Robust Attribute and Structure Preserving Graph Embedding

Bhagya Hettige(✉), Weiqing Wang, Yuan-Fang Li, and Wray Buntine

Monash University, Melbourne, Australia
{bhagya.hettige,teresa.wang,yuanfang.li,wray.buntine}@monash.edu

Abstract. Graph embedding methods are useful for a wide range of graph analysis tasks including link prediction and node classification. Most graph embedding methods learn only the topological structure of graphs. Nevertheless, it has been shown that the incorporation of node attributes is beneficial in improving the expressive power of node embeddings. However, real-world graphs are often noisy in terms of structure and/or attributes (missing and/or erroneous edges/attributes). Most existing graph embedding methods are susceptible to this noise, as they do not consider uncertainty during the modelling process. In this paper, we introduce RASE, a **R**obust **A**ttribute and **S**tructure preserving graph **E**mbedding model. RASE is a novel graph representation learning model which effectively preserves both graph structure and node attributes through a unified loss function. To be robust, RASE uses a denoising attribute auto-encoder to deal with node attribute noise, and models uncertainty in the embedding space as Gaussians to cope with graph structure noise. We evaluate the performance of RASE through an extensive experimental study on various real-world datasets. Results demonstrate that RASE outperforms state-of-the-art embedding methods on multiple graph analysis tasks and is robust to both structure and attribute noise.

Keywords: Robust graph embedding · Node classification · Link prediction

1 Introduction

Much real-world data can be naturally delineated as graphs, e.g. citation networks [1,7,16], social-media networks [2,18] and language networks [16]. Graph embedding methods [6,7,13,16] have been proposed as an effective way of learning low-dimensional representations for nodes to enable down-stream machine learning tasks, such as link prediction and node classification, on these complex graph data. Most existing graph embedding methods learn node embeddings from graph topological structure only [6,13,16,17]. However, nodes in a graph usually have supplementary attribute information which can be utilized in graph embedding along with the graph structure to produce more meaningful node embeddings [7,11,15,21].

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12085, pp. 593–606, 2020.
https://doi.org/10.1007/978-3-030-47436-2_45

Graphs constructed from the real-world data are usually non-deterministic and ambiguous [14], manifested by uncertain and ambiguous edges and/or node attributes. For example, most knowledge graphs follow the "Open World Assumption" [14] (i.e. the unobserved edges are unknown instead of untrue), so that graph structures are far from complete and many edges are missing. Also, much attribute information is abstracted from free text (e.g. users' post on social media) and is usually imprecise or ambiguous due to the limitations in data sources or abstraction tools. We term this non-deterministic and ambiguous phenomenon in graph structure and node attributes as **"structure noise"** and **"attribute noise"** respectively.

A great challenge that the existing graph embedding methods face when incorporating both graph structure and node attributes, is the noise prevalent in these two aspects which can mislead the embedding technique to result in learning invalid latent information. Recently, several studies have been proposed to model the uncertainty present in graph data [1,8,11,20]. Most of these work, including VGAE [11], and Graph2Gauss [1], focuses on modelling the uncertainty of the node embeddings by representing the nodes with a probabilistic distribution in the embedding space. Since these studies attempts to preserve the observable graph structural proximity by measuring the distance between probability distribution embeddings, uncertainty modelling of these methods can only capture structure noise. Therefore, they do not explicitly account for the node attribute noise which is common in the real-world graphs.

In this work, we introduce RASE, a novel graph embedding framework to address the aforementioned challenges. RASE learns robust node representations via carefully-designed strategies, exploiting both graph structure and node attributes simultaneously. Attribute noise is modelled with a denoising attribute auto-encoder to maintain the discreteness and sparseness of textual data by introducing a noise in the input through a binomial distribution. Structure noise is modelled in the latent layer by modelling the embeddings as Gaussian distributions. To preserve the transitivity in the embedding space with a linear computational cost, 2-Wasserstein distance is used as the similarity measure between the distributions in Gaussian space. Extensive experiments have been conducted on five different real-world datasets. The experimental results show that our method significantly outperforms state-of-the-art methods in generating effective embeddings for node classification and link prediction. Moreover, we introduce a novel experimental setting to simulate random structure noise and random attribute noise to demonstrate the robustness of our model in embedding noisy graphs.

2 Related Work

There are three lines of effort most related to this work: structure-preserving graph embedding, attributed graph embedding and noise modelled graph embedding.

Structure-Preserving Graph Embedding: These embedding methods attempt to conserve observable graph structure properties in the embedding space. LINE [16] learns from structural closeness considering first- and second-order proximity. DeepWalk [13] and node2vec [6] learn node embeddings from random walk sequences with a technique similar to Skip-Gram [12]. DVNE [20] uses an auto-encoder architecture to encode and reconstruct the graph structure. All these algorithms focus on graph structure only.

Attributed Graph Embedding: Recent studies [1,7,8,11,15,19,21] show that the incorporation of node attributes along with graph structure produces better node embeddings. TADW [19] incorporates text attributes and graph structure with low-rank matrix factorization. GraphSAGE [7] is a CNN-based technique that samples and aggregates neighbouring node attributes. Graph2Gauss [1] finds the neighbours in each hop up to a pre-defined number of hops which is space inefficient. Also, it uses node attributes for embedding initialization and does not explicitly preserve attributes when learning embeddings. VGAE [11] is a graph convolution network (GCN) method, which aggregates neighbouring attributes. In most studies, node attributes are only used for embedding initialization, but not during model training. DANE [4] proposes a deep non-linear architecture to preserve both aspects.

Noise Modelled Graph Embedding: Most of the existing graph embedding methods represent nodes as point vectors in the embedding space, ignoring the uncertainty of the embeddings. In contrast, Graph2Gauss [1], VGAE [11], DVNE [20] and GLACE [8] capture the uncertainty of graph structure by learning node embeddings as Gaussian distributions. DVNE [20] proposes to measure distributional distance using the Wasserstein metric as it preserves transitivity. A recent study [3] learns a discrete probability distribution on the graph edges. However, these works ignore the modelling of the uncertainty of node attributes.

3 Methodology

Problem Formulation. Let $G = (\mathcal{V}, E, \mathbf{X})$ be an **attributed graph**, where $\mathcal{V}$ is the set of nodes, E is the set of edges in which each ordered pair of nodes $(i, j) \in E$ is associated with a weight $w_{ij} > 0$ for edge from i to j, and $\mathbf{X}_{|\mathcal{V}| \times D}$ is the node attribute matrix, where $\mathbf{x}_i \in \mathbf{X}$ is a D-dimensional attribute vector of node i. We learn to embed each node $i \in \mathcal{V}$ as a low-dimensional Gaussian distribution $\mathbf{z}_i = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$[1], where $\boldsymbol{\mu}_i \in \mathbb{R}^L$, $\boldsymbol{\sigma}_i^2 \in \mathbb{R}^{L \times L}$ with the embedding dimension $L \ll |\mathcal{V}|$, D. The learning goal is such that, nodes that are closer in the graph and have similar attributes are closer in the embedding space, and node embeddings are robust to structure noise and attribute noise.

[1] We learn diagonal covariance vector, $\boldsymbol{\sigma}_i^2 \in \mathbb{R}^L$, instead of a covariance matrix to reduce the number of parameters to learn.

3.1 RASE Architecture

Figure 1 shows the architecture of RASE which is an end-to-end embedding framework that learns from both node attributes and graph structure, with two main components: *Node Attribute Learning* and *Graph Structure Learning*. To deal with attribute noise, RASE corrupts node attributes by introducing a random noise ε_i sampled from a binomial distribution, which are then projected to a low-dimensional representation $\mathbf{u}_i$. RASE takes this $\mathbf{u}_i$ as input and simultaneously performs node attribute learning and graph structure learning. By reconstructing the node attributes from $\mathbf{u}_i$, the model preserves attributes (with Euclidean distance to preserve transitivity) while being robust to attribute noise. RASE models uncertainty of the graph structure noise by learning Gaussian embeddings and capturing neighbourhood information measured with Wasserstein metric to preserve transitivity property in the embedding space.

Node Attribute Learning. We learn node attributes in an unsupervised manner. To deal with noisy attributes, we slightly corrupt the attribute vectors using a random noise. In most real-world graphs, node attributes can be very sparse, since they are either tf-idf vectors of textual features or one-hot vectors of categorical features. A Gaussian noise would substantially change a sparse attribute vector and would not characterise the trends observed in the real data. Thus, we draw noise from a binomial distribution as a masking noise but it still depicts the original data trends. Accordingly, we inject some impurity to the original node attribute vector $\mathbf{x}_i \in \mathbb{R}^D$ by sampling a random binary noise vector $\varepsilon_i \in \{0, 1\}^D$ from a binomial distribution B with D (i.e. attribute vector dimension) trials and p success probability. We set $p \in (0.90, 0.98)$ to ensure that the noise is small and its introduction does not change the data trends. We produce the corrupted attribute vector $\mathbf{x}'_i \in \mathbb{R}^D$ by performing Hadamard product: $\mathbf{x}'_i = \mathbf{x}_i \otimes \varepsilon_i$.

The corrupted attribute vector is transformed into an intermediate representation $\mathbf{u}_i \in \mathbb{R}^m$ where m is a reduced vector dimension using an encoding

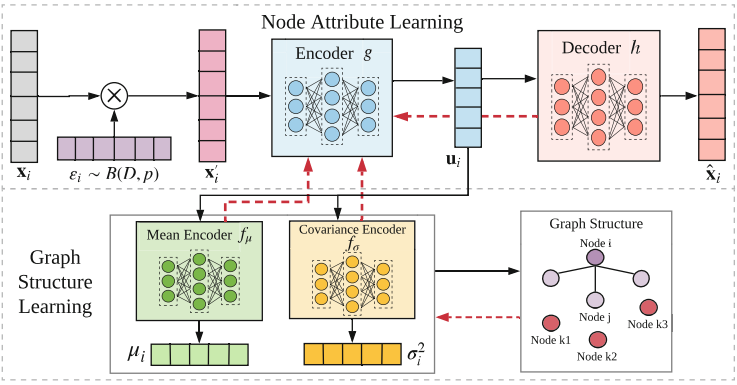

Fig. 1. RASE architecture.

transformation function, $g : \mathbb{R}^D \rightarrow \mathbb{R}^m$. Subsequently, this intermediate vector is fed as input to a decoder, $h : \mathbb{R}^m \rightarrow \mathbb{R}^D$, to reconstruct the attribute vector $\hat{\mathbf{x}}_i \in \mathbb{R}^D$. Note that, these encoder and decoder functions can easily be implemented with MLP layers or sophisticated GCN layers [11] and to capture the non-linearity in data we can have deep neural networks. But we observe that MLP architecture is more simple and efficient, hence scalable on large-scale graphs. We define the attribute reconstruction loss as the Euclidean distance between the original and reconstructed attribute vectors:

$$\mathcal{L}_a = \sum_{i \in \mathcal{V}} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 \tag{1}$$

L1 regularization has been adopted as we have sparse attribute vectors constructed from textual data. By minimizing the attribute reconstruction loss we encourage the *encoder* g to generate robust latent representations, $\mathbf{u}_i$, which are used as inputs to the *Graph Structure Learning* component.

Graph Structure Learning. We use the intermediate vector, $\mathbf{u}_i$, from the auto-encoder in the *Node Attribute Learning* component, as it encodes attribute latent relationships between nodes. We define two parallel transformations to model a node's embedding as a Gaussian distribution to account for structural uncertainty (due to noise), i.e. f_μ and f_σ, that learn the mean vector $\boldsymbol{\mu}_i$ and the diagonal covariance vector $\boldsymbol{\sigma}_i^2$ of $\mathbf{u}_i$ respectively. To obtain positive $\boldsymbol{\sigma}_i^2$ for interpretable uncertainty we choose activation function at the output layer accordingly. Thus, the final latent representation of node i is $\mathbf{z}_i = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$, where:

$$\boldsymbol{\mu}_i = f_\mu(\mathbf{u}_i) \;\text{ and }\; \boldsymbol{\sigma}_i^2 = f_\Sigma(\mathbf{u}_i) \tag{2}$$

To preserve the structural proximity of nodes in the graph, we assume that the nodes which are connected with a higher edge weight are more likely to be similar and we attempt to pull the embeddings of these nodes closer in the embedding space. We define the prior probability for connected nodes as $\hat{P}(i,j) = \frac{w_{ij}}{\Sigma_{(k,l)\in E} w_{kl}}$ where w_{ij} is the weight of the edge $(i,j) \in E$. Since RASE's node embeddings are Gaussians, we choose a probability distance metric to compute the distance between nodes. Thus, motivated by DVNE [20], to preserve transitivity property in the embedding space, we choose the Wasserstein distance: 2-nd moment (W_2). This metric allows to discover specific relations between nodes based on their semantic relations and similarities by leveraging the geometric properties of the embedding space. As a result, when we model the explicit local neighbourhood edges, implicit global neighbourhood proximity can be modelled due to triangle inequality property. We define $\delta(\mathbf{z}_i, \mathbf{z}_j)$ as the W_2 distance for our node embeddings, i and j. Modelling only the diagonal covariance vectors results in $\boldsymbol{\sigma}_i^2 \boldsymbol{\sigma}_j^2 = \boldsymbol{\sigma}_j^2 \boldsymbol{\sigma}_i^2$. Hence, W_2 computation [5] simplifies to:

$$\delta(\mathbf{z}_i, \mathbf{z}_j) = W_2(\mathbf{z}_i, \mathbf{z}_j) = (\| \boldsymbol{\mu}_i - \boldsymbol{\mu}_j \|_2^2 + \| \boldsymbol{\sigma}_i - \boldsymbol{\sigma}_j \|_F^2)^{1/2} \tag{3}$$

The likelihood of an edge between nodes i and j is defined as the similarity of the two node embeddings [16]:

$$P(i,j) = Sigmoid(-\delta(\mathbf{z}_i, \mathbf{z}_j))) = \frac{1}{1+\exp\left(\delta(\mathbf{z}_i, \mathbf{z}_j)\right)} \tag{4}$$

We minimize the distance between the prior and the observed probability distributions for the edges to preserve the node proximity in the embedding space. Since $\hat{P}$ and P are discrete probability distributions, we define structural loss using KL divergence:

$$\mathcal{L}_s = D_{KL}(\hat{P}||P) = \sum_{(i,j)\in E} \hat{P}(i,j) \log\left(\frac{\hat{P}(i,j)}{P(i,j)}\right) \propto -\sum_{(i,j)\in E} w_{ij} \log P(i,j) \tag{5}$$

For regularization of $\mathcal{L}_s$, instead of regularizing mean and covariance functions separately, RASE uses the strategy similar to [10] minimizing KL divergence between the learned Gaussian representation and the standard normal distribution. Thus, it will ensure that the final latent space will be closer to a standard Gaussian space other than pushing values in both mean vectors and variance vectors to be small. Different from RASE, Graph2Gauss [1] does not regularize the Gaussian functions. The regularization for node i is:

$$D_{KL}(\mathbf{z}_i||\mathcal{N}(\mathbf{0},\mathbf{1})) = \frac{1}{2}\Big(\boldsymbol{\sigma}_i^2 + \boldsymbol{\mu}_i^2 - \ln(\boldsymbol{\sigma}_i{}^2) - 1\Big) \tag{6}$$

By minimizing the overall structural loss function, we attempt to construct an embedding space where nodes that are similar in terms of graph structure are also similar in the embedding space and robust to noisy graph structure.

Unified Training and Optimization. To jointly preserve node attributes and graph structure, we define a unified loss function by combining Eq. 1 and Eq. 5 with hyperparameter $\alpha > 0$. For simplicity, we omit the regularization terms in the two components of RASE in the overall loss function to be minimized:

$$\mathcal{L} = \alpha\mathcal{L}_a + \mathcal{L}_s \quad = \alpha \sum_{i\in\mathcal{V}} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 - \sum_{(i,j)\in E} w_{ij} \log P(i,j) \tag{7}$$

For large graphs, this unified loss function is computationally expensive, since it has to compute the attribute reconstruction loss ($\mathcal{L}_a$) for all the nodes and the structural loss ($\mathcal{L}_s$) for all the edges. To optimize $\mathcal{L}_a$, we sample only a batch of nodes in each epoch. To optimize $\mathcal{L}_s$, we employ the negative sampling approach [12] and sample K negative edges for each edge in the training batch.

Table 1. Statistics of the real-world graphs.

Dataset	$\|V\|$	$\|E\|$	D	#Labels
Social media networks				
BlogCatalog	5,196	369,435	8,189	6
Flickr	7,535	239,738	12,047	9
Citation networks				
Cora	2,995	8,416	2,879	7
Citeseer	4,230	5,358	602	6
Pubmed	18,230	79,612	500	3

Therefore, for each edge (i, j) in the batch, $b \subset E$, with the noise distribution, $P_n(v)$ for $v \in \mathcal{V}$, we can compute the structural loss as:

$$\log\ \sigma(-\delta(\mathbf{z}_i, \mathbf{z}_j)) + \sum_{n=1}^{K} \mathbb{E}_{v_n \sim P_n(v)} \log\ \sigma(\delta(\mathbf{z}_i, \mathbf{z}_{v_n})) \tag{8}$$

4 Experiments

We evaluate RASE against state-of-the-art baselines in several graph analysis tasks, node classification, link prediction and robustness on several public datasets. Source code for RASE is publicly available at https://github.com/bhagya-hettige/RASE.

4.1 Datasets

Social Media Networks [9] **(Table 1):** Nodes on these networks are users. The following relationships are used to construct the edges. Attributes on BlogCatalog and Flickr are constructed with keywords in users' blog description and users' predefined tags of interests, respectively. Node labels are users' interest topics on BlogCatalog and groups users joined in Flickr. **Citation Networks** [1] **(Table 1):** Nodes denote papers and edges represent citation relations. We use tf-idf word vectors of the paper's abstract as node attributes. Each paper is assigned a label based on the topic of the paper.

4.2 Compared Algorithms

We compare RASE to several state-of-the-art graph embedding methods: structure-based non-attributed embedding methods (node2vec, LINE and DVNE); attributed embedding methods (GraphSAGE, VGAE and Graph2Gauss); and uncertainty modelling embedding methods (DVNE, VGAE and Graph2Gauss).

node2vec [6] is a random walk based node embedding method that maximizes the likelihood of preserving nodes' neighbourhood. **LINE** [16] preserves first- and second-order proximity. We report results on a concatenated representation of the two proximities (as suggested). **DVNE** [20] learns Gaussian distributions in the Wasserstein space from plain graphs. **GraphSAGE** [7] is an attributed graph embedding method which learns by sampling and aggregating features of local neighbourhoods. We use its unsupervised version, since all other methods are unsupervised. **VGAE** [11] is an attributed GCN-based embedding method which implements an auto-encoder model with Gaussian node embeddings. **Graph2Gauss (G2G)** [1] is an attributed embedding method which represents each node as a Gaussian and preserves the graph structure based on a ranking scheme of multiple neighbouring hops. In addition, we evaluate task performance on node **attributes** as input features instead of learning node embeddings, for down-stream machine learning tasks.

RASE is our full model which jointly preserves node attribute and graph structure, and is robust to noise in real-world graphs. We also consider a non-robust version, **RASE(¬*R*)**, for an ablation study. **RASE(¬*R*)** does not model attribute noise and learns point vectors, thus also ignoring structural uncertainty.

4.3 Experimental Settings

For all the models that learn point vectors, we set $L = 128$ as the embedding dimension. For a fair evaluation, we set $L = 64$ in methods learning probability distributions, including ours, so that the parameters learned per node is still 128 ($\boldsymbol{\mu}_i \in \mathbb{R}^{64}$ and $\boldsymbol{\sigma}_i^2 \in \mathbb{R}^{64}$). The other parameters for baselines are referred from the papers and tuned to be optimal. α is tuned to be optimal using grid search on a validation set. We report the results averaged over 10 trials.

4.4 Node Classification

In this task, each method learns the embeddings in an unsupervised manner, and a logistic regression (LR) classifier is trained on these embeddings to classify each node into their associated class label. We randomly sample different percentages of labeled nodes (i.e. $1\%, 2\%, \ldots, 10\%$) from the graph as training set for the classifier, and use the rest for evaluation. We report micro- and macro-F1 scores which have been widely used in multi-class classification evaluation [16]. We only present micro-F1 in Fig. 2, and a similar trend is observed in macro-F1.

Based on the results in Fig. 2, we can see that RASE consistently outperforms all the baselines in all the datasets with all the training ratios. Furthermore, in all five datasets, RASE has demonstrated a larger improvement margin to the baselines when only smaller numbers of nodes are used for training, e.g., a 174.9% improvement over best performing baseline in Flickr at 1% labeled nodes. This performance improvement is due to the attribute preserving component, which learns meaningful latent representations from node attributes. Moreover, denoising the attributes in this process also helps our model to deal with scarce data which is common in the real-world graphs. Also, our proposed structure

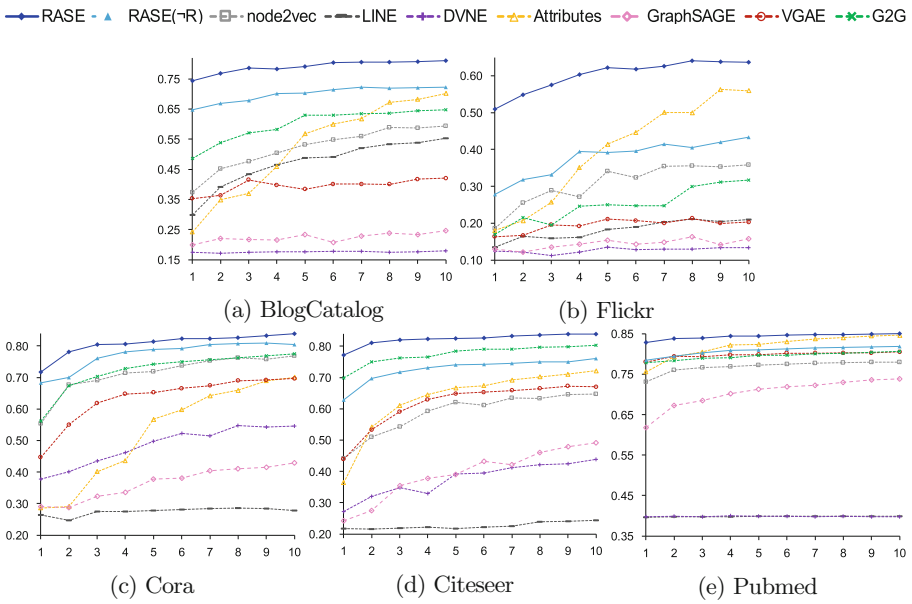

Fig. 2. Node classification performance measured by micro-F1 score (y-axis) in terms of percentage of labelled nodes (x-axis). RASE's improvements are statistically significant for $p < 0.01$ by a paired t-test.

learning method has captured useful local and global node similarities (due to the transitivity-preserving property in W_2 metric).

Overall, RASE($\neg R$) manifests superiority among the non-probabilistic methods (i.e. node2vec, LINE and GraphSAGE) consistently outperforming them in all datasets. Interestingly, on BlogCatalog, Flickr and Cora, RASE($\neg R$) also substantially outperforms the probabilistic models DVNE, VGAE and G2G. This emphasizes the effectiveness of our attribute preservation and structure learning method, even in the absence of uncertainty modelling.

4.5 Link Prediction

This task aims to predict future links using the graph structure and attributes. We randomly select 20% edges and an equal number of non-edges, and combine the two as the test set. The remaining 80% are used for training. Then, the node embeddings are used to compute the similarity between each test node pair, which is regarded as the likelihood of a link's existence between them. In Gaussian embedding methods, we use negative Wasserstein distance (RASE, DVNE) and negative KL divergence (G2G) to rank the node pairs [1,20]. For other methods, we use dot product similarity of node embeddings. We measure AUC and AP scores [1,11]. For brevity reasons, we only present citation networks in Table 2, and the trend is similar in the social media networks.

RASE clearly outperforms the state-of-the-art methods by a significant margin in all the graphs, demonstrating the effectiveness of our model in capturing structural and attribute information. RASE outperforms RASE($\neg R$), showing that accounting for structure and attribute noise collectively is beneficial. This is also validated by the performance gain of the uncertainty modelling methods, VGAE and G2G, over the non-robust RASE($\neg R$). Moreover, the methods that learn from graph structure only (i.e. node2vec, LINE and DVNE) are significantly outperformed by the attributed embedding methods (i.e. RASE, RASE($\neg R$), GraphSAGE, VGAE and G2G). RASE($\neg R$) is the best performing model among the non-probabilistic methods (i.e. node2vec, LINE and GraphSAGE), demonstrating that it has learnt meaningful structural similarities between nodes along with node attributes.

Table 2. Link prediction performance.

Algorithm	Cora		Citeseer		Pubmed	
	AUC	AP	AUC	AP	AUC	AP
node2vec	79.11	77.99	79.91	82.08	91.18	91.49
LINE	79.12	78.91	71.20	72.11	75.32	76.81
DVNE	65.73	70.33	68.16	73.42	50.66	50.78
Attributes	88.06	83.66	81.53	75.60	82.98	77.71
GraphSAGE	81.76	83.19	83.33	85.38	89.43	90.90
VGAE	93.53	95.33	95.46	96.47	96.11	96.09
G2G	95.92	95.82	96.28	96.54	95.75	95.65
RASE($\neg R$)	95.42	96.18	95.60	96.25	94.54	93.84
RASE	**96.88**	**96.82**	**97.82**	**97.69**	**96.40**	**96.21**

4.6 Robustness

We evaluate RASE and state-of-the-art baselines to see how they can deal with noise in graphs. In this section, we introduce a novel evaluation task to assess the robustness of graph embeddings to *random structure and attribute noise*. We inject some random noise into the graphs by intentionally corrupting the graph structure and node attributes. This experiment is conducted on all datasets. We report the results on Citeseer, since all the datasets demonstrate similar trends.

Structural Noise: We corrupt the graph structure by hiding randomly selected edges 50% (to mimic *missing edges* which we also use as the test set) and randomly adding some non-existing edges (edges not in the original graph to mimic *erroneous edges*). We vary the percentage of noisy edges added to the graph from 0%–50%, and observe AUC and AP decline with the increasing noise in link prediction task. The results are presented in Table 3.

From Table 3, we see that RASE performs the best in all the structural noise percentages, showing that it is robust to noisy graph structures. In addition to

this, with the increase in noise ratio from 0% to 50%, RASE's AUC degradation is only 3.8%. Also, RASE outperforms its non-robust version, RASE($\neg R$), which shows that the proposed uncertainty modelling technique to mitigate structure noise is effective. In contrast, though DVNE, VGAE and G2G also model uncertainty in the embeddings, their performance degradation is quite significant (7.4%, 6.5% and 14.3% in AUC respectively) when the noise ratio is increased from 0% to 50%. VGAE is based on GCN, which aggregates the neighbouring attributes into a convex embedding. Thus, it is heavily affected by noisy neighbours, as errors get further exaggerated. The hop-based structural ranking in G2G is sensitive to false neighbourhoods. Furthermore, the square-exponential loss function used for pair-wise ranking in both G2G and DVNE does not have a fixed margin and pushes the distance of the negative edges to infinity with an exponentially decreasing force [1]. Hence, these methods are highly sensitive to erroneous and missing edges. In contrast, RASE is mildly affected due to its carefully designed structural loss function and the extra information learned from the neighbourhood via the transitivity property of W_2 metric.

Table 3. Link prediction performance in Citeseer with structural noise.

Noisy edge %	0%		10%		20%		30%		40%		50%	
	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
node2vec	73.0	76.8	66.7	70.6	62.8	66.3	57.6	61.3	56.7	60.2	56.2	59.1
LINE	53.1	50.0	52.2	49.2	52.2	48.8	52.6	49.5	51.1	48.4	51.2	48.6
DVNE	57.9	59.4	56.0	57.1	52.9	55.2	55.5	56.6	53.2	55.4	53.6	55.4
GraphSAGE	75.5	78.2	73.9	76.5	73.1	75.4	73.0	74.9	71.8	73.4	71.3	72.1
VGAE	90.2	92.5	88.6	91.1	86.9	89.8	85.8	88.9	86.0	88.7	84.3	87.5
G2G	91.3	91.9	84.8	87.2	79.7	83.2	77.6	81.8	76.4	80.2	78.2	81.4
RASE($\neg R$)	90.1	91.2	90.2	90.6	89.7	89.3	88.8	89.4	87.0	87.7	86.1	86.5
RASE	**96.0**	**95.9**	**94.9**	**94.9**	**94.3**	**94.5**	**93.5**	**93.5**	**92.6**	**93.2**	**92.2**	**92.5**

Attribute Noise: To evaluate the robustness of the methods to random attribute noise, we corrupt the node attribute vectors randomly. Then, we assess node classification performance of the learned embeddings on these corrupted graphs. Specifically, we sample a masking noise from a binomial distribution with D (i.e. attribute dimension) trials and $p = 0.70$ probability, and perform Hadamard product with attribute vectors of some randomly selected nodes. Thus, approximately 30% of the attributes for each selected node are corrupted. We also vary the percentage of nodes corrupted from 0%–50% to investigate the micro- and macro-F1 decline. Since we are interested in evaluating the attribute robustness of the embedding methods, we experiment with attributed embedding methods only. The results are reported in Table 4.

Table 4 shows that RASE is robust to random node attribute noise, having the highest macro- and micro-F1 scores steadily across all noisy node percentages. Moreover, RASE only shows a 3.3% degradation in micro-F1, when we

increase the proportion of corrupted nodes from 0% to 50%. The small degradation can be attributed to the node attribute denoising step. RASE also outperforms its non-robust counterpart, showing the effectiveness of attribute noise modelling component. GraphSAGE shows a poorer node classification performance when compared to others, which shows that noisy attributes has misled the model to learn inexact node embeddings. Negative effect of noisy attributes of neighbouring nodes in GCN's aggregation step causes the lower performance of VGAE when its performance is compared against RASE and G2G. Overall, G2G shows a modest micro-F1 decline (i.e. 4.9% from 0% to 50%), since the Gaussian node embeddings have captured the attribute uncertainty via the variance terms.

4.7 Visualization

We visualize the node embeddings produced by RASE on Cora and Blogcat. We train RASE with $L = 128$ and $\boldsymbol{\mu}$ vectors are projected to two dimensions using t-SNE (Fig. 3). RASE produces an adequate visualization with tightly clustered nodes of the same class label with clearly visible boundaries.

Table 4. Node classification performance in Citeseer with 30% attribute noise.

Corrupted node %	micro (mi)- and macro(ma)- F1 score											
	0%		10%		20%		30%		40%		50%	
	mi	ma	mi	ma	mi	ma	mi	ma	mi	ma	mi	ma
GraphSAGE	42.9	13.2	42.3	14.8	41.8	13.3	41.7	16.2	41.6	16.2	41.3	16.2
VGAE	77.9	77.0	77.0	77.0	76.1	76.1	75.2	75.3	74.0	74.2	73.8	73.7
G2G	84.1	84.2	82.3	82.4	81.5	81.5	79.8	79.9	79.0	78.9	79.9	79.9
RASE($\neg R$)	82.0	82.2	81.2	81.4	80.2	80.5	79.9	79.9	78.1	78.2	77.7	77.6
RASE	**85.8**	**85.8**	**85.0**	**85.0**	**83.8**	**83.8**	**83.5**	**83.4**	**82.5**	**82.4**	**83.0**	**82.9**

(a) Cora (b) Blogcat

Fig. 3. Visualization of RASE embeddings. Colour of a node denotes its class label. (Color figure online)

4.8 Parameter Sensitivity Analysis

We study the sensitivity of attribute reconstruction learning weight (α) and embedding dimension (L) in RASE. Figure 4 shows micro-F1 score for node classification task on Citeseer averaged over 10 trials. In general, $\alpha > 0$ shows better performance than $\alpha = 0$, demonstrating the positive effect of learning from node attributes. The impact of attribute preservation is optimal near $\alpha = 10$ in RASE and $\alpha = 40$ in RASE($\neg R$). Also, RASE performs increasingly better when the embedding dimension L is increased, since larger dimensions can encode more meaningful latent information. When $L \geq 32$, RASE and RASE($\neg R$) are already complex enough to handle the data and further increments are less helpful.

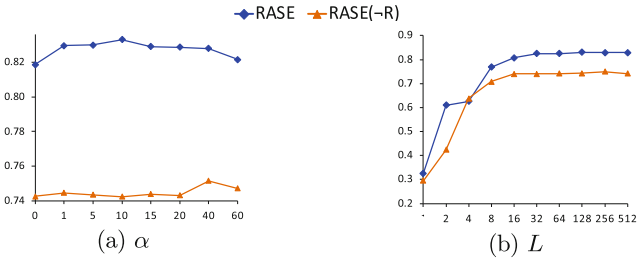

Fig. 4. Parameter sensitivity analysis. Micro-F1 in node classification on Citeseer.

5 Conclusion

In this work, we present RASE, an end-to-end embedding framework for attributed graphs. RASE learns robust node embeddings by preserving both graph structure and node attributes considering random structure and attribute noise. RASE has been evaluated w.r.t. several state-of-the-art methods in different graph analysis tasks, and the results demonstrate that RASE significantly outperforms all the evaluated baselines.

Acknowledgements. This work has been supported by the Monash Institute of Medical Engineering (MIME), Australia.

References

1. Bojchevski, A., Günnemann, S.: Deep Gaussian embedding of attributed graphs: unsupervised inductive learning via ranking. In: ICLR (2018)
2. Chen, H., Yin, H., Wang, W., Wang, H., Nguyen, Q.V.H., Li, X.: PME: projected metric embedding on heterogeneous networks for link prediction. In: SIGKDD (2018)
3. Franceschi, L., Niepert, M., Pontil, M., He, X.: Learning discrete structures for graph neural networks. In: ICML (2019)

4. Gao, H., Huang, H.: Deep attributed network embedding. In: IJCAI (2018)
5. Givens, C.R., Shortt, R.M., et al.: A class of Wasserstein metrics for probability distributions. Mich. Math. J. **31**(2), 231–240 (1984)
6. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: ACM SIGKDD (2016)
7. Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NIPS (2017)
8. Hettige, B., Li, Y.-F., Wang, W., Buntine, W.: Gaussian embedding of large-scale attributed graphs. In: Borovica-Gajic, R., Qi, J., Wang, W. (eds.) ADC 2020. LNCS, vol. 12008, pp. 134–146. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-39469-1_11
9. Huang, X., Li, J., Hu, X.: Label informed attributed network embedding. In: ACM WSDM (2017)
10. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
11. Kipf, T.N., Welling, M.: Variational graph auto-encoders. In: NIPS Workshop on Bayesian Deep Learning (2016)
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS (2013)
13. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: ACM SIGKDD (2014)
14. Shi, B., Weninger, T.: Open-world knowledge graph completion. In: AAAI (2018)
15. Sun, G., Zhang, X.: A novel framework for node/edge attributed graph embedding. In: Yang, Q., Zhou, Z.-H., Gong, Z., Zhang, M.-L., Huang, S.-J. (eds.) PAKDD 2019. LNCS (LNAI), vol. 11441, pp. 169–182. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-16142-2_14
16. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: large-scale information network embedding. In: WWW (2015)
17. Wang, Q., Yin, H., Wang, W., Huang, Z., Guo, G., Nguyen, Q.V.H.: Multi-hop path queries over knowledge graphs with neural memory networks. In: Li, G., Yang, J., Gama, J., Natwichai, J., Tong, Y. (eds.) DASFAA 2019. LNCS, vol. 11446, pp. 777–794. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-18576-3_46
18. Wang, W., Yin, H., Du, X., Hua, W., Li, Y., Nguyen, Q.V.H.: Online user representation learning across heterogeneous social networks. In: SIGIR (2019)
19. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.Y.: Network representation learning with rich text information. In: IJCAI (2015)
20. Zhu, D., Cui, P., Wang, D., Zhu, W.: Deep variational network embedding in Wasserstein space. In: ACM SIGKDD (2018)
21. Zhu, D., Dai, X., Yang, K., Chen, J., He, Y.: PCANE: preserving context attributes for network embedding. In: Yang, Q., Zhou, Z.-H., Gong, Z., Zhang, M.-L., Huang, S.-J. (eds.) PAKDD 2019. LNCS (LNAI), vol. 11441, pp. 156–168. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-16142-2_13