AGAN: ATTRIBUTE GENERATIVE ADVERSARIAL NETWORK

M. Zhang[*], *P. Xu*[*†]

Abstract: Graph generative adversarial network has achieved remarkable effectiveness, such as link prediction, node classification, user recommendation and node visualization in recent years. Most existing methods mainly focus on how to represent the proximity between nodes according to the structure of the graph. However, the graph nodes also have rich attribute information in social networks, the traditional methods mainly consider the node attributes as auxiliary information incorporate into the embedding representation of the graph to improve the accuracy of node classification and link prediction. In fact, in social networks, these node attributes are often sparse. Due to privacy and other reasons, the attributes of many nodes are difficult to obtain. Inspired by the application of generative adversarial network in image field, we propose an innovative framework to discover node latent attribute. Through experiments, we demonstrate the effectiveness of our proposed methods.

Key words: *node attribute, generative adversarial network, network embedding, policy gradient, bidirectional long short term memory*

Received: August 19, 2018 **DOI:** 10.14311/NNW.2021.31.008
Revised and accepted: April 30, 2021

1. Introduction

More and more attention has been paid to network representation learning in recent years. It learns a low-dimensional embedding vector to capture highly nonlinear network structures, especially for large-scale networks such as social networks and citation networks, which benefits some network downstream tasks, such as clustering, link prediction, visualization, user recommendation in social network, community discovery, product recommendation of e-commerce website, etc.

In the past decades, a large number of network embedding methods have been proposed to discover the latent structural proximity of networks. DeepWalk [1] first introduces deep learning to network embedding task, which utilizes unsupervised features to represent learning and obtains the local network structure representation through random walk and skip-gram. Specifically, it maximizes the likelihood

[*]Maoyuan Zhang; Peihua Xu – Corresponding author; School of educational information technology, Faculty of Artificial Intelligence Education, Central China Normal University, Wuhan 430079 China; E-mail: 36109439@qq.com; National Language Resources Monitoring & Research Center for Network Media, Central China Normal University, Wuhan 430079 China;
[†]Peihua Xu; Hubei Meteorological Service Center, Wuhan 430079 China

©CTU FTS 2021

probability distribution of a given node. LINE [2] proposes the breadth-first search (BFS) strategy to discover the first-order and second-order proximity of weighted and unweighted networks. Node2Vec [3] extends this idea further, using both breadth first search (BFS) and depth first search (DFS) to learn network structures. DANE [4] is proposed by designing an AutoEncoder to learn the similarity of the topology of the attribute network.

In addition to the above these deep learning model, affected by the generate adversarial network which has successfully applied in the image field [5–8], generate adversarial network is introduced into the network representation learning. However, different from the image, the complexity, disorder and discreteness of the graph structure hinder the application of GAN in the graph structure representation learning, especially the discreteness prevents these classical methods from being used on the graph structure. Therefore, some improved methods are proposed, such as GraphGAN [9] using the policy gradient Monte-Carlo search algorithm for network presentation learning, NetGAN [10] using Wasserstein distance to generate network subgraph, ProGAN [11] using the triple representation method to capture network proximity, HeGAN [12] extending the application domain to heterogeneous information network.

Although these methods have been successful in network representation learning, most of the current algorithms only focus on the proximity of network structure, the connection distribution between nodes, clustering analysis, and ignore the rich description information in attribute network. However, these descriptions contain very important content. As shown in Fig. 1, in social networks, user A is connected to user B, and user B is connected to user D through C. The attribute

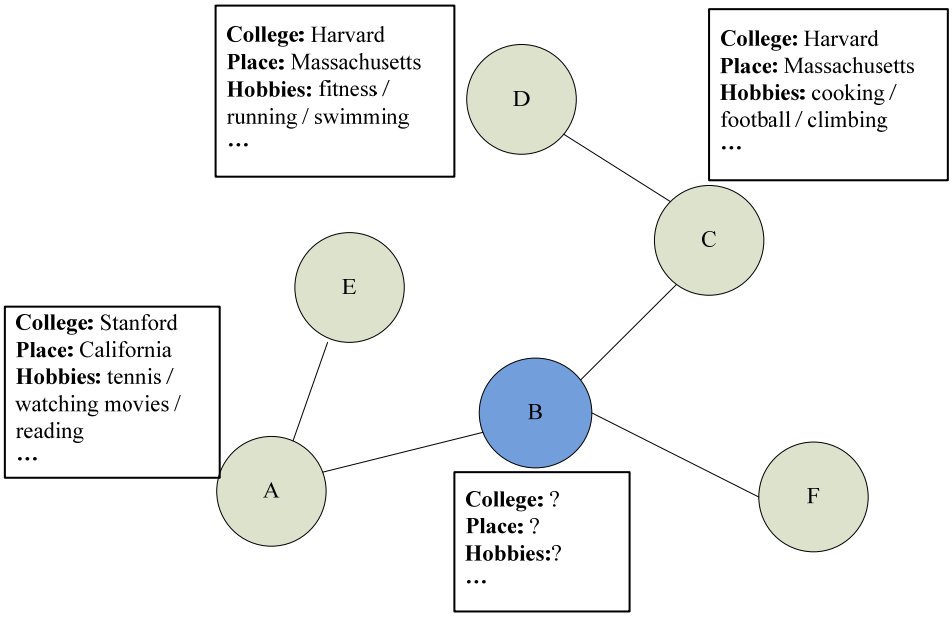

Fig. 1 *An example of attribute network.*

information of user A, C and D, such as college, place and hobbies, is known, but the attribute of B cannot be obtained. How to generate the corresponding attribute through the node connected to B? Through the graph, we can intuitively infer the attribute of B. in the real world, B is more closely connected with A and C (the distance is shorter), so B may have more common attributes (hobbies) with A and C. However, we can reasonably infer that B may also have some common attributes (hobbies) with D.

In order to address above intuitionistic and interesting problem, inspired by the application of GANs in image, we propose a robust attribute generate adversarial network (AGAN) to generate the attributes of nodes according to the given node with context network information. As far as we know, this is the first to apply GAN to the generation task of graph network attributes. In this paper, we generate corresponding attribute information according to the neighboring information of nodes in social network, such as users' political tendency, interest and hobbies, etc. However, graph network structure information and attribute text information belong to two different modal data structures, so how to integrate these two kinds of information is a challenge. Our contributions as follows:

(1) We propose an AGAN neural network framework to generate the attributes of network nodes. We solve the challenge of discover node latent attribute through highly nonlinear network structure with given node context, and propose an improved variant Bi-AGAN method.

(2) Under the condition of no strong assumption, the attribute of given node is generated smoothly by AGAN, which makes it approximate to the attribute distribution of network node in the real world.

(3) We establish a data set of our own, and carried out extensive experiments on this data set. By comparing with the baseline method, experiment proves the effectiveness of our method.

2. Related Work

So far, network representation learning has attracted more and more attention, and different network representation learning methods have been proposed. In this section, we will review these related work.

Network embedding maps network structure or node data to a low dimensional latent space. Perozzi et al. [1] proposed a seminal method for embedding deepwalk networks. It employs random walk to obtain node sequences, and uses skip-gram model to learn node representation. Inspired by this method, a large number of network representation methods based on deepwalk have emerged. For example, Grover et al. [3] proposed a node2vec method, which uses breadth first and depth first methods to preserve local and global proximity. Cao et al. [13] proposed the grarep method to preserve the high order approximation of nodes.

However, most of these methods are only based on network topology, ignoring the importance of node attributes. Therefore, some researchers propose joint embedding methods of attribute and network topology. For example, Yang et al. [14] proposed a linear inductive matrix decomposition method which combines network

topology and attributes, and Huang et al. [15] utilized the method of graph Laplacian to jointly embedding from topology and attribute learning. Cao et al. [4] proposed a deep attribute network embedding method, which can preserve proximity of topological structure and node attributes.

Recently, some researchers propose a network embedding method based on GAN [16–19], which is used to learn robust latent representation in various applications. GANs used the idea of adversarial learning, through the competition between discriminators and generators, learning the latent data distribution of the network, making the model more robust to sparse or noisy data. For example, Wang et al. [9] proposed GraphGAN simulates the probability of each individual edge. Bojchevski et al. [10] proposed a NetGAN model using Wasserstein GAN to train targets, which can generate graphs of known network patterns.

Most of the aforementioned methods are aimed at the prediction, classification and visualization of network nodes. Although some researches consider the inference of network node attributes by explicit probability model. For example, Gong et al. [20] proposed a social behavior attribute (SBA) network model, and established a voting distribution attack based on SBA model to infer the attributes of social network users. Jia et al. [21] proposed a Markov random probability model based on the definition of social network structure, and used the loopy belief propagation (LBP) to infer the attributes of users. However, these methods are all shallow models, and there is little work to generate the attributes of network nodes through deep learning.

3. Preliminaries

Long short term memory (LSTM) network has shown significant improvement in some NLP applications [22–24]. In order to better capture the potential dependence between vertices and attributes and improve the generalization ability of the model, we use one-dimensional convolutional neural network to extract feature information, and then use bidirectional long short term memory (Bi-LSTM) [25–27] to automatically capture and characterize the information of nodes and attribute text sequences. Thus, we review some foundational knowledge about Bi-LSTM.

Generally, a typical LSTM recurrent neural network consists of four components: one input gate i_t with corresponding weight matrix W_i and bias vector b_i; one forget gate f_t with corresponding weight matrix W_f and bias vector b_f; one memory gate c_t with corresponding weight matrix W_c and bias vector b_c; one output gate o_t with corresponding to the weight matrix W_o and bias vector b_o, h_{t-1} denotes the previous state, h_t denotes the current state, $\odot$ denotes element-wise multiplication, and the detailed calculation formula as follows:

Input gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i). \tag{1}$$

Forget gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \tag{2}$$

State update:

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \tag{3}$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1}. \tag{4}$$

Output gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{5}$$

$$h_t = o_t \odot \tanh(c_t). \tag{6}$$

Bi-directional LSTM (Bi-LSTM) consists of two LSTMs, which considers the input characteristics of the past and the future. The calculations of Bi-LSTMs can be formulated as follows:

$$\begin{aligned} \vec{h}_t &= LSTM(x_t, \vec{h}_{t-1}), \\ \overleftarrow{h}_t &= LSTM(x_t, \overleftarrow{h}_{t-1}), \\ h_t &= \vec{h}_t + \overleftarrow{h}_t. \end{aligned} \tag{7}$$

4. Deep attribute generate adversarial network

In this section, we describe our AGAN—a deep attribute generate adversarial network framework and the implementation of generator and discriminator. Then we present the training strategy.

4.1 AGAN framework

The overall framework of our model is shown in Fig. 2. The generator receives node contexts and noisy attribute distribution, and then generates the joint embedding representation as the output. The discriminator utilizes the node contexts (real node and synthesized node) and node attribute (real attribute text and noise attribute text) to learn the context and attribute joint embedding matrix. Furthermore, LSTM indicates long short term memory network, LN indicates layer normalization.

As same as other classical GAN model, our AGAN has two components: a generator and a discriminator. Because the node structure and the attribute text of node are discrete, we use DeepWalk to sample the node context to generate the node context sequence set. The attribute text of node forms the attribute text set, and then uses LSTM coding respectively. We adversarial train generator (G) and descriminator (D) based on node attribute feature.

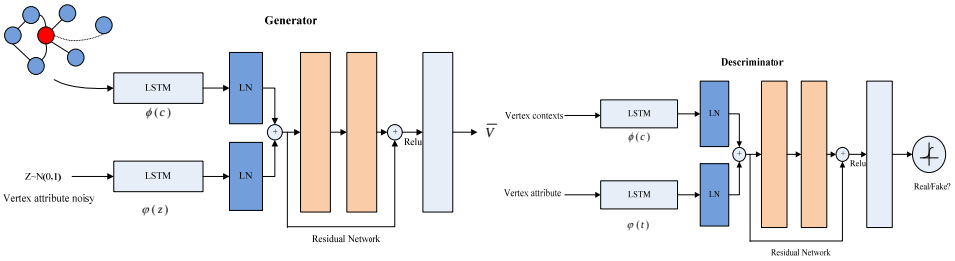

Fig. 2 *The framework of our proposed model.*

In the GAN model, the generator and the discriminator can be regarded as two players playing a maximum-minimum game. The generator tries to generate false sample data to deceive the discriminator D, while the discriminator tries to distinguish false and real samples. Specifically, in our model, the generator tries to perfectly fit the real attributes distribution of vertices, while the discriminator tries to detect which are the attributes of vertices generated by G and which are the real attributes of vertices. A typical GAN is defined as follows [16]:

$$\min_\phi \max_\theta V(G,D) = E_{x\sim p(x)}[\log D_\theta(x)] + E_{z\sim p(z)}[\log(1 - D_\theta(G_\phi(z)))], \quad (8)$$

where $p(x)$ represents the latent distribution of real sample space, $p(z)$ denotes distribution space of prior sample, $G\phi(\cdot)$ and $D_\theta(\cdot)$ denote generator and discriminator respectively. Goodfellow et al. proved effectiveness of Eq. (5) [16]. Theoretically, there is a Nash equilibrium in which the generator will generate a distribution that approximates real distribution. In other words, generator G can map the prior distribution $p(z)$ to the complex real distribution $p(x)$. However, we want to generate attributes for vertices, not just approximate the real distribution. Therefore, origin GAN does not solve this problem.

In our propose an AGAN framework, different from the original GAN, we denote the real node as V, the synthesized node as $\overline{V}$, the real attribute text as T, and the noise attribute text as $\overline{T}$. Discriminator D can distinguish true and false from three types of data, which are the real node context pair with real attributes (V,T), the synthesized node context pair with real attributes $(\overline{V},T)$ and the real node context pair with noise attributes $(V,\overline{T})$. Through optimization learning, the discriminator can provide auxiliary signals to the generator. Our objective function can be defined as follows:

$$\begin{aligned}\min_\phi \max_\theta V(G,D) \quad = \quad & E_{x\sim p(x)}[\log D_\theta(V,T)] + E_{\phi\sim p(\phi)}[\log(1 - D_\theta(G_\phi(\overline{V},T))) \\ & + \log(1 - D_\theta(G_\phi(V.\overline{T})))]. \end{aligned} \quad (9)$$

4.2 Node contexts and node attributes deep joint embedding

Definition 1. (Graph) A graph is defined as $G = \{V, C, T, L\}$, where $V = \{v_1, v_2, \ldots, v_n\}$ represents n nodes, $C = \{c_1, c_2, \ldots, c_n\}$ represents a set of node context which random sample by DeepWalk, $T = \{t_1, t_2, \ldots, t_n\}$ represents node attribute text and $L = \{l_1, l_2, \ldots, l_n\}$ represents node class.

As the same as previous multimodal works [5], in order to obtain a discriminator vector representation of given node contexts and attributes, we learn a deep joint embedding of node contexts and node attributes. Specifically, we employ the inner product of two LSTM feature encoders to learn the corresponding functions of node proximity and node attributes. In short, by maximizing the loss of matching node contexts and node attributes and minimizing the loss of node attributes and other node contexts, the loss function is defined as follows:

$$L = \frac{1}{N}\sum_{i=1}^{N} \Delta(l_i, f_c(c_i)) + \Delta(l_i, f_t(t_i)), \quad (10)$$

where Δ is the $0-1$ loss. Classifiers f_c and f_t are defined as follows:

$$f_c(c) = \arg\max_{l\in L} E_{t\sim T(L)}[\phi(c)^T \varphi(t)], \quad (11)$$

$$f_t(t) = \arg\max_{l \in L} E_{t \sim C(L)}[\phi(c)^T \varphi(t)], \tag{12}$$

where ϕ and φ represent node context encoder and node attribute text encoder (e.g., LSTM) respectively. $T(L)$ is the set of node attribute texts of class l and likewise $C(L)$ for node contexts. In order to minimize equation 11, we used character-level convolutional networks for text classification [28, 29].

4.3 Discriminator for AGAN framework

Three types of node context and attribute text pairs, such as node context pair with real attributes, synthesized node context pair with real attributes and real node context pair with noise attributes were employed. In order to evaluate score of these pairs, the discriminator D is modeled the interaction between the observed node context and the corresponding node attributes, in which three neural network frameworks are used: long short term memory layer, residual neural network and full connection layer, as shown in Fig. 2.

In the discriminator, node context and node attributes are mapped to two different embedding spaces which are related to each other. $C^{r \times d}$ denotes node context embedding matrix, $T^{m \times k}$ denotes node attributes embedding matrix, and r and m representing the number of node and attribute texts respectively. It is worth noting that $k = d$ in this paper, that is to say, the dimension of node context and node attribute embedding matrix is equal to the given node.

The layer normalization which is a classic regularization layer is connected after the node context encoder and attribute text encoder to avoid over-fitting [30], and the output vectors of the two encoders are denote as $E_\phi \in R^r$ and $E_\varphi \in R^m$ respectively. Then the E_ϕ and E_φ are connected and input to the later network layer. In order to accelerate convergence and avoid gradient vanish, the residual network is used. Finally, through the calculation of the full connection layer by the discriminator, the score can be formally expressed as follows:

$$D(C, T) = \text{Concat}(E_\phi, E_\varphi) \cdot M \cdot F \cdot K, \tag{13}$$

where $M \in R^{(r+m) \times k}$ denotes residual network for feature mapping and extraction, $F \in R^{k \times n}$ denotes full connection layer, $K \in R^{n \times 1}$ denotes a vector which is utilized to calculate the final score.

4.4 Generator for AGAN framework

The goal of generator G is to generate a graph topology by sampling a given node with random walk. A series of random walk sequences with length t of node v is randomly extracted from graph G, that is, the context set c of node v, the node context embedding $\phi(c)$ is obtained by node context encoder ϕ, the node attributes embedding $\varphi(t)$ obtained by attribute text encoder φ which generate by Z. Utilizing layer normalization to normalize embedding vector $\phi(c)$ and $\varphi(z)$ respectively, and then connect the two embedding vectors. Using $G(\phi(c), \varphi(t))$ to generate the joint embedding representation $\overline{V}$ of node context and node attributes.

The traditional stochastic gradient descent method is suitable for continuous real value space. But our graph structure and node attribute text are discrete, we can't use the traditional GANs stochastic gradient descent method to optimize.

Specifically, the discriminator uses the calculated gradient to guide the generator to approach the real distribution through slight changes. In the field of computer vision, for example, the RGB three channel image is continuous in the real value space $R \in [0, 255]$, but for the discrete text, the discriminator can't propagate the gradient back to the generator, resulting in the loss of the training direction of generator.

Several studies have proposed to address aforementioned application problems of GANs. One is to use the Gumbel distribution method proposed by Jang et al. [31], which uses a gradient estimator to replace the non differentiable samples in the distribution with the differentiable samples Gumbel softmax distribution; another method, based on reinforcement learning(RL), utilizes policy gradient method proposed by Yu et al. [17]. To avoid the non differentiable problem of generator, the random strategy in RL is used to model generator and update policy gradient and return to the intermediate state through Monte Carlo search. Following this idea, Wang et al. proposed an improved GAN framework for graph network using policy gradient.

Inspired by the policy gradient, we utilize the policy gradient to optimize our generator. The policy gradient derivation of AGAN generator is calculated as follows:

$$
\begin{aligned}
\nabla_\phi L_{G_\phi} &= \nabla_\phi E_{\phi\sim p(\phi)}[\log(1 - D_\theta(G_\phi(\overline{V}, T))) + \log(1 - D_\theta(G_\phi(V, \overline{T})))] \\
&= \sum_{i=1}^{N} \nabla_\phi p_\phi(G_\phi(\overline{V}_i, T_i) \log(1 - D_\theta(G_\phi(\overline{V_i}, T_i))) + \\
&\quad \sum_{i=1}^{N} \nabla_\phi p_\phi(G_\phi(V_i, \overline{T}_i) \log(1 - D_\theta(G_\phi(V_i, \overline{T}_i))) \\
&= \sum_{i=1}^{N} p_\phi(G_\phi(\overline{V}_i, T_i) \nabla_\phi \log p_\phi(G_\phi(\overline{V}_i, T_i) \log(1 - D_\theta(G_\phi(\overline{V_i}, T_i))) + \\
&\quad \sum_{i=1}^{N} p_\phi(G_\phi(V_i, \overline{T}_i) \nabla_\phi \log p_\phi(G_\phi(V_i, \overline{T}_i) \log(1 - D_\theta(G_\phi(V_i, \overline{T}_i))) \\
&= E_{\phi\sim p(\phi)}[\nabla_\phi \log p_\phi(G_\phi(\overline{V}, T) \log(1 - D_\theta(G_\phi(\overline{V}, T))) + \\
&\quad \nabla_\phi \log p_\phi(G_\phi(V, \overline{T}) \log(1 - D_\theta(G_\phi(V, \overline{T}))] \\
&\approx \frac{1}{M}[\nabla_\phi \log p_\phi(G_\phi(\overline{V}, T) \log(1 - D_\theta(G_\phi(\overline{V}, T))) + \\
&\quad \nabla_\phi \log p_\phi(G_\phi(V, \overline{T}) \log(1 - D_\theta(G_\phi(V, \overline{T}))]. \qquad (14)
\end{aligned}
$$

where distribution $p(\phi)$ is determined by the parameter ϕ of generator. When the parameter ϕ is updated in each iteration, the new distribution $p(\phi)$ is approximate to the real distribution. M denotes the number of samples, for more details in Algorithm 1.

Algorithm 1 Attribute generative adversarial network based on policy gradient.

while AGAN not converge **do**
  for G-steps **do**
    Sample the node deepwalk sequence and node noise attribute text
    G generates synthesis node $\overline{V}$
    Update G parameter according to Eq. (9) and Eq. (14)
  end for
  for D-steps **do**
    Sample real node context pair with real attributes (V, T)
    Sample the synthesized node context pair with real attributes $(\overline{V}, T)$
    Sample the real node context pair with noise attributes $(V, \overline{T})$
    Update D parameter according to Eq. (9), Eq. (13) and Eq. (14)
  end for
end while

5. Experiment Settings

In this section, we firstly establish our own data set, and then compare the performance of our proposed framework AGAN and variant method Bi-AGAN with baseline methods.

5.1 Data set

In order to evaluate the performance of our proposed framework, we investigate several public attribute networks, such as Facebook and twitter datasets. These datasets are anonymously transformed into bag vectors for privacy reasons, which is not suitable for our experiment.

We worked with a talent website to create a new dataset for this trial. We invite users to participate in our experiment by email, and fill in user attributes online. In order to make the attribute data rich and diverse, we selected 10 attributes (such as school, major, political inclination, hobbies, employers, etc.) for the invited users to fill in. About 12,000 people were sent an invitation email and asked them to forward their circle of friends to invite them to attend. About 50,000 users and about 100,000 connections were collected. College and work place information is extracted through education and place.

5.2 Comparison methods

In our experiments, we randomly select 10 %, 20 %, 30 %, 50 %, 80 % users as the training set and remove rest users attribute as the testing set respectively. We employ Accuracy, Recall and F-Score as evaluation metrics. A node may have multiple attribute; we define that a node is correctly predicted if the predicted value matches any of its ground truth attribute.

To show the effectiveness of our co-profiling algorithm, we compare our proposed model AGAN and variant model Bi-AGAN with two typical attribute reasoning baseline methods of social network. These methods briefly introduce as follow:

VIAL [20]: VIAL defines the attribute reasoning of network nodes as an attack on the privacy information of network users and proposes a social-behavior-attribute (SBA) net-work model and design the vote distribution attack (VIAL) to perform attribute inference based on the SBA model.

Attr-Infer [21]: Attr-infer is a Markov random probability model based on the definition of social network structure. It infers a posteriori probability by using loopy belief propagation (LBP), and then infers the attributes of users by using a posteriori probability.

AGAN: Our framework of attribute generation model, which is an implicit GAN model. It uses node context and node attribute text to generate the missing attribute information of vertices.

Bi-AGAN: An improved AGAN variant method, which replaces LSTM with Bi-LSTM, and uses one-dimensional convolutional neural network to extract features of input information, and then input it into Bi-LSTM network.

5.3 Parameter settings

The parameter settings of VIAL and Attr-Infer are as same as the original paper. We apply the Adam optimization method for AGAN training. The hyper-parameters of our model mainly include learning rate $l = \{1e^{-2}, 1e^{-3}, 1e^{-4}\}$, Node contexts and node attributes deep joint embedding dimension $d = \{50, 70, 100, 150\}$, batch size $m = \{32, 64, 128, 256\}$, deep walk length $s = \{10, 30, 50, 60, 80\}$. The grid search method is used to determine the hyper-parameters, and the optimal configuration of model parameters is obtained on the validation set. The final optimal parameters are $\{l = 1e^{-3}, d = 100, m = 128, s = 50\}$.

6. Results and Analysis

method	10 %	20 %	30 %	50 %	80 %
VIAL	30.12	40.37	54.29	57.12	60.32
Attr-Infer	31.35	45.45	56.33	58.76	67.67
AGAN	32.46	50.66	63.64	68.83	75.75
Bi-AGAN	34.85	54.71	67.01	74.95	84.85

Tab. I *Accuracy performance of user's college attribute of different method (%).*

In this section, we depict our experiment results. In this paper, limited to space, we only show the college attribute of the generated node, and compare our method AGAN conclusion with the other baseline methods.

As described in Algorithm 1, the input data of our model are: real node context and real attribute college (V, T), synthesized node context pair with real college

method	10 %	20 %	30 %	50 %	80 %
VIAL	30.63	32.42	33.93	34.27	45.04
Attr-Infer	32.78	34.58	36.15	49.86	55.32
AGAN	34.66	38.82	45.97	55.78	66.58
Bi-AGAN	35.71	41.88	50.49	63.84	75.97

Tab. II *Recall performance of user's college attribute of different method (%).*

method	10 %	20 %	30 %	50 %	80 %
VIAL	23.12	27.23	34.04	36.12	55.32
Attr-Infer	28.61	33.51	35.48	38.52	58.94
AGAN	31.46	34.68	37.64	48.83	65.98
Bi-AGAN	31.96	37.52	39.93	53.87	73.37

Tab. III *F-score performance of user's college attribute of different method (%).*

attribute $(\overline{V}, T)$, the real node context pair with noise college attribute $(V, \overline{T})$, and the predicted variable is node's college attribute. We run each method five times and calculate the average value as the final evaluation result. Tab. I to Tab. II show the accuracy, Recall and F-score performance indicators of these four algorithms. It can be seen from Tab. I that AGAN method is slightly better than VIAL and Attr-Infer methods when the training set is 10 %. With the increase of training data set size, the accuracy rates of the four methods are improved. When the training set reaches 80 %, the accuracy of AGAN is 15.43 % and 8.08 % higher than that of VIAL and Attr-Infer methods, respectively. The improved variant Bi-AGAN method has better performance than VIAL, Attr-Infer and AGAN. When the training set is 10 %, the advantage is not obvious. When the training set reaches 80 %, the accuracy of Bi-AGAN is 9.1 % higher than AGAN. Compared with the traditional explicit models, such as reasoning method and probability model, implicit generation model AGAN has better performance. However, compared AGAN framework with LSTM network, variant Bi-AGAN framework with one-dimensional convolutional neural network and Bi-LSTM models can extract network features and capture the latent distribution of node attributes better.

Through in-depth analysis, we found that the same attribute in different nodes have different descriptions, such as abbreviations, singular and plural problems, which affect the accuracy of prediction. It is worthwhile to further study that AGAN combined with the semantics of node attribute text.

In order to have a more intuitive display of the generated node attributes, we draw the attribute fraction of nodes (users) having a common college. Different colleges are represented by index, as shown in Fig. 3. It can be seen from Fig. 3 that the distribution of the real value could be well reflected by our proposed model, which has a good stability and robustness.

In terms of computational time complexity, the computational time of Attr-

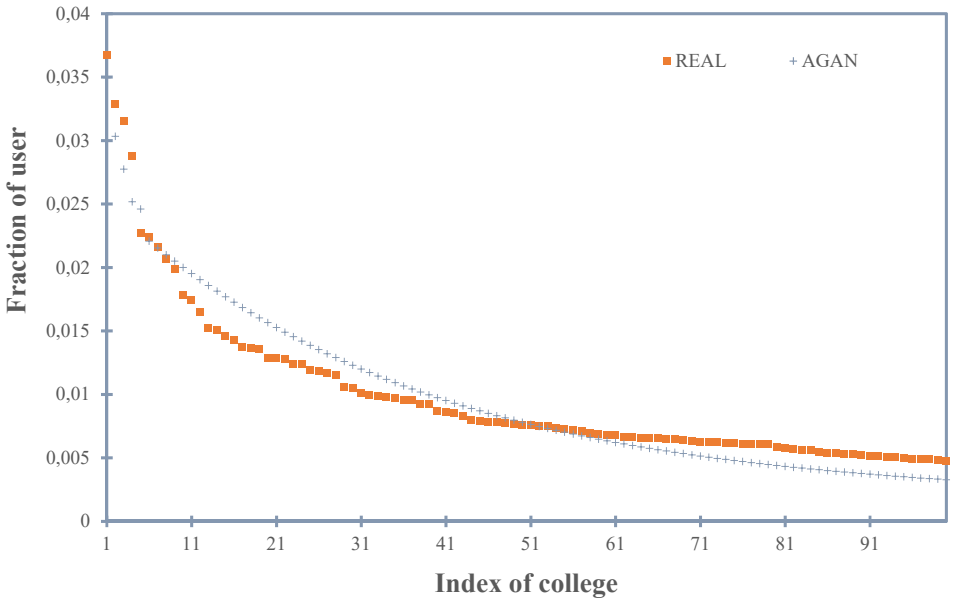

Fig. 3 *College fraction of user.*

Infer is composed of the time needed to compute the prior probability by logistic regression and the time needed to copute the posterior probability by iteration. The time complexity of the second part is $O(t \cdot |V|)$, where t is the number of iterations. VIAL computes the stationary probability distribution of random walks in the augmented graph starting from the target node. The time complexity of VIAL is $O(t \cdot (|V| + m_1 + m_2) \cdot n_t)$, where t is the number of iterations required for random walk convergence, m_1 is the total number of attributes of all nodes, m_2 is the total number of behaviors of all nodes, and n_t is the number of nodes. When the number of nodes increases, it has higher time complexity than Attr-Infer. The time complexity of our model AGAN is determined by the convergence times of generator, discriminator and deep walk algorithm, and the time complexity of AGAN is $O(t \cdot (|V| \log |V|)$, where t is the number of iterations required for AGAN convergence. Obviously, like other deep learning model, AGAN has higher time complexity but higher prediction accuracy compared with attribute reasoning model VIAL and Attr-Infer based on Markov probability model.

7. Conclusion

In this paper, we propose an attribute generate adversarial network framework called AGAN for the node attribute generation problem. In this framework, we use context encoder and attribute text encoder to encode and incorporate for different modalities. The implicit distribution for latent representation is modeled in a generate adversarial learning method. We further evaluate the quality of generated note attributes through generating three different node attributes. The

experimental results show that this method is superior to other explicit modeling method.

References

[1] PEROZZI B., AL-RFOU R., SKIENA S. DeepWalk: Online learning of social representations. In: *2014 ACM Conference on Knowledge Discovery and Data Mining*. 2014, pp. 701–710, doi: 10.1145/2623330.2623732.

[2] TANG J., QU M., WANG M., ZHANG M., YAN J., MEI Q. LINE: Large-scale information network embedding. *WWW 2015 – Proceedings of the 24th International Conference on World Wide Web*. 2015, pp. 1067–1077, doi: 10.1145/2736277.2741093.

[3] GROVER A., LESKOVEC J. Node2Vec. In: *2016 ACM Conference on Knowledge Discovery and Data Mining*. 2016, pp. 855–864, doi: 10.1145/2939672.2939754.

[4] GAO H., HUANG H. Deep attributed network embedding. In: 2018 *IJCAI International Joint Conference on Artificial Intelligence*. 2018, pp. 3364–3370, doi: 10.24963/ijcai.2018/467.

[5] REED S., AKATA Z., LEE H., SCHIELE B. Learning deep representations of fine-grained visual descriptions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2016, pp. 49–58, doi: 10.1109/CVPR.2016.13.

[6] REED S., AKATA Z., YAN X., LOGESWARAN L., SCHIELE B., LEE H. Generative adversarial text to image synthesis. In: 2016 *33rd International Conference on Machine Learning, ICML*. 2016, pp. 1681–1690.

[7] MANSIMOV E., PARISOTTO E., BA J.L., SALAKHUTDINOV R. Generating images from captions with attention. In: *2016 International Conference on Learning Representations*. 2016, pp. 1–12.

[8] MIRZA M., OSINDERO S. Conditional generative adversarial nets. 2014, pp. 1–7, [Online]. Available: http://arxiv.org/abs/1411.1784.

[9] WANG H., WANG J., WANG J.L. learning graph representation with generative adversarial nets. *IEEE Transactions on Knowledge and Data Engineering*. 2019, pp. 2508–2515, doi: 10.1109/TKDE.2019.2961882.

[10] BOJCHEVSKI A., OLEKSANDR S., ZUGNER D., GUNNEMANN S. NetGAN: Generating graphs via random walks. In: *Proceedings of the 35th International Conference on Machine Learning*. 2018, pp. 1–16, arXiv:1803.00816v2.

[11] GAO H., PEI J., HUANG H. Progan: Network embedding via proximity generative adversarial network. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019, pp. 1308–1316, doi: 10.1145/3292500.3330866.

[12] CHEN H., HU Y., PEROZZI B., SKIENA S. HARP: Hierarchical representation learning for networks. *2nd AAAI Conference on Artificial Intelligence*. 2018, pp. 2127–2134.

[13] CAO S., LU W. GraRep: Learning graph representations with global structural information. In: *The 24th ACM International Conference on Information and Knowledge Management*. 2015, pp. 891–900.

[14] YANG C., LIU Z., ZHAO D., SUN M., CHANG E.Y. Network representation learning with rich text information. In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015, pp. 2111–2117.

[15] HUANG X., LI J., HU X. Label informed attributed network embedding. In: *International Conference on Web Search and Data Mining*. 2017, pp. 731–739.

[16] GOODFELLOW I.J., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D. Generative adversarial nets. In: *Neural Information Processing Systems*. 2014, pp. 1–9.

[17] YU L., ZHANG W., WANG J., YU Y. SeqGAN: Sequence generative adversarial nets with policy gradient. In: *31st AAAI Conference on Artificial Intelligence*. 2017, pp. 2852–2858, 2017.

[18] TAVAKOLI S., HAJIBAGHERI A., SUKTHANKAR G. Learning social graph topologies using generative adversarial neural networks. In: *Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*. 2017, pp. 1–6.

[19] ZHU J., PARK T., EFROS A.A., AI B., BERKELEY U.C. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *International Conference on Computer Vision*. 2018, pp. 1–18.

[20] GONG N.Z., LIU B.N., GONG Z., LIU B. You are who you know and how you behave: Attribute inference attacks via users'social friends and behaviors. In: *Proceedings of the 25th USENIX Security Symposium*. 2016, pp. 980–995.

[21] JIA J., WANG B., ZHANG L., GONG N.Z. AttriInfer: Inferring user attributes in online social networks using Markov random fields. In: *26th International World Wide Web Conference*. 2017 pp. 1561–1569, 2017, doi: 10.1145/3038912.3052695.

[22] HOCHREITER S., SCHMIDHUBER J. Long short-term memory. *Neural Computation*. 1997, 9(5), pp. 1735–1780, doi: 10.1162/neco.1997.9.8.1735.

[23] GRAVES A., MOHAMED A., HINTON G. Speech recognition with deep recurrent neural networks. In: *International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 1–5.

[24] MESNIL G., HE X., DENG L., BENGIO Y. Investigation of recurrent – neural – network architectures and learning methods for spoken language understanding. In: *14th Annual Conference of the International Speech Communication Association*. 2013, pp. 1–5.

[25] SUTSKEVER I., VINYALS O., V. LE Q. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*. 2014, pp. 1–9.

[26] LE Q., MIKOLOV T., COM T. G. Distributed representations of sentences and documents. In: *Proceedings of the 31st International Conference on Machine Learning*. 2014, 32(2), pp. 1188–1196.

[27] GRAVES A., SCHMIDHUBER J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. In: *IEEE International Joint Conference on Neural Networks*. 2005, 18, pp. 602–610, doi: 10.1016/j.neunet.2005.06.042.

[28] DENTON E., SZLAM A., FERGUS R. Deep generative image models using a laplacian pyramid of adversarial networks. In: *Advances in Neural Information Processing Systems*. 2015, pp. 1–9.

[29] DONAHUE J., HENDRICKS L., GUADARRAMA S. Long-term recurrent convolutional networks for visual recognition and description. *Computer Vision and Pattern Recognition*. 2014, pp. 1-13.

[30] BA J.L., KIROS J.R., HINTON G.E. Layer normalization. 2015, https://arxiv.org/abs/1607.06450.

[31] JANG E., GU S., POOLE B. Categorical reparameterization with gumbel-softmax. In: *International Conference on Learning Representations*. 2017, pp. 1–13.