space.

Additionally, we constrain our problem definition to generative model. Given a generative model, $p(\theta | X, A)$ infers the model parameters $\theta$ from the observed network $G$. Likewise, $G'$ can be generated from the same set of parameters such that $p(A' | \theta) = G'$. Under the model selection criterion, the model is a good model when $p(G' | \theta) \approx p(G)$ and satisfies the condition of having community structures i.e., $G'$ is not an Erdős-Rényi network.

III. METHODOLOGY

A. Variational Graph Autoencoder for Community Detection

Following prior work, Variational Graph Autoencoder for Community Detection (VGAEC) [15] generalizes the generation process of VAE [14] by introducing a mixture of gaussian in the generation process (decoder). Formally, the generative procedure of VGAEC can be express as

$$p(A | Z) = \prod_{i=1}^{N} \prod_{j=1}^{N} p(A_{ij} = 1 | z_i, z_j) = \phi(z_i^T z_j), \quad (1)$$

where $Z \in \mathbb{R}^{N \times F}$ is the latent representation of all nodes sampled from $\mathcal{N}(\cdot | \mu, \sigma^2 I)$ and $\phi(\cdot)$ is the logistic sigmoid function.

The generation process can be generalized to a mixture of gaussian by introducing a community assignment parameter $c$. Specifically, we would like calculate the joint probability distribution of $p(a, z, c)$ such that,

$$p(a, z, c) = p(a | z) p(z | c) p(c) \quad (2)$$

$$p(a | z) = \phi(z_i^T z; \mathcal{N}(\cdot | \mu_a, \sigma^2_a I))$$

$$p(c) = \text{Cat}(\cdot | \gamma)$$

$$p(z | c) = \mathcal{N}(\cdot | \mu_c, \sigma^2_c I)$$

For simplicity, we drop the explicit subscript to denote $z = z_i$ and $a = A_i$ (defined in section III-B). In (2), we obtain $p(c)$ from the categorical distribution parameterized by $\gamma$ with $K$ communities. The reconstruction probability, $p(a | z)$ is simply the inner product between latent representations $z$ parameterized by embeddings of the gaussian distribution.

B. Linearization of Encoder

VGAEC utilizes Graph Convolution layer (GCN) for its encoder to estimate parameters $\mu$ and $\sigma$. The variational distribution is formulated as,

$$q(Z | X, A) = \prod_{i=1}^{N} q(z_i | X, A) \quad (3)$$

$$q(z_i | X, A) = \mathcal{N}(z_i | \mu_i, \sigma^2 I).$$

The mean $\mu$ and $\sigma$ are obtained using a two-layer GCN define as,

$$\text{GCN}(X, A) = \hat{A} \phi(\hat{A} X W_0) W_1, \quad (4)$$

where $\hat{A}$ is obtained through a renormalization trick [6], $\hat{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ and $\{W_0, W_1\}$ are the trainable weight filters for each GCN layer. Albeit powerful, GCN is more computational expensive due to the non-linearity and increase in training parameters required. Wu et al. recently proposed a simplification of GCN by removing the non-linearity component, effectively linearizing GCN [17].

$$\text{SGC}(X, A) = \hat{A} \ldots \hat{A} X W^{(1)} W^{(2)} \ldots W^{(L)}. \quad (5)$$

Equation (5) describes SGC layer formally. In (5), the non-linear function $\phi$ is removed and features from $L$-hop neighbors are accumulated. Equation (5) further simplified to

$$\text{SGC}(X, A) = \hat{A}^L X W, \quad (6)$$

with $W = W^{(1)} W^{(2)} \ldots W^{(L)}$. Similar to the renormalization trick, $\hat{A}^L$ can be pre-computed before training. As we show in our experiments, this linearization drastically reduces training time for convergence and reduces computational complexity resulting in higher performance.

C. Dual Optimization

Fig. 1: The deviation problem exhibited when training VGAEC. The NMI drops approximately after 80 epochs and begins its reascend in a slow manner. In most cases, it deteriorate in favor of its secondary objective of minimizing the reconstruction loss.
In this paper, we consider optimizing the dual objective function,
\[ \mathcal{L} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{comm}}, \] (7) such that the reconstruction loss $\mathcal{L}_{\text{recon}}$ and the community’s quality loss, $\mathcal{L}_{\text{comm}}$ is minimized. It follows that, optimizing the loss is not trivial. Given enough capacity in an autoencoder, training with stochastic gradient descent would favor optimizing the reconstruction loss ($\mathcal{L}_{\text{recon}}$), eventually deviating from its true objective (i.e. community detection). As explained in [18, 19, 20], unsupervised deep learning algorithms has tendency to deviate from its true objectives and converges slowly if given insufficient guidance. We depict this problem exhibited in VGAECD in Figure 1. To rectify this issue, we employ a dual optimization process proposed by Greff et al. [21].

With that in mind, we optimize (7) using Neural Expectation-Maximization (N-EM) to exploit training VGAECD end-to-end with stochastic gradient descent. From (2), the objective function can be defined as,
\[ \log p(a) \geq \mathcal{L}_{\text{ELBO}}(a) = \mathbb{E}_{q(z,c|a)} \left[ \log \frac{p(a, z, c)}{q(z, c | a)} \right]. \] (8)

Reformulating it, we obtain
\[ \mathcal{L}_{\text{ELBO}}(a) = \mathbb{E}_{q(z,c|a)}[\log p(a | z)] - D_{KL}[q(z, c | a) \parallel p(z, c)]. \] (9)

In (9), by utilizing a dual optimization process, the reconstruction loss is first optimized, followed by the community loss. This process is then repeated until convergence. Similar to [14, 15], the reconstruction loss is minimized using a binary cross entropy loss and optimized using Adam [22]. The community loss is then minimized using an Expectation-Maximization (EM) algorithm [23] which guarantees a local optima. Given $\psi_{i,k} = f_{\phi}(\mu_k)$ parameterized by $\phi$ and $\theta = \mu_1, \ldots, \mu_K$, the loss of our variational distribution follows,
\[ \mathcal{L}_{\text{comm}}(\theta, \theta^{\text{old}}) = \sum_c p(c | a, \psi^{\text{old}}) \log p(a, c | \psi). \] (10)

To optimize (10), we use N-EM as the optimization algorithm. Firstly, we compute the expectation step, obtaining $\gamma$ the soft-assignment of each node, $v_i$,
\[ \gamma_{i,k} := p(c_i = 1 | z_i, \psi_{i,k}^{\text{old}}). \] (11)

Next, the maximization step follows,
\[ \theta^{\text{new}} = \theta^{\text{old}} + \eta \frac{\partial \mathcal{L}}{\partial \theta} \] (12) where,
\[ \frac{\partial \mathcal{L}}{\partial \mu_k} \propto \sum_{i=1}^{N} \gamma_{i,k} (\psi_{i,k} - x_i) \frac{\partial \psi_{i,k}}{\partial \mu_k} \] (13) with the learning rate parameter $\eta$. This process can be repeated $R$-times or until convergence. In practice, we found that $R \approx 5$ would suffice to achieve convergence. The full complete algorithm is described in Algorithm 1.

To explain this intuition, we refer to the theoretical formulation of VAE [22] and the bits-back argument [24]. Intuitively, in a high capacity VGAECD, gradient signals would favor $\mathcal{L}_{\text{recon}}$ over minimizing $\mathcal{L}_{\text{comm}}$. Consequently, the centroid’s embedding $\mu_k$ has higher tendency to converge slower or be ignored. Hence, to encourage encoding towards the centroid’s embedding, using a dual-optimization process allows gradient signals to be backpropagated to $\mu_k$ and $\sigma_k$. More specifically, turning $\mathcal{L}_{\text{comm}}$ to a variational EM optimization problem guarantees that the centroid embeddings has higher presence of encoding useful information.

Algorithm 1 VGAECD-OPT

Output: Community Assignment Probability $\gamma$ and Reconstructed Adjacency matrix $A$

$\pi \sim U(0, 1)$ for $l = 1, \ldots, L$ do
for $i = 1, \ldots, N$ do
\[ \mu_i = \text{GCN}_\mu(x_i, a_i) \]
\[ \sigma_i = \text{GCN}_\sigma(x_i, a_i) \]
Sample $z_i \sim \mathcal{N}(\mu_{i,k}, \text{diag}(\sigma_{i,k}^{2}))$
Obtain $\tilde{a}_i = \sigma(z_i, z_j)$
Compute loss, $\mathcal{L}_{\text{ELBO}}$ and backpropagate gradients.
end for
for $r = 1, \ldots, R$ do
Compute E-Step: $\gamma, \{\mu_c, \sigma_c\}$ \hspace{1cm} \(\triangleright\) From (11)
Compute M-Step: $\mu_c, \sigma_c, \{\gamma\}$ \hspace{1cm} \(\triangleright\) From (13)
Compute loss, $\mathcal{L}_{\text{comm}}$ and backpropagate gradients.
end for
end for

Extract community assignment $\arg \max_k \gamma$
Return $A = \{\tilde{a}_1, \ldots, \tilde{a}_N\}$

IV. EXPERIMENTS

In this section, we evaluate the improved version of [15], termed VGAECD-OPT. In subsequent subsections, we provide statistics of our datasets and experimental settings.

A. Dataset

We first measure the performance difference between VGAECD and VGAECD-OPT in terms of community structure recovery performed on two most common synthetic datasets, the Girvan-Newman (GN) [2, 25, 26] and Lancichinetti-Fortunato-Radicchi (LFR) benchmark graph [27]. We follow [15] by setting similar hyperparameters for each synthetic graph. Specifically, we use 128 nodes with an average degree $k = 16$ for GN benchmark graph. Number of nodes $N = 1000$, average degree $k = 15$, minimum and maximum number of nodes in a community $c_{\text{min}} = 30$ and...
\( c_{max} = 50 \) for the LFR benchmark graph. The generation follows the scale-free parameters settings of exponents \( \tau_1 = -2 \) and \( \tau_2 = -1 \) respectively. This result in a benchmark graph with about 25 communities on average.

To compare on real-world dataset, we chose to use Cora [28, 6, 15] in preference for comparable statistics evaluated in [15]. This dataset has 2708 nodes, 5429 edges, 7 clusters and 1433 features.

**B. Experimental Settings**

For baseline comparison, we have chose to compare with VGAE and VGAECD. For discriminative models, Spectral Clustering [29], Deepwalk [8] and node2vec [30] were chosen due to their popularity as baseline algorithms. The parameters for Deepwalk are dimensionality \( d = 128 \), walk length \( r = 10 \), number of random walks, \( l = 80 \), and window size \( k = 10 \) for all datasets. Node2vec on the other hand, comprises of parameters \((p = 1, q = 0.25)\) which follows [15]. For generative models, we chose SBM, SBM (D.C) [13], VGAE and VGAECD as baseline comparisons. The encoder of VGAE and VGAECD consists of a 2-layer GCN with configuration settings of \((32-16)\) D-dimension for synthetic networks and \((32-8)\) for empirical datasets. Since VGAECD-OPT consists of only a single layer \( W \) and for comparable performance, we use \((16)\) for synthetic networks and \((8)\) which are the deepest layer’s dimension in VGAE and VGAECD.

\[ c_{max} = 50 \]

for the LFR benchmark graph. The generation follows the scale-free parameters settings of exponents \( \tau_1 = -2 \) and \( \tau_2 = -1 \) respectively. This result in a benchmark graph with about 25 communities on average.

To compare on real-world dataset, we chose to use Cora [28, 6, 15] in preference for comparable statistics evaluated in [15]. This dataset has 2708 nodes, 5429 edges, 7 clusters and 1433 features.

**B. Experimental Settings**

For baseline comparison, we have chose to compare with VGAE and VGAECD. For discriminative models, Spectral Clustering [29], Deepwalk [8] and node2vec [30] were chosen due to their popularity as baseline algorithms. The parameters for Deepwalk are dimensionality \( d = 128 \), walk length \( r = 10 \), number of random walks, \( l = 80 \), and window size \( k = 10 \) for all datasets. Node2vec on the other hand, comprises of parameters \((p = 1, q = 0.25)\) which follows [15]. For generative models, we chose SBM, SBM (D.C) [13], VGAE and VGAECD as baseline comparisons. The encoder of VGAE and VGAECD consists of a 2-layer GCN with configuration settings of \((32-16)\) D-dimension for synthetic networks and \((32-8)\) for empirical datasets. Since VGAECD-OPT consists of only a single layer \( W \) and for comparable performance, we use \((16)\) for synthetic networks and \((8)\) which are the deepest layer’s dimension in VGAE and VGAECD.

**C. Evaluation Metrics**

Similar to classical community detection algorithms, we evaluate our method using Normalized Mutual Information (NMI) and Variation of Information to measure the quality of partitions given its ground truth. In the absence of ground truth, we use measures such as Modularity, Conductance and Triangle Participation Ratio.

- NMI and VI are based on information theory. NMI measures the ‘similarity’ between two community covers, while VI measures their ‘disimilarity’ in terms of uncertainty. Correspondingly, a higher NMI indicates a better match between both covers while VI indicates the opposite. Formally [31],

\[
\text{NMI}(C, C') = \frac{2I(C, C')}{H(C) + H(C')}
\]

and

\[
\text{VI}(C, C') = H(C) + H(C') - 2I(C, C'),
\]

where \( H(\cdot) \) is the entropy function, and \( I(C, C') = H(C) + H(C') - H(C, C') \) is the mutual information function.

- Modularity (Q) [32] measures the quality of a particular community structure when compared to a null (random)
model. Intuitively, intra-community links are expected to be stronger than inter-community links. Specifically,  

\[
Q = \frac{1}{2m} \sum_{ij} \left( A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j),
\]

where \(A_{ij} - \frac{k_i k_j}{2m}\) measures the actual edge connectivity versus the expectation at random and \(\delta(c_i, c_j)\) defines the Kronecker delta, where \(\delta(c_i, c_j) = 1\) when both node \(i\) and \(j\) belongs to the same community, and 0 otherwise. Essentially, \(Q\) approaches 1 when the partitions are considered good.

- **Conductance (CON)** [33, 34] measures the separability of a community across the fraction of outgoing local volume of links in the community, which is defined as,  

\[
\text{CON}(C) = \frac{\sum_{i \in C} \sum_{j \in \bar{C}} A_{ij}}{\min(a(C), a(C'))},
\]

where the nominator defines the total number of edges within community \(C\) and \(a(C) = \sum_{j \in \bar{C}} A_{ij}\) defines the volume of set \(C \subseteq V\). A better local separability of community is achieved when the overall conductance value is the smallest.

- **Triangle Participation Ratio (TPR)** [34] measures the fraction of triads within the community \(C\).  

\[
\text{TPR}(C) = \left| \left\{ v_i \in C, \{v_j, v_k\} : v_j, v_k \in C, (v_i, v_j), (v_i, v_k), (v_j, v_k) \in E \right\} \right| / |C|,
\]

where, \(E\) denotes the total number of edges in the graph \(G\). A larger TPR value indicates a denser community structure.

\[\text{V. RELATED WORK}\]

Traditionally, modularity maximization [32, 3] was widely used to identify communities in an unsupervised fashion. Louvain’s method [35], is an example of fast greedy algorithm for maximizing the modularity objective. On the other hand, DeepWalk [8] and node2vec [30] are popular algorithms for graph embedding. To generate a co-occurrence context, random walks are used in conjunction with negative sampling for large scale datasets. More recently, this line of algorithms can be generalized to a class of matrix factorization algorithm [36, 37]. Albeit powerful, the model has many hyperparameters to tune which are at times not very straightforward. More recently, models such as GraphEncoder [38], MGAE [39], VGAE [14] and ARVGA [16] leveraging on graph neural network framework for community detection. These methods are variants of GraphEncoders with different encoders and decoders. For instance, VGAE adopts GCN as the encoder.

\[\text{A. Conclusion & Future Work}\]

In this paper, we propose an optimization algorithm for VGAECD. Consequently, we show that using such optimization, we can improve the performance of VGAE’s community detection. For future work, we would like to explore capabilities of VGAECD-OPT’s. Concretely, we would like to explore its weaknesses and performance in other mainly unexplored areas, such as overlapping communities and weighted networks.

\[\text{ACKNOWLEDGMENT}\]

We would like to thank JST CREST (Grant Number JP-MJCR1687), JSPS Grant-in-Aid for Scientific Research(B) (Grant Number 17H01785) and JSPS Grant-in-Aid for Early-Career Scientists (Grant Number 19K20352)

\[\text{REFERENCES}\]


