ANRL: *A*ttributed *N*etwork *R*epresentation *L*earning via Deep Neural Networks

Zhen Zhang[1,2], **Hongxia Yang**[3], **Jiajun Bu**[1], **Sheng Zhou**[1,2],
Pinggang Yu[1,2], **Jianwei Zhang**[3], **Martin Ester**[4], **Can Wang**[1*]
[1] College of Computer Science, Zhejiang University, China
[2]Alibaba-Zhejiang University Joint Institute of Frontier Technologies, China
[3] Alibaba Group, China
[4] Simon Fraser University, Canada
{zhen_zhang, bjj, zhousheng_zju, pgyu, canw}@zju.edu.cn,
{yang.yhx, zhangjianwei.zjw}@alibaba-inc.com, ester@cs.sfu.ca

Abstract

Network representation learning (RL) aims to transform the nodes in a network into low-dimensional vector spaces while preserving the inherent properties of the network. Though network RL has been intensively studied, most existing works focus on either network structure or node attribute information. In this paper, we propose a novel framework, named ANRL, to incorporate both the network structure and node attribute information in a principled way. Specifically, we propose a neighbor enhancement autoencoder to model the node attribute information, which reconstructs its target neighbors instead of itself. To capture the network structure, attribute-aware skip-gram model is designed based on the attribute encoder to formulate the correlations between each node and its direct or indirect neighbors. We conduct extensive experiments on six real-world networks, including two social networks, two citation networks and two user behavior networks. The results empirically show that ANRL can achieve relatively significant gains in node classification and link prediction tasks.

1 Introduction

Networks are general data structures to explore and model complex systems in the real world, including social networks, academic networks and the World Wide Web, etc. In the era of big data, networks have been an important medium to efficiently store and access relational knowledge of interacting entities. Mining knowledge in networks has drawn continuous attention in both academia and industry, e.g., online advertisement targeting and recommendation. Most of these tasks require carefully designed models with a lot of expert efforts for feature engineering, while RL is an alternative for relative automatic feature representation. Equipped with RL, knowledge discovery in networks, such as clustering [Narayanan *et al.*, 2007], link prediction [Lü and Zhou, 2011] and classification [Kazienko and Kajdanowicz, 2012], can be greatly facilitated by learning in low-dimensional vector spaces.

Related works in network RL can be traced back to graph based dimensional reduction methods, such as Locally Linear Embedding (LLE) [Roweis and Saul, 2000] and Laplacian Eigenmap (LE) [Belkin and Niyogi, 2003]. Both LLE and LE maintain the local structure in data space by constructing a nearest neighbor graph. To keep connected nodes closer to each other in the representation space, corresponding eigenvectors of the affinity graph are calculated as its representations. A major issue with these methods is that they are difficult to scale to large networks due to the high computational complexity in calculating eigenvectors. Inspired by the recent success of word2vec model [Mikolov *et al.*, 2013a; 2013b], many network structure based RL methods have been proposed and shown promising performance in various applications [Perozzi *et al.*, 2014; Cao *et al.*, 2015; Tang *et al.*, 2015b; Grover and Leskovec, 2016; Wang *et al.*, 2016]. However, node attribute information, which may play important roles in many applications, has not been paid much attention. Nodes affiliated with various attributes are commonly observed in real-world networks, termed as attributed information networks (AINs). For example, in Facebook social network, a user node is often associated with personalized profile information including age, gender, education as well as posted contents. Some recent efforts have explored AINs by integrating both network topology and node attribute information to learn better representations [Tang *et al.*, 2015a; Yang *et al.*, 2015; Pan *et al.*, 2016].

Representation learning in AINs is still at its early stage with rather limited capability due to the reasons that: (1) network topology and node attributes are two heterogeneous information sources, thus it is challenging to preserve their properties in a common vector space; (2) the observed net-

[*] Corresponding Author

work data is often incomplete and even noisy, which brings more difficulties for obtaining informative representations. To address the aforementioned challenges, we propose a unified framework, termed as ANRL, by jointly integrating network structure and node attribute information to learn robust representations in AINs. More specifically, we leverage the strong representation power of deep neural networks to capture the complex correlations of the two information sources, which is composed of a neighbor enhancement autoencoder and attribute-aware skip-gram model. To summarize, our main contributions are as follows:

- We propose a unified framework ANRL, which seamlessly integrates network structural proximity and node attributes affinity into low-dimensional representation spaces. To be more specific, we design a neighbor enhancement autoencoder, which can retain better similarity between data samples in the representation space. We also propose an attribute-aware skip-gram model to capture the structure correlations. These two components share connections to the encoder, which captures the node attributes as well as network structure information.
- We conduct extensively experiments on six datasets through two tasks: link prediction and node classification, and empirically demonstrate the effectiveness of the proposed model.

2 Related Work

Some earlier works [Roweis and Saul, 2000; Belkin and Niyogi, 2003] and other spectral methods target to preserve the local geometry structure of the data, and represent them with a lower dimension space. These approaches are parts of dimensionality reduction techniques and can be regarded as the pioneer of graph embedding. Recently, network representation learning has received increasing popularity in network analysis and they concentrate on embedding an existing network instead of constructing its affinity graph. Among them, DeepWalk [Perozzi *et al.*, 2014] performs truncated random walks to generate node sequences, which are treated as sentences and fed into skip-gram model to learn representations. Node2vec [Grover and Leskovec, 2016] extends DeepWalk by employing breadth-first (BFS) and depth-first (DFS) graph searches to explore diverse neighborhoods. Instead of performing random walks, LINE [Tang *et al.*, 2015b] optimizes both first order and second order graph proximities. Later, GraRep [Cao *et al.*, 2015] proposes to capture k-th order relational information for graph representation. SDNE [Wang *et al.*, 2016] incorporates graph structure into deep autoencoder to preserve the highly non-linear first order and second order proximity.

Attributed information networks are ubiquitous in many domains. It is promising to achieve better representations by including both network structure and node attributes information. Some existing algorithms have investigated the possibility of jointly embedding these two information sources into a unified space. For example, TADW [Yang *et al.*, 2015] incorporates DeepWalk and associated text features into the matrix factorization framework. PTE [Tang *et al.*, 2015a] utilizes label information and different levels of word co-occurrence information to generate predictive text representations. TriDNR [Pan *et al.*, 2016] uses information from three parties including node structure, node content, and node labels (if available) to jointly learn node representations. Although the above mentioned approaches indeed incorporate node attributes information into representations, they are specifically designed for text data and not suitable for many other types of features (e.g., continuous numerical features).

More recently, several feature type independent representation learning algorithms have been proposed to further enhance the performance via unsupervised or semi-supervised manner, which can handle all kinds of feature types and capture structural proximity as well as attribute affinity [Huang *et al.*, 2017; Liao *et al.*, 2017; Rossi *et al.*, 2018]. AANE [Huang *et al.*, 2017] is a distributed embedding approach that jointly learns node representations by decomposing attribute affinity matrix and penalizing the embedding difference between connected nodes with network lasso regularization. Planetoid [Yang *et al.*, 2016] develops both transductive and inductive methods to jointly predict the class label and neighborhood context in the graph. SNE [Liao *et al.*, 2017] generates embeddings by leveraging an end-to-end neural network model to capture the complex interrelations between network structure and node attribute information. Another semi-supervised learning framework SEANO [Liang *et al.*, 2018] takes the input form the aggregation of input sample attributes and its average neighborhood attributes to mitigate the negative effect of outliers in the representation learning procedure.

There also has been some efforts exploring representation learning in the heterogeneous information networks. Metapath2vec [Dong *et al.*, 2017] utilizes meta-path based random walks to generate heterogeneous node sequences and employs a heterogeneous skip-gram model to learn node representations. [Li *et al.*, 2017] proposes a model that can handle the representation learning in a dynamic environment instead of static networks. [Wang *et al.*, 2017] study the problem of representation learning in signed information networks. We leave these possible extensions as future work.

3 Proposed Model

3.1 Notations and Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an attributed information network, where $\mathcal{V}$ denotes the set of n nodes and $\mathcal{E}$ represents the set of edges. $\mathbf{X} \in \mathbb{R}^{n \times m}$ is a matrix that encodes all node attributes information, and $\mathbf{x}_i$ describes the attributes associated with node i. We formally define the attributed information network representation learning as follows:

Definition 3.1 *Given a network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, we aims to represent each node $i \in \mathcal{V}$ as a low-dimensional vector $\mathbf{y}_i$ by learning a mapping function $f : v_i \mapsto \boldsymbol{y_i} \in \mathbb{R}^d$, where $d \ll |\mathcal{V}|$ and the mapping function f preserves not only network structure but also node attribute proximity.*

3.2 Neighbor Enhancement Autoencoder

To encode node's attribute information, we design a neighbor enhancement autoencoder model to facilitate the noise-resilient representation learning procedure. Similarly, the neighbor enhancement autoencoder consists of an encoder and a decoder, while we aim to reconstruct its target neighbors instead of the node itself. It is worth noting that the proposed model degenerates to the traditional autoencoder when our target neighbor is the input node itself. More specifically, for the node v_i with its feature vector $\mathbf{x}_i$ and the target neighbors function $T(\cdot)$, the hidden representation of each layer is defined as follows:

$$\mathbf{y}_i^{(1)} = \sigma(\mathbf{W}^{(1)}\mathbf{x}_i + \mathbf{b}^{(1)}),$$
$$\mathbf{y}_i^{(k)} = \sigma(\mathbf{W}^{(k)}\mathbf{y}_i^{(k-1)} + \mathbf{b}^{(k)}),\ k = 2, ..., K, \qquad (1)$$

where K denotes the number of layers for the encoder and decoder. $\sigma(\cdot)$ represents the possible activation functions such as ReLU, sigmod or tanh. $\mathbf{W}^{(k)}$ and $\mathbf{b}^{(k)}$ are the transformation matrix and bias vector in the k-th layer, respectively. Our goal is to minimize the following autoencoder loss function:

$$\mathcal{L}_{ae} = \sum_{i=1}^{n} \|\hat{\mathbf{x}}_i - T(v_i)\|_2^2, \qquad (2)$$

where $\hat{\mathbf{x}}_i$ is the reconstruction output of decoder and $T(v_i)$ returns the target neighbors of v_i. $T(\cdot)$ incorporates prior knowledge into the model and can be adopted by the following two formulations:

- **Weighted Average Neighbor**. For a given node v_i, the target neighbors can be calculated as corresponding weighted average neighborhood. That is to say, $T(v_i) = \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} w_{ij}\mathbf{x}_j$, where $\mathcal{N}(i)$ is the neighbors of node v_i in the network and $\mathbf{x}_j$ is the attributes associated with node v_j. $w_{ij} > 0$ for weighted networks and $w_{ij} = 1$ for unweighted networks.
- **Elementwise Median Neighbor**. Similarly to weighted average neighbor, the elementwise median neighbor of node v_i is defined as: $T(v_i) = \tilde{\mathbf{x}}_i = [\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_m]$, where $\tilde{x}_k$ is the median value of its relevant neighborhood feature vectors at the k-th dimension, i.e., $\tilde{x}_k = \mathrm{Median}(w_{i1}\mathbf{x}_{1k}, w_{i2}\mathbf{x}_{2k}, \cdots, w_{i|\mathcal{N}(i)|}\mathbf{x}_{|\mathcal{N}(i)|k})$. $\mathrm{Medain}(\cdot)$ returns the median value of its input.

Our approach possesses more advantages than traditional autoencoders in retaining better proximity among nodes. Intuitively, the obtained representations are more robust to variations, since it constrains closely located nodes to have similar representations by forcing them to reconstruct the similar target neighbors. Thus, it captures both node attributes and local network structure information. In addition, the proposed neighbor enhancement autoencoder model is a general framework that can be applied to autoencoder variants such as denoising autoencoder and variational autoencoder.

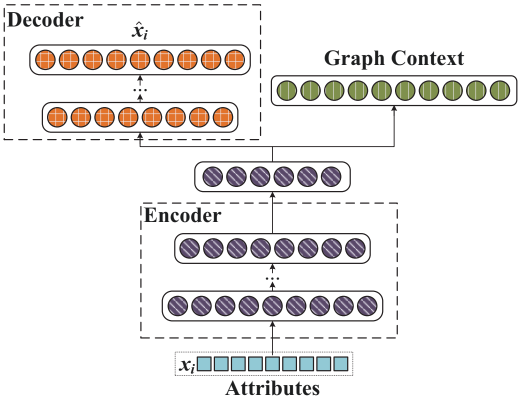

Figure 1: The architecture of the proposed ANRL model.

3.3 Attribute-aware Skip-gram Model

To formulate the network structure information, skip-gram model has been widely adopted in recent works [Perozzi *et al.*, 2014; Grover and Leskovec, 2016], which assumes nodes with similar context should be similar in latent semantic space. Based on that, we propose an attribute-aware skip-gram model to incorporate attribute information for more smooth representations. Specifically, the objective function minimizes the following log probability of skip-gram model by giving current node v_i with its associated attributes $\mathbf{x}_i$ for all random walk contexts $c \in C$:

$$\mathcal{L}_{sg} = -\sum_{i=1}^{n}\sum_{c\in C}\sum_{-b\le j\le b, j\neq 0} \log p(v_{i+j}|\mathbf{x}_i), \qquad (3)$$

where v_{i+j} is the node context in the generated random sequence and b is the window size. The conditional probability of $p(v_{i+j}|\mathbf{x}_i)$ is the likelihood of the target context given the node attributes and we formally define $p(v_{i+j}|\mathbf{x}_i)$ as:

$$p(v_{i+j}|\mathbf{x}_i) = \frac{\exp(\mathbf{v}_{i+j}^{'\mathrm{T}} f(\mathbf{x}_i))}{\sum_{v=1}^{n}\exp(\mathbf{v}_v^{'\mathrm{T}} f(\mathbf{x}_i))}, \qquad (4)$$

where $\mathbf{x}_i$ is the attribute information associated with node v_i and $f(\cdot)$ can be arbitrary attribute encoder function, e.g., CNN for image data and RNN for sequential data. $\mathbf{v}_i^{'}$ is the corresponding representations when node v_i is treated as "context" node.

It models not only node attributes but also global structure information. Directly optimizing Equation (4) is computationally expensive, which requires the summation over the entire set of nodes when computing the conditional probability of $p(v_{i+j}|\boldsymbol{x}_i)$. We adopt the negative sampling approach proposed in [Mikolov *et al.*, 2013b] that samples multiple negative samples according to some noisy distributions. In details, for a specific node-context pair (v_i, v_{i+j}), we have the following objective:

$$\log\sigma(\mathbf{v}_{i+j}^{'\mathrm{T}} f(\mathbf{x}_i)) + \sum_{s=1}^{|\mathrm{neg}|} \mathbb{E}_{v_n\sim P_n(v)}[\log\sigma(-\mathbf{v}_n^{'\mathrm{T}} f(\mathbf{x}_i))], \ (5)$$

where $\sigma(x) = 1/(1+\exp(-x))$ is the sigmoid function and $|\mathrm{neg}|$ is the number of negative samples. We set $P_n(v) \propto d_v^{3/4}$ as suggested in [Mikolov *et al.*, 2013b], where d_v is the degree of node v.

3.4 ANRL Model: A Joint Optimization Framework

In this subsection, we present the ANRL model to jointly utilize network structure and node attributes information to learn their latent representations. As illustrated in Figure 1, the overall architecture consists of two coupled modules, i.e., neighbor enhancement autoencoder and attribute-aware skip-gram model. The encoder transforms the input attributes into a low-dimensional vector space and extends out two output branches. The left output branch is a decoder which reconstructs the target neighbors of its input samples. The right output branch predicts the associated graph context of the given inputs. These two components are tightly interconnected as they share the first several layers. Through this way, the final representation of $\mathbf{y}_i^K$ captures the node attributes as well as network structure information.

The objective function of the joint ANRL model is formulated as the weighted combination of $\mathcal{L}_{sg}$ and $\mathcal{L}_{ae}$ as defined in Equations (2) and (3):

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{sg} + \alpha\mathcal{L}_{ae} + \beta\mathcal{L}_{reg} \qquad (6)\\ &= -\sum_{i=1}^{n}\sum_{c\in C}\sum_{-b\leq j\leq b, j\neq 0} \log\frac{\exp(\mathbf{u}_{i+j}^{\mathrm{T}}\mathbf{y}_i^{(K)})}{\sum_{v=1}^{n}\exp(\mathbf{u}_v^{\mathrm{T}}\mathbf{y}_i^{(K)})}\\ &+ \alpha\sum_{i=1}^{n}\|\hat{\mathbf{x}}_i - T(v_i)\|_2^2 + \frac{\beta}{2}\sum_{k=1}^{K}(\|\mathbf{W}^{(k)}\|_F^2 + \|\hat{\mathbf{W}}^{(k)}\|_F^2),\end{aligned}$$

where n is the total number of nodes, C is the set of node sequence generated by random walks and b is the window size. $\boldsymbol{x}_i$ represents node v_i's feature vector and $\boldsymbol{y}_i^{(K)}$ is the representation for node v_i after encoding with K layers; $\boldsymbol{W}^{(k)}, \hat{\boldsymbol{W}}^{(k)}$ are weight matrices for encoder and decoder respectively in the k-th layer. $\boldsymbol{U}$ is the weight matrix for graph context prediction and $\boldsymbol{u}_v$ corresponds to the v-th column in $\boldsymbol{U}$. α is the hyper parameter to balance the loss of autoencoder module and skip-gram module. β is the ℓ_2 norm regularizer coefficient.

In this way, ANRL preserves node attributes, local network structure and global network structure information in a unified framework. It is worth noting that the function $f(\cdot)$ of attribute-aware skip-gram module is exactly the encoder part of the autoencoder module, which transforms the node attributes information into representation space $\boldsymbol{y}_i^{(K)}$. As a result, the network structure and node attributes information will jointly affect $\boldsymbol{y}_i^{(K)}$. Furthermore, we only use one output layer to capture the graph context information for simplicity and it can be easily extended to multiple non-linear transformation layers.

To minimize $\mathcal{L}$, we adopt the stochastic gradient algorithm for optimizing Equation (6). We iteratively optimize these two coupled components until the model converges. All model parameters are denoted as $\mathbf{\Theta}$ and the learning algorithm is summarized in Algorithm 1.

4 Experiments

In this section, we conduct extensive experiments to verify the superiority of the proposed ANRL through comparing with several state-of-the-art methods on multiple real-world datasets.

4.1 Datasets

We summarize statistics of the six datasets in Table 1 with more descriptions as follows:

Algorithm 1 Joint ANRL Learning Framework

```
Input: graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, window size $b$, walks per vertex $\gamma$, walk length $t$, trade-off parameters $\alpha$, regularizer coefficient $\beta$, embedding size $d$
Output: node representations $\mathbf{Y} \in \mathbb{R}^{|\mathcal{V}|\times d}$
Construct node context corpus $C$ by starting $\gamma$ times of random walks with length $t$ at each node
Construct target neighbors for each node by function $T(\cdot)$
Random initialization for all parameters set $\mathbf{\Theta}$
while not converged do
    Sample a mini-batch of nodes with its context
    Compute the gradient of $\nabla\mathcal{L}_{ae}$ based on Equation (2)
    Update autoencoder module parameters
    Compute the gradient of $\nabla\mathcal{L}_{sg}$ based on Equation (5)
    Update skip-gram module parameters
end while
Obtain representations $\mathbf{Y} = \mathbf{Y}^{(K)}$ based on Equation (1)
```

Datasets	$\#\vert\mathcal{V}\vert$	$\#\vert\mathcal{E}\vert$	$\#\vert Attr\vert$	$\#\vert L\vert$
Facebook	4,039	88,234	1,283	-
UNC	18,163	766,800	2,789	-
UniID	23,377	29,188	3,924	-
Citeseer	3,312	4,714	3,703	6
Pubmed	19,717	44,338	500	3
Fraud Detection	40,386	1609,569	97	2

Table 1: Statistics of the datasets. '-' indicates unknown labels.

- **Social Network.** Facebook[1] [Leskovec and Mcauley, 2012] and UNC [Traud *et al.*, 2012] datasets are two typical social networks used in [Grover and Leskovec, 2016; Liao *et al.*, 2017]. Nodes represent users and edges represent friendship relations.
- **Citation Network.** Citeseer and Pubmed [2] which are used in [Yang *et al.*, 2016] consist of bibliography publication data. The edge represents that each paper may cite or be cited by other papers. The publications are classified into one of the following six classes: Agents, AI, DB, IR, ML, HCI in Citeseer and one of the three classes (i.e., "Diabetes Mellitus Experimental", "Diabetes Mellitus Type 1", "Diabetes Mellitus Type 2") in Pubmed.
- **User behavior Network.** We also employ two real-world user behavior datasets named UniID and Fraud Detection provided by Alibaba Group. For UniID dataset, the nodes in the network represent the identifiers for physical devices and the edges indicate the observed co-occurrence of two identifiers in the same user behavior records. Fraud Detection dataset includes cookies associated with attributes and their interactions with sellers. To get the homogeneous cookie graph, we project this bipartite graph onto cookie nodes, i.e., we connect two cookies only if they have at least five common seller nodes. We want to identify whether those cookies are suspicious or not.

[1] https://snap.stanford.edu/data/

[2] http://linqs.cs.umd.edu/projects/projects/lbc

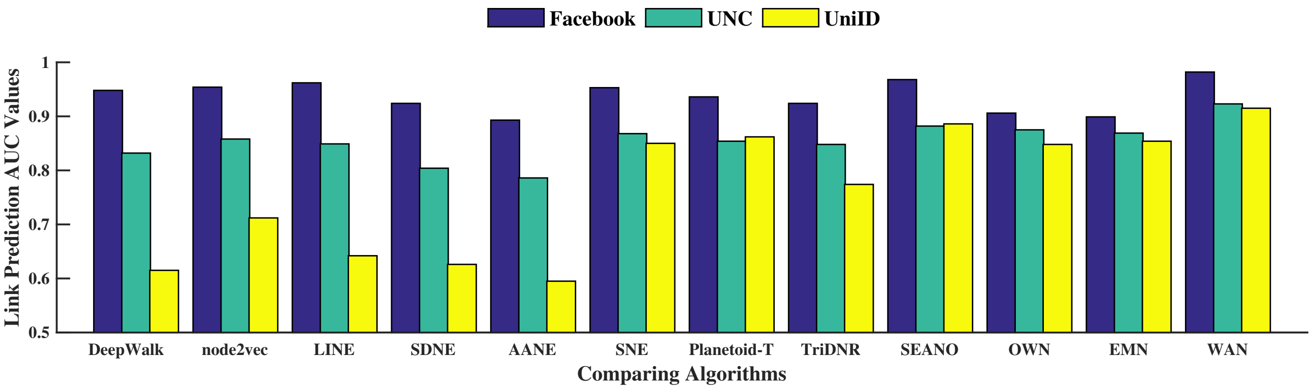

Figure 2: Link prediction performance comparisons of different algorithms on the Facebook, UNC and UniID. The y-axis represents the AUC value of each method while the x-axis shows the name of different methods. Note that we omit the prefix of the proposed ANRL variants.

4.2 Competitors

We compare ANRL with several state-of-the-art network RL algorithms that can be divided as the following groups :

1. **Attribute-only**: The first group of algorithms consider node attributes information only, which are used to verify the effectiveness of attributes in node classification task. We choose SVM and autoencoder as our baseline algorithms.
2. **Structure-only**: This group of baselines leverage network structure information only and ignore the node attributes. DeepWalk [Perozzi *et al.*, 2014] and node2vec [Grover and Leskovec, 2016] use truncated random walks to generate node sequences, then they are fed into skip-gram model to obtain the latent node representations. LINE [Tang *et al.*, 2015b] and SDNE [Wang *et al.*, 2016] exploits the network structure's first-order proximity and second-order proximity.
3. **Attribute + Structure:** Methods of this group try to preserve node attribute and network structure proximity, which are competitive competitors. We consider AANE [Huang *et al.*, 2017], SNE [Liao *et al.*, 2017], Planetoid-T [Yang *et al.*, 2016], TriDNR [Pan *et al.*, 2016] and SEANO [Liang *et al.*, 2018] as our compared algorithms. More detailed descriptions can be found in Section 2.
4. **ANRL Variants:** To analyze the performance of our proposed model, we consider three variants: ANRL-WAN which use **W**eighted **A**verage **N**eighbor function to construct its target neighbors, ANRL-EMN which takes **E**lementwise **M**edian **N**eighbor function to generate its target neighbors as defined in Section 3.2 and ANRL-OWN which reconstructs itself (i.e., **OWN**) as traditional autoencoder does.

For all baselines, we used the implementation released by the original authors. The parameters for baselines are tuned to be optimal. We set the embedding size d as 64 in Fraud Detection dataset and 128 for the remaining datasets. For LINE, we concatenate both first-order and second-order as our final representations. Furthermore, we set window size b as 10, walk length l as 80, walks per node γ as 10, negative samples as 5. For ANRL, the number of layers and dimensions for left output branch are shown in Table 2 and we only use one layer in right output branch.

Datasets	Number of neurons in each layer
Facebook	1283–500–128–500–1283
UNC	2789–1000–500–128–500–1000–2789
UniID	3924–1000–500–128–500–1000–3924
Citeseer	3703–1000–500–128–500–1000–3703
Pubmed	500–200–128–200–500
Fraud Detection	97–64–97

Table 2: Detailed network layer structure information.

4.3 Link Prediction

In this subsection, we evaluate the ability of node representations in reconstructing the network structure via link prediction. We generate the labeled dataset of edges as many other works do [Grover and Leskovec, 2016; Wang *et al.*, 2016], which randomly holds out 50% existing links as positive instances; For negative instances, we randomly sample an equal number of non-existing links. Then, we use the residual network to train the embedding models. After having obtained the representations for each node, we use these representations to perform link prediction task in the labeled edge dataset. Specifically, we rank both positive and negative instances according to the cosine similarity function. To judge the ranking quality, we employ the AUC to evaluate the ranking list and a higher value indicates a better performance.

We perform link prediction task on three unlabeled datasets (i.e., Facebook, UNC, UniID datasets) and the results is shown in Figure 2. We summarize the following observations and analyses:

- A general observation we can draw from the result is that our method achieves relatively significant improvements in AUC over the baselines in both three datasets. For instance, our method achieve about 3.5% AUC improvement over the best performance baseline in UNC dataset.
- Since DeepWalk, node2vec, LINE and SDNE only utilize network structure information, their performance

Datasets	Citeseer		Pubmed		Fraud Detection	
Evaluation	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
SVM	0.667	0.626	0.856	0.855	0.725	0.719
autoencoder	0.630	0.565	0.792	0.800	0.732	0.726
DeepWalk	0.583	0.534	0.809	0.795	0.509	0.464
node2vec	0.607	0.561	0.815	0.802	0.571	0.519
LINE	0.542	0.512	0.766	0.749	0.659	0.654
SDNE	0.569	0.528	0.699	0.677	0.662	0.656
AANE	0.579	0.541	0.784	0.765	0.654	0.643
SNE	0.632	0.615	0.803	0.797	0.662	0.654
Planetoid-T	0.656	0.594	0.851	0.847	0.692	0.693
TriDNR	0.633	0.587	0.843	0.824	0.686	0.685
SEANO	0.713	0.662	0.859	0.848	0.703	0.704
ANRL-OWN	0.652	0.606	0.842	0.845	0.724	0.720
ANRL-EMN	0.716	0.668	0.865	0.867	0.733	0.731
ANRL-WAN	**0.729**	**0.673**	**0.876**	**0.871**	**0.759**	**0.755**

Table 3: Node classification results on Citeseer, Pubmed and Fraud Detection datasets. We use **blue** to highlight wins.

is relative worse when the network is extremely sparse such as UniID dataset. Interestingly, we notice that node2vec achieves the best results among this group baselines, which is mainly because that node2vec can explore diverse network structure information via biased random walks.

- Being consistent with previous works' findings, we also observe that incorporating both node attributes and network structure information improves the link prediction performance, which reflects the value of attributes. Among them, AANE and TriDNR only exploit first-order network structure information and fails to capture sufficient information for link prediction. However, the remaining algorithms gain superior performance by performing random walks on the network to capture higher-order proximity information.
- More specifically, ANRL exploits node attributes information (via both two modules), global network structure information (via attribute-aware skip-gram module) and local network structure information (via neighbor enhancement autoencoder module). We argue that one major reason for the performance lift is because our model takes both local and global network structure information into consideration.

4.4 Node Classification

Similarly to previous works [Perozzi *et al.*, 2014; Grover and Leskovec, 2016], we report the performances of node classification. We randomly select 20 samples from each class and treat them as the labeled data to train semi-supervised baselines following the same strategy in [Yang *et al.*, 2016]. After having obtained the node representations, we randomly sample 30% labeled nodes to train a SVM classifier and the rest of the nodes are used to test performances. We repeat this process 10 times, and report the average performances in terms of both Macro-F1 and Micro-F1. The detailed results are shown in Table 3 and to summarize, we have the following observations:

- ANRL-WAN achieves the best performance among all the methods for all settings. The classification performance is followed by other structure and attribute based methods, and then followed by structured based methods with several exceptions. This further justifies the usefulness of attributes, and properly modeling them can lead to better representations with significant performance gains.
- It is worth noting that traditional attribute-only methods outperform most of structure-only approaches, because network structure alone provides very limited useful information (compared to node attributes) for node classification task. Yet, we observe that autoencoder is a little weaker than SVM, which indicates the dimension reduction procedure may lose some useful information.
- In particular, SEANO outperforms several state-of-the-art attribute and structure preserving methods by aggregating additional neighborhood attributes into representation learning phase. AANE performs poorly in this group of competitors, which involves the decomposition operation of attribute affinity matrix. This significant degenerates the performance of AANE, because we usually do not know the similarities between each node and need to compute them based on certain similarity measure.
- Finally, ANRL-WAN and ANRL-EMN perform better than ANRL-OWN and most of the other baselines algorithms. As can be seen from the table, ANRL-WAN outperforms ANRL-OWN with a significant improvement, which shows the effectiveness of our proposed neighbor enhancement autoencoder. Furthermore, our attribute-aware skip-gram module and neighbor enhancement autoencoder module force the latent representations more smooth and robust, which are important properties in many tasks.

5 Conclusions

In this paper, we investigate the representation learning in attributed information networks. Accordingly, we design a coupled deep neural network model, which incorporates both node attributes and network structure information into the embedding. To further address the structure proximity and attribute affinity preserving, we develop a neighbor enhancement autoencoder and attribute-aware skip-gram model to exploit the complex interrelations between structural information and attributes. Experimental results on several real-world datasets show that the proposed ANRL outperforms representative state-of-the-art embedding approaches.

Acknowledgments

This work is supported by Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Zhejiang Provincial Key Research and Development Plan (Grant no. 2017C01012), the National Natural Science Foundation of China (No. 61173185, 61173186) and the National Key Technology R&D Program of China (No. 2012BAI34B01, 2014BAK15B02).

References

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

[Cao *et al.*, 2015] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *CIKM*, pages 891–900, 2015.

[Dong *et al.*, 2017] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. Metapath2vec: Scalable representation learning for heterogeneous networks. In *SIGKDD*, pages 135–144, 2017.

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

[Huang *et al.*, 2017] Xiao Huang, Jundong Li, and Xia Hu. Accelerated attributed network embedding. In *SIAM*, pages 633–641, 2017.

[Kazienko and Kajdanowicz, 2012] Przemyslaw Kazienko and Tomasz Kajdanowicz. Label-dependent node classification in the network. *Neurocomputing*, 75(1):199–209, 2012.

[Leskovec and Mcauley, 2012] Jure Leskovec and Julian J. Mcauley. Learning to discover social circles in ego networks. In *NIPS*, pages 539–547. 2012.

[Li *et al.*, 2017] Jundong Li, Harsh Dani, Xia Hu, Jiliang Tang, Yi Chang, and Huan Liu. Attributed network embedding for learning in a dynamic environment. In *CIKM*, pages 387–396, 2017.

[Liang *et al.*, 2018] Jiongqian Liang, Peter Jacobs, Jiankai Sun, and Srinivasan Parthasarathy. Semi-supervised embedding in attributed networks with outliers. In *SDM*, 2018.

[Liao *et al.*, 2017] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Attributed social network embedding. *arXiv preprint arXiv:1705.04969*, 2017.

[Lü and Zhou, 2011] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[Mikolov *et al.*, 2013a] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119. 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013.

[Narayanan *et al.*, 2007] Hariharan Narayanan, Mikhail Belkin, and Partha Niyogi. On the relation between low density separation, spectral clustering and graph cuts. In *NIPS*, pages 1025–1032, 2007.

[Pan *et al.*, 2016] Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang. Tri-party deep network representation. In *IJCAI*, pages 1895–1901, 2016.

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

[Rossi *et al.*, 2018] Ryan A. Rossi, Rong Zhou, and Nesreen K. Ahmed. Deep inductive network representation learning. In *WWW BigNet*, page 8, 2018.

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

[Tang *et al.*, 2015a] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *SIGKDD*, pages 1165–1174, 2015.

[Tang *et al.*, 2015b] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.

[Traud *et al.*, 2012] Amanda L Traud, Peter J Mucha, and Mason A Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

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

[Wang *et al.*, 2017] Suhang Wang, Charu Aggarwal, Jiliang Tang, and Huan Liu. Attributed signed network embedding. In *CIKM*, pages 137–146, 2017.

[Yang *et al.*, 2015] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *IJCAI*, pages 2111–2117, 2015.

[Yang *et al.*, 2016] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, pages 40–48, 2016.