A3Graph: Adversarial Attributed Autoencoder for Graph Representation Learning

Mingliang Hou
teemohold@outlook.com
School of Software, Dalian University of Technology

Lei Wang
leonard_wl@outlook.com
School of Software, Dalian University of Technology

Jiaying Liu
jiaying_liu@outlook.com
School of Software, Dalian University of Technology

Xiangjie Kong
xjkong@acm.org
College of Computer Science and Technology, Zhejiang University of Technology

Feng Xia
f.xia@acm.org
School of Engineering, IT and Physical Sciences, Federation University Australia

ABSTRACT

Recent years have witnessed a proliferation of graph representation techniques in social network analysis. Graph representation aims to map nodes in the graph into low-dimensional vector space while preserving as much information as possible. However, most existing methods ignore the robustness of learned latent vectors, which leads to inferior representation results due to sparse and noisy data in graphs. In this paper, we propose a novel framework, named A3Graph, which aims to improve the robustness and stability of graph representations. Specifically, we first construct an aggregation matrix by the combining positive point-wise mutual information matrix with the attribute matrix. Then, we enforce the autoencoder to reconstruct the aggregation matrix instead of the input attribute matrix. The enhancement autoencoder can incorporate structural and attributed information in a joint learning way to improve the noise-resilient during the learning process. Furthermore, an adversarial learning component is leveraged in our framework to impose a prior distribution on learned representations has been demonstrated as an effective mechanism in improving the robustness and stability in representation learning. Experimental studies on real-world datasets have demonstrated the effectiveness of the proposed A3Graph.

CCS CONCEPTS

• **Networks** → **Network algorithms**; • **Applied computing** → **Bioinformatics**.

SAC '21, March 22–26, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8104-8/21/03…$15.00
https://doi.org/10.1145/3412841.3442042

KEYWORDS

adversarial autoencoder, graph learning, attributed information, social network

ACM Reference Format:
Mingliang Hou, Lei Wang, Jiaying Liu, Xiangjie Kong, and Feng Xia. 2021. A3Graph: Adversarial Attributed Autoencoder for Graph Representation Learning. In *The 36th ACM/SIGAPP Symposium on Applied Computing (SAC '21), March 22–26, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3412841.3442042

1 INTRODUCTION

Social networks are an important class of networks for social computing, including a wide variety of media such as social websites (e.g., Facebook, Twitter) and citation networks (e.g., CiteSeer, Cora). Lots of latent knowledge can be obtained by analyzing social networks. For instance, by analyzing citation graphs, we can make a lot of interesting discoveries in research development such as the innovation flow [13], change of interests [28] and relationships between mentors and students [15]. However, the nonlinearity of graphs poses great challenges in graph analysis tasks such as node classification, node clustering, link prediction and graph visualization.

More recently, lots of approaches have been proposed to learn graph representations that map a graph into low-dimensional vector space while preserving original graph information such as structure information [22], node attribute information [5], and node label information [12]. Previous machine learning techniques always treat this problem as a preprocessing step. Feature extraction requires a lot of manual effort. In contrast, graph representation regards the preprocessing step as a learning task by using a data-driven approach to learn latent vectors that encode graph information automatically. The learned results (low-dimensional vectors) can be used as inputs for downstream graph analysis tasks.

Inspired by techniques of natural language processing, many graph representation methods have been proposed such as DeepWalk [22], node2vec [9], LINE [25] and GraRep [4]. They aim to preserve graph structural information, namely, pairwise proximity between nodes. Most methods adopt the

random walk as sampling strategy to model graph structural information. Experimental results on real-world datasets have demonstrated their superiority in common graph learning tasks.

Although existing methods have explored different ways to preserve the structural information, three limitations should be considered in designing a graph embedding framework. The first is how to capture the higher-order proximity of structural information in graph-structured data. DeepWalk and LINE capture the local consistency while node2vec explores partial global consistency. However, in practice, the global consistency relationship is indispensable. Users in social networks may be influenced by both local consistency relationships (e.g., friends, families) and global consistency relationships (e.g., community, nations). Thus, the global consistency should be considered in the learning process. The second limitation is how to combine topology information with attribute information. The node attribute information provides useful information than topological information in some specific scenarios. For example, nodes with same labels across networks tend to share similar attributes rather than similar topological structures [24]. Finally, the learned vectors may perform poorly in a noise environment. The robustness of the learned latent vectors should not be neglected.

To address above mentioned limitations, we propose a novel graph representation framework, termed as A3Graph. It incorporates node topology and attributed information in a joint learning way. Three major contributions are summarized as follows:

- We propose a novel graph representation framework, A3Graph, to learn robust node embeddings by considering topology and attribute information for a given graph.
- We design a structure-attribute aggregation matrix as the decoder objective to enforce autoencoder to learn local and global consistency between nodes. Moreover, a prior distribution is imposed to the learned latent vector as a regularizer though adversarial learning.
- The proposed framework, A3Graph, is extensively evaluated on three real datasets through node classification and node clustering tasks. Experiment results verify the effectiveness of A3Graph.

This paper is organized as follows. In Section 2, we review the related work including graph representation and generative adversarial networks. In Section 3, A3Graph will be introduced in detail. In Section 4, we elaborate on the experiment part. Section 5 concludes our work.

2 RELATED WORK

2.1 Graph Representation

Recently, graph representation has captured amounts of attentions [11]. These methods can be classified into three categorizes: random walk based methods, matrix factorization based methods, and deep neural network based methods.

Random walks based methods are commonly used in graph representation by exploring node structural information with randomly sampled paths. Inspired by word2vec [18], DeepWalk [22] treats nodes as words and random paths as sentences. It assumes that the vector of a node should be similar to the co-occurrence nodes in the sampled random walk paths [27]. Node2vec [9] extends a more flexible sample strategy in generating walk paths that can explore diverse structural neighboring nodes. LINE [25] preserves first-order and second-order structural information in large-scale graphs by incorporating alias table edge sampling in the random walk process.

Unlike random walk-based methods that use the sampling path as the input, the input of factorization based methods is an adjacent matrix. Then, matrix factorization techniques are used as feature learning extractors to obtain low-dimensional vectors. LLE [23] and LE [3] are two earlier methods factorizing a pre-processing matrix containing local structural information. GraRep [4] devotes to preserving the high-order global structural information by computing a k-step transformation probability matrix and utilizes SVD [1] to learn latent representations.

The goal of graph representation is to learn a mapping function from the original graph space to the latent representation space. Matrix factorization based methods approximate this mapping process linearly. Vast quantities of information are lost in the linear matrix factorization procedure due to nonlinearity of graphs. The deep neural network is an effective technique to model this nonlinear mapping process. SDNE [26] addresses graph representation by autoencoder network which can capture both the first-order and second-order structural information. SEANO [14] designs a representation framework with multi-layer perceptron (MLP) to learn node structural, attribute, and label information simultaneously. ANRL [29] proposes a neighbor enhancement autoencoder network with an attribute-aware skip-gram component to learn the structural and attribute information.

2.2 Generative Adversarial Networks

Generative Adversarial Networks [8] (GAN), inspired by Nash equilibrium [19], as a generative deep neural network model, consists of a discriminator and a generator. The discriminator aims at distinguishing the fake data from real data and the generator tries to generate data to fool the discriminator. This process can be seen as a minimax adversarial game.

Rich works such as GAN [8], Adversarial Autoencoder [17], DCGAN [7], WGAN [2] have proved that the adversarial learning mechanism is effective in obtaining stable states and reliable representations in image generation. Furthermore, several works like AIDW [6], and ARGA [21] have been proposed to learn more robust and stable graph representations.

Most GAN methods are designed for generating image data rather than graph data incapable of learning representations for graph-sturctured data. AIDW only considers the structural information. For ARGA, the robustness of learned representations is only guaranteed by the adversarial learning.

In this paper, different with previous method, we design a jointly learning framework to learn robust latent embeddings

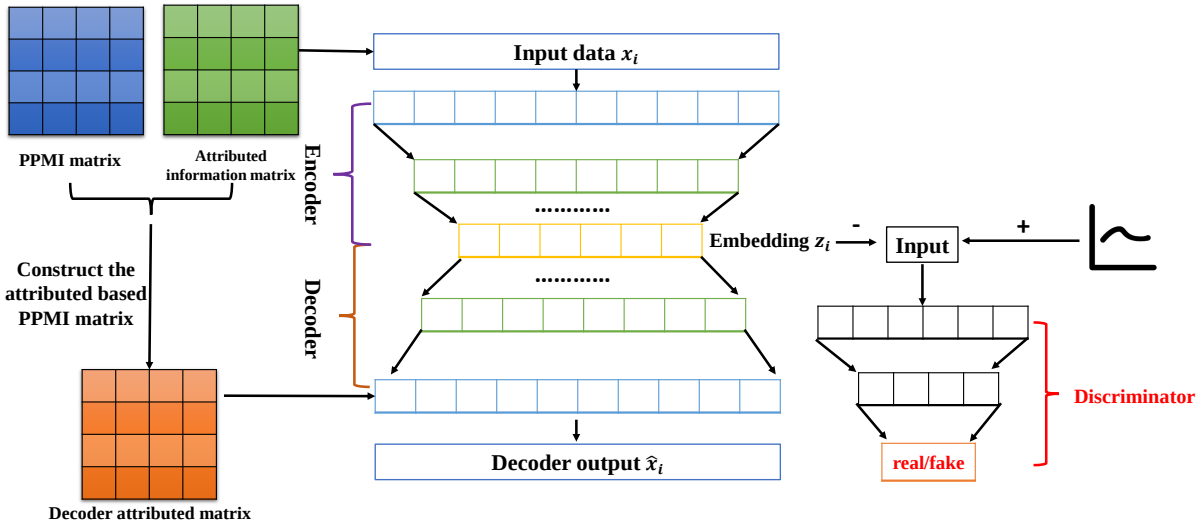


Figure 1: The framework of A3Graph.

by considering the local and global topology and attribute information with adversarial mechanism for graph-structured data. Compared with graph learning methods in related works, our A3Graph incorporates both structural and attributed information in a joint learning way. Besides, we combine structure-attribute aggregation reconstruction autoencoder with adversarial learning module to learn robust and stable graph representations simultaneously.

3 ADVERSARIAL ATTRIBUTED AUTOENCODER REPRESENTATION

3.1 Notations and Problem Definition

Definition 1. *An attributed graph is defined as* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$ *where* $\mathcal{V}$ *represents the node set and* $\mathcal{E}$ *represents the edge set in a graph.* $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ *is a node attributed matrix where* n *represents node number and* m *represents dimension of the attributed information.* x_i *is the attributed information vector of node* i.

Definition 2. *A Positive Pointwise Mutual Information (PPMI) matrix [4] of a given graph can be defined as:*

$$P_{ij} = \max\{\log(\frac{M_{i,j}}{\sum_k M_{k,j}}) - \log(\eta), 0\} \tag{1}$$

where $M = A + A^2 + ... + A^t$. A^t *is the* t*-step probability transformation matrix, denoting the probability for a transition between two nodes in* t *steps.* $A = D^{-1}S$ *where* D *is the diagonal matrix for a given graph with the adjacent matrix* S. η *is set to* $\frac{1}{n}$ *in this paper.*

Definition 3. *Given a graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$*, for each node* $i \in \mathcal{V}$*, we aim to learn a mapping function* $f : v_i \mapsto \mathbf{z_i} \in \mathbb{R}^d$ *where* $d \ll |\mathcal{V}|$*. The mapping function* f *devotes to capturing node structural information as well as attributed information.*

3.2 An Overview of the Framework

In order to learn topology and attribute information for a given graph in a regularized way, we propose an adversarial attributed autoencoder for graph representation learning (A3Graph). The framework, as shown in Figure 1, mainly consists of the following two module:

- Aggregation matrix enhance autoencoder. In order to learn topology and attribute information of each node, we employ autoencoder as the basic feature extractor. The decoder objective is replaced with an aggregation matrix to enforce the autoencoder to learn topology and attribute information by considering the local and global consistency simultaneously.
- Adversarial learning module. We design an adversarial learning module to enforce learned vectors to match a prior distribution. The discriminator discriminates whether the latent embedding vectors comes from the encoder or from the predefined distribution.

3.3 Aggregation matrix Enhancement Autoencoder Module

In order to capture graph local and global structural information incorporating with attributed information, we design a PPMI matrix based reconstruction autoencoder as the main part of A3Graph framework. The encoder part is similar to the conventional autoencoder which encodes node attributes to a low-dimensional latent space. However, the decoder part maps the encoder representations to aggregation matrix values. More specifically, for a given node v_i with attribute information $\boldsymbol{x_i} \in \mathbb{R}^{1 \times m}$, the encoder part learns the embedding of v_i as:

$$z_i^{(k)} = \sigma(W^{(k)} z_i^{(k-1)} + b^{(k)}), k = 1, 2, ..., K \tag{2}$$

where K represents the number of layers for encoder part. $z_i^{(k-1)}$ is the output of previous layer, $z_i^0 = \boldsymbol{x_i}$. z_i^K is the encoded representation of v_i and the $\sigma(\cdot)$ is the activation nonlinear function. $W^{(k)}$ and $b^{(k)}$ are the linear transformation weight matrices and bias vectors in k-th layer, respectively. The decoder part is symmetrical to encoder part which aims to reconstruct the high-dimensional output from the low-dimensional embedding z_i^K. In this paper, we construct a matrix $R \in \mathbb{R}^{n\times m}$ which is the normalized product of PPMI matrix $M_{ppmi} \in \mathbb{R}^{n\times n}$ and attributed information matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$. The i-th row in the matrix represents the aggregated result of structural information and attribute information for a given node v_i as:

$$R(i) = \frac{1}{n}\sum_{k=1}^{n} P_{ik}x_{ki} \qquad k = 1, 2, ..., n \tag{3}$$

where P_{ik} represents the global consistency between v_i and v_k. $R(i)$ combines global information with attribute information of node v_i. We replace x_i with this aggregated form to enforce the autoencoder to learn more information of node. Therefore, the autoencoder devotes to minimizing the reconstruction loss as follows:

$$\mathcal{L}_{AE} = \sum_{i=1}^{n} ||\boldsymbol{x}_i - R(i)||_2^2 \tag{4}$$

where $\hat{x}_i$ is the decoder output.

3.4 Adversarial Learning Module

The adversarial learning module is designed to learn regularized representations. It imposes a prior distribution to the output of encoder part. This process can be regarded as a regularization to graph representation learning. Adversarial learning module consists of a generator and a discriminator. The generator is the encoder part in A3Graph and the discriminator is implemented by a multi-layer perceptron (MLP) which tells encoder output latent vector from a prior distribution. In this paper, we use p to represent the real data distribution and q to represent the generated (fake) distribution. Therefore, $p(z)$ represents an arbitrary prior distribution, $q(z|x)$ represents encoding distribution and p_{data} represents input data distribution. In the proposed framework, the adversarial autoencoder defines an posterior distribution of $q(z)$ on learned vectors of autoencoder as follows:

$$q(z) = \sum_{1}^{n} q(z_i|x_i)p_{data}(x_i). \tag{5}$$

The adversarial learning module aims to guide the encoder posterior distribution $q(z)$ to match $p(z)$. In other words, the autoencoder is regularized by the adversarial procedure. Cross-entropy is used to measure the distance between the $p(z)$ and $q(z)$. Therefore, the loss function of adversarial learning part is defined as:

$$\mathcal{L}_{AD} = -\frac{1}{2}\mathbb{E}_{z\sim p(z)}\log\mathcal{D}(Z) - \frac{1}{2}\mathbb{E}_{z\sim q(z)}\log(1-\mathcal{D}(\mathcal{G}(Z))) \tag{6}$$

where $\mathcal{D}(\cdot)$ and $\mathcal{G}(\cdot)$ indicate the discriminator and generator.

3.5 Algorithm of A3Graph

In this section, we will first elaborate on the final objective function of A3Graph. Then, a detailed algorithm of A3Graph is presented.

Algorithm 1 Adversarial Attributed Autoencoder for Graph Representation

Require: graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, representation size d, hyper parameters α, β, λ, outer iterations T, discriminator iterations D
Ensure: : node representations $Z \in \mathbb{R}^{n\times d}$

```
Construct PPMI matrix based on Equation(1)
Construct reconstruction objective of decoder for each node based on Equation(3)
for t = 1,2,...,T do
    Sample h elements {x_(1), x_(2), ..., x_(h)} from X
    Generate latent h representations {z_(1), z_(2), ..., z_(h)} based on Equation(2).
    Generate h random liner interpolation coefficients {ε_(1), ε_(2), ..., ε_(h)}
    for k=1,2,...,D do
        Sample h entities{z'_(1), z'_(2), ..., z'_(h)} from prior distribution p_z
        Compute ẑ based on Equation(8)
        Compute the gradient of ∇L_AD and ∇L_GP based on Equation(7)
        Update adversarial learning module parameters
    end for
    Compute the gradient of ∇L_AE and ∇L_REG based on Equation(7)
    Update autoencoder module parameters
end forreturn node representatios Z
```

For autoencoder part, to avoid overfitting, we add the ℓ_2 norm to autoencoder loss function. For adversarial learning part, to improve the stability in training GAN and avoid the mode collapse, we add a gradient penalty to adversarial learning loss function as [10], namely, L_{GP}. The final loss function of A3Graph is defined as:

$$\begin{aligned}\mathcal{L} &= \alpha\mathcal{L}_{AE} + \beta\mathcal{L}_{REG} + \mathcal{L}_{AD} + \lambda L_{GP} \\ &= \alpha\sum_{i=1}^{n}||\boldsymbol{x}_i - R(i)||_2^2 + \frac{\beta}{2}\sum_{1}^{K}(||W^{(k)}||_F^2 + ||\hat{W}^{(k)}||_F^2) \\ &+ (-\frac{1}{2}\mathbb{E}_{z\sim p(z)}\log\mathcal{D}(Z) - \frac{1}{2}\mathbb{E}_{z\sim q(z)}\log(1-\mathcal{D}(\mathcal{G}(Z)))) \\ &+ \mathbb{E}_{z\sim p(\hat{z})}[(||\nabla_{\hat{z}}\mathcal{D}||_2 - 1)^2]\end{aligned} \tag{7}$$

where α is the hyper parameter to balance the loss of autoencoder module and adversarial module. β is the hyper parameter of ℓ_2 norm. $W^{(k)}$ and $\hat{W}^{(k)}$ are weight matrices of the encoder and decoder. $\hat{z}$ is the data calculated by linear interpolation between the prior distribution and encoder

distribution defined as:

$$\hat{z} = \epsilon \mathbb{E}_{z \sim p(z)} z + (1 - \epsilon) \mathbb{E}_{z \sim q(z)} z, \epsilon \in (0, 1) \quad (8)$$

where the $p(\hat{z})$ represents the distribution between $p(z)$ and $q(z)$ in straight line direction.

4 EXPERIMENTS AND DISCUSSION

4.1 Datasets

The statistics of three datasets are presented in Table 1, where $\mathcal{V}$, $\mathcal{E}$, $\mathbf{X}$ correspond to node set, edge set, and attributes set as defined previously and $\mathcal{C}$ denotes the label set. CiteSeer, Cora and Pubmed are three commonly used paper-citation graphs in several graph representation works [6, 9, 22]. The edges represent citation relationships between papers (cite or be cited). In this work, we regard these graphs as undirected graphs.

Table 1: Statistics of the datasets used in A3Graph.

Datasets	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathbf{X}\rvert$	$\lvert\mathcal{C}\rvert$
CiteSeer	3312	4715	3703	6
Cora	2708	5429	1443	7
Pubmed	19717	44338	500	3

4.2 Baseline Methods

In this subsection, we will make a brief introduction about baseline methods as follows:

- DeepWalk [22]: Deepwalk is the first method that utilizes random walk to sample node sequences in graph and incorporates NLP method to construct graph representations.
- LINE [25]: LINE combines the first-order and second-order structural information into output representations through node co-occurrence and node conditional probability.
- AIDW [6]: AIDW is an adversarial graph representation method incorporating random walk based methods to preserve structural information and adversarial learning to obtain robust and stable representation results.
- ARGA [21]: ARGA is an adversarial graph representation method. It leverages graph convolutional encoder to incorporate node structural and attributed information. Furthermore, an adversarial component is taken as a regularization for representation output.
- SEANO [14]: SEANO is a semi-supervised graph representation method. It designs a novel deep neural network framework to preserve the node structural and attributed and label information in a joint learning way.
- ANRL [29]: To learn a robust and stable representation, different with adversarial learning based methods, ANRL designs a neighbor nodes enhancement autoencoder framework with Skip-gram to combine structural and attributed information of graph.

4.3 Evaluation Metrics and Parameter Settings

For node classification task, we choose Accuracy as evaluation metric which are commonly used in lots of related works [22] [14] [29]. For node clustering task, we choose ACC (Accuracy), NMI (Normalized Mutual Information) and ARI (Average Rand Index) as evaluation metrics following by [6].

For CiteSeer, we set our framework hidden layer neurons of encoder to 1000 and 500, α to 1, β to 10. For Cora, we set our framework hidden layer neurons of encoder to 500, α to 1, β to 0.1. For Pubmed, we set our framework hidden layer neurons of encoder to 200, α to 50, β to 0.01. All learning rates in A3Graph are set to 0.0001 and the representation size is 128. The PPMI matrix is obtained by setting the order as 4 for CiteSeer, Cora, and Pubmed. We choose the Uniform distribution as real latent distribution in adversarial learning. For all baseline methods, we perform the implementation released by the original authors.

4.4 Node Classification

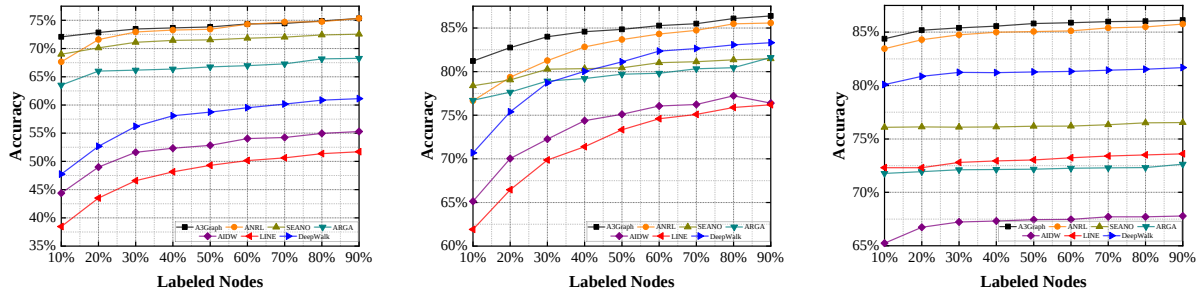
For node classification task, after obtaining representation results, we randomly sample 10% to 90% labeled nodes to train a support vector classifier [20] and use the rest data to test. By repeating this process 10 times, we report the average result of Accuracy. The detailed results are illustrated in Figure 2 and we have the following observations:

- A3Graph has achieved ideal performance results on three real-world datasets especially when the training ratio is lower than 30%. The structural-only information preserving methods perform worse than the structural and attributed information preserving methods except in the Pubmed and Cora (training ratio $< 40\%$). One reason may be that compared with CiteSeer, Cora and Pubmed have less attributed information and more structural information as shown in Table 1.
- A3Graph and ANRL perform better than the other methods in all three datasets. It demonstrates that the attributed enhancement based decoder criterion is crucial to guide neural network to learn robust and stale representations.
- It worth noting that, two adversarial learning methods ARGA and AIDW can not achieve ideal results in all three datasets. ARGA performs better than AIDW, showing that the structural and attributed information preserving is more effective than structure-only information preserving.

4.5 Node Clustering

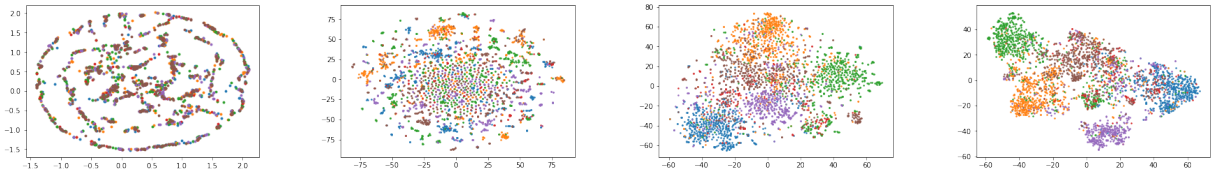
For node clustering task, we perform the K-means clustering algorithm on learned representations 10 times and calculate the average value of ACC, NMI, and ARI. The detailed results are shown in Table 2. We can obtain the following observations:

- A general observation we can draw from the result is that the proposed graph representation framework,



(a) CiteSeer (b) Cora (c) Pubmed

Figure 2: Node classification results of A3Graph on CiteSeer, Cora and Pubmed.



(a) DeepWalk (b) LINE (c) ANRL (d) A3Graph

Figure 3: Visualization results on CiteSeer dataset.

Table 2: Node clustering results of A3Graph on Cora and CiteSeer.

Datasets	CiteSeer			Cora		
Evaluation Metrics	ACC	NMI	ADJ	ACC	NMI	ADJ
DeepWalk	0.382	0.136	0.141	0.496	0.367	0.232
LINE	0.298	0.117	0.045	0.461	0.309	0.209
AIDW	0.281	0.083	0.026	0.323	0.171	0.052
ARGA	0.524	0.293	0.248	0.638	0.457	0.384
ANRL	0.519	0.326	0.249	0.508	0.388	0.277
A3Graph	**0.676**	**0.419**	**0.423**	**0.649**	**0.485**	**0.400**

A3Graph, has achieved the best performance on all datasets.

- Structural and attributed information preserving methods perform better than structure-only preserving methods. Both structural and attributed information should be incorporated in representation learning.
- Compared with node classification task, ARGA has made obvious improvement in the node clustering task. It demonstrates that the adversarial learning component affects the performance due to its adversarial procedure for improving the robustness of representation results.
- In node clustering task, ANRL performs better on CiteSeer than Cora. As previously mentioned, the structural information and attributed information are both crucial in graph representation. Even though ANRL incorporates a Skip-Gram component to preserve structural information, the neighbor attributed information enhancement mechanism plays a vital role in representation learning. However, only first-order structural attributed information preserving can not achieve the best representations in the clustering task.

4.6 Graph Visualization

In graph representation, a promising algorithm should preserve the original graph topology information well in a low-dimensional space. To prove the ability of A3Graph more intuitively, t-SNE [16] is utilized to visualize the represented latent vectors in a two-dimensional space. Figure 3 shows the visualization results on the CiteSeer dataset. It can be concluded that the A3Graph achieves impressive performance in preserving the topology information even though it learns the representations in an unsupervised manner. Meanwhile, compared with the results learned by ANRL, the visualization of A3Graph is more discriminative with a smaller distance of within-class and larger distance of inter-class.

4.7 Ablation Test

The proposed framework, A3Graph, contains two key modules: the first is the aggregated matrix and the second is the regularized adversarial learning module. In this subsection, we perform the ablation test to demonstrate the effectiveness of each module. The following A3Graph variants are designed for comparison.

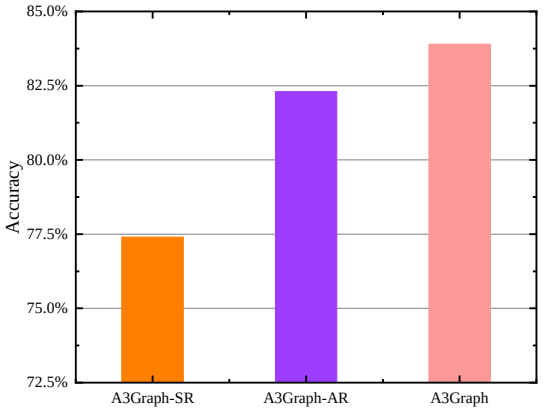


Figure 4: Ablation test result of A3Graph.

- A3Graph-SR: A variant of A3Graph with self-reconstruction in the autoencoder, the high-order PPMI matrix is removed.
- A3Graph-AR: A variant of A3Graph with the adversarial learning module being removed.

We perform the ablation test with node classification task on Cora. The training ratio is set to 30%. The result is shown in Figure 4. We can conclude that the A3Graph performs better than A3Graph-SR which confirms the superiority of the aggregated matrix module. Moreover, the performance of A3Graph and A3Graph-AR shows the effectiveness of adversarial learning module in A3Graph.

4.8 Parameter Sensitivity

In this subsection, we evaluate the affection of hyperparameters to the performance of the proposed framework. Specially, we investigate how the representation dimension size d, hyperparameters α and β affect node classification performance on CiteSeer with training ratio as 30%. Furthermore, we also evaluate node clustering performance with different prior distributions in adversarial learning components. It worth noting that all the other settings are set to default values besides the being tested parameter.

As shown in Figure 5, we investigate d, α and β from 8 to 128, 0.1 to 20 and 0.1 to 30, respectively. We can observe that for all three parameters, the accuracy of node classification shows a sharp increasing at first. Then, for d , there is an decreasing when it reaches around 128. For α there is slightly decreasing when it reaches to 1 and then becomes stable. However, for β, a drastic decreasing trend can be observed when it reaches to 10.

For prior distribution, we perform the node clustering task on CiteSeer and Cora. By replacing the prior distribution with Gaussian distribution, we find a similar performance in node clustering results (ACC:0.667, NMI:0.414, ADJ:0.419). It demonstrates both Uniform and Gaussian distributions can improve the robustness and stability of learned representations.

4.9 Extensive Discussion

In subsections 4.4-4.7, we report the experimental results of A3Graph in node classification, node clustering, and graph visualization. In this subsection, we aim to provide a more detailed discussion about the proposed framework.

First, A3Graph is an unsupervised graph learning framework by considering topology and structural information into a joint learning pipeline. The autoencoder is the basic feature extractor to learn effective features. The PPMI matrix, as the core part of A3Graph, has been extensively investigated in terms of natural language processing (NLP) to perform well on scenarios of calculating semantic similarity. In A3Graph, PPMI matrix is utilized to build the global consistency of a graph by measuring the co-occurrence between two nodes in a k-order context. Combined with the attribute information, the encode-decode structure will enforce the latent vectors to learn information in the aggregated matrix. ARGA adopts GCN as the basic feature extractor which only considers the local consistency in learning structural information. Moreover, GCN, as a representative graph embedding framework, is semi-supervised and incapable of supporting mini-batch training.

Second, the adversarial mechanism utilized in A3Graph can be seen as the regulation of the learned latent vectors. Most graph embedding methods learn the latent vectors without any regulations which leads to a free latent vector space for any structure [17]. In this work, we utilize the adversarial mechanism to make the learned embeddings match a predefined distribution results in regularized learned results for embeddings. It is the reason why A3Graph can bate SEANO and ANRL in the node classification task. These two baseline frameworks learn local and global topology information in an unregularized way which leads to a degenerate identity mapping.

Last, two reasons enable the robustness of A3Graph. The first is that we aggregate proximity information of a given node rather than itself. Assume that a node is an outlier that differs largely from its proximity nodes, this strategy will decrease the noise effects from this outlier node in learning the representation [29]. Second is the regularization strategy (adversarial learning module) which has been demonstrated in lots of works in eliminating the effects in a noise environment [17] [6] [21].

5 CONCLUSION

In this work, for graph representing, we propose a novel graph representation framework, A3Graph, to incorporate local and global structural information as well as attributed information in a joint learning way. We argue that most existing graph representation methods can not learn robust and stable results due to disinformation or misinformation in graphs. In A3Graph, we take PPMI matrix based attributed information aggregation as the objective of decoder reconstruction. Furthermore, we design an adversarial learning component to enforce the latent representations to be in accordance with a prior distribution. By adding these two regularization

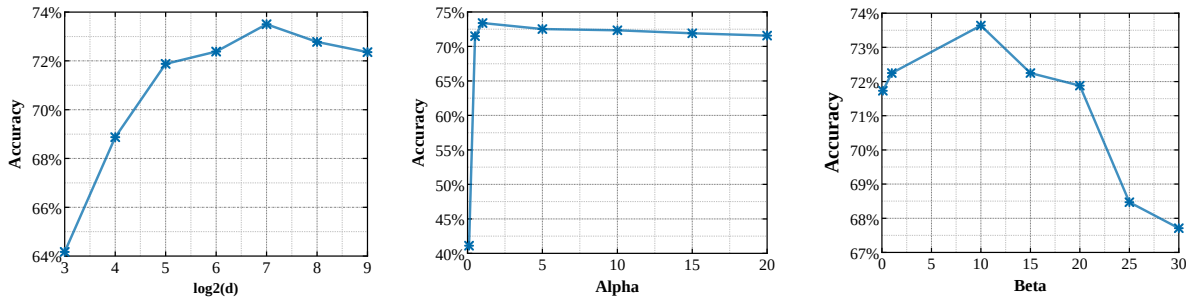


Figure 5: Parameter sensitivity results using node classification task on CiteSeer with train ratio as 30%

mechanisms, the learned vectors can be more robust and stable. Experimental results validate the effectiveness of the proposed framework.

REFERENCES

[1] Anish Acharya, Rahul Goel, Angeliki Metallinou, and Inderjit Dhillon. 2019. Online embedding compression for text classification using low rank matrix factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6196–6203.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017).

[3] Mikhail Belkin and Partha Niyogi. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*. 585–591.

[4] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. ACM, 891–900.

[5] Jianrui Chen, Danwei Liu, Fei Hao, and Hua Wang. 2020. Community detection in dynamic signed network: an intimacy evolutionary clustering algorithm. *Journal of Ambient Intelligence and Humanized Computing* 11, 2 (2020), 891–900.

[6] Quanyu Dai, Qiang Li, Jian Tang, and Dan Wang. 2018. Adversarial network embedding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[7] Fei Gao, Yue Yang, Jun Wang, Jinping Sun, Erfu Yang, and Huiyu Zhou. 2018. A deep convolutional generative adversarial networks (DCGANs)-based semi-supervised method for object recognition in synthetic aperture radar (SAR) images. *Remote Sensing* 10, 6 (2018), 846.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[9] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in neural information processing systems*. 5767–5777.

[11] Mingliang Hou, Jing Ren, Da Zhang, Xiangjie Kong, Dongyu Zhang, and Feng Xia. 2020. Network embedding: Taxonomies, frameworks and applications. *Computer Science Review* 38 (2020), 100296.

[12] Xiao Huang, Qingquan Song, Yuening Li, and Xia Hu. 2019. Graph recurrent networks with attributed random walks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 732–740.

[13] Xiangjie Kong, Yajie Shi, Shuo Yu, Jiaying Liu, and Feng Xia. 2019. Academic social networks: Modeling, analysis, mining and applications. *Journal of Network and Computer Applications* 132 (2019), 86–103.

[14] Jiongqian Liang, Peter Jacobs, Jiankai Sun, and Srinivasan Parthasarathy. 2018. Semi-supervised embedding in attributed networks with outliers. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 153–161.

[15] Jiaying Liu, Feng Xia, Lei Wang, Bo Xu, Xiangjie Kong, Hanghang Tong, and Irwin King. 2019. Shifu2: A Network Representation Learning Based Model for Advisor-advisee Relationship Mining. *IEEE Transactions on Knowledge and Data Engineering* (2019). https://doi.org/10.1109/TKDE.2019.2946825

[16] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[17] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644* (2015).

[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[19] John F Nash et al. 1950. Equilibrium points in n-person games. *Proceedings of the national academy of sciences* 36, 1 (1950), 48–49.

[20] Jackie L Norrie, Marybeth S Lupo, Beisi Xu, Issam Al Diri, Marc Valentine, Daniel Putnam, Lyra Griffiths, Jiakun Zhang, Dianna Johnson, John Easton, et al. 2019. Nucleome Dynamics during Retinal Development. *Neuron* 104, 3 (2019), 512–528.

[21] Shirui Pan, Ruiqi Hu, Sai-fu Fung, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Learning graph embedding with adversarial training methods. *arXiv preprint arXiv:1901.01250* (2019).

[22] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[23] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290, 5500 (2000), 2323–2326.

[24] Xiao Shen, Quanyu Dai, Sitong Mao, Fu-lai Chung, and Kup-Sze Choi. 2020. Network together: Node classification via cross-network deep network embedding. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[25] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 1067–1077.

[26] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1225–1234.

[27] Feng Xia, Jiaying Liu, Hansong Nie, Yonghao Fu, Liangtian Wan, and Xiangjie Kong. 2019. Random Walks: A Review of Algorithms and Applications. *IEEE Transactions on Emerging Topics in Computational Intelligence* 4, 2 (2019), 95–107.

[28] Shuo Yu, Jin Xu, Chen Zhang, Feng Xia, Zafer Almakhadmeh, and Amr Tolba. 2019. Motifs in Big Networks: Methods and Applications. *IEEE Access* 7 (2019), 183322–183338.

[29] Zhen Zhang, Hongxia Yang, Jiajun Bu, Sheng Zhou, Pinggang Yu, Jianwei Zhang, Martin Ester, and Can Wang. 2018. ANRL: Attributed Network Representation Learning via Deep Neural Networks.. In *IJCAI*, Vol. 18. 3155–3161.