Denoising Node Embeddings

Removed for Review
removed for review

Abstract

Node embedding (NE) algorithms capture features of graph's nodes and represent them in a low dimensional vector space. Graphs are inherently noisy structures, which might reduce the learned representations quality. We propose a novel approach using denoising autoencoders to reduce noise in the learned representation of nodes. Experiments with three state-of-the-art NE algorithms show that our approach effectively reduces noise in a link prediction task.

1 Introduction

The increasing volume of available information has made graph processing a hard and costly task and real world problems are often noisy, as well as their graph's representation. NE algorithms can be used to address these problems by transforming graph's nodes into a low dimension vector space while preserving the graph information [1, 2]. NE algorithms have been proposed to improve graph representation [3, 4, 5] and their effectiveness is closely related to the quality of the node's representations learned by them. Additionally, NE algorithms can introduce more noise while they learn node's representations due to the randomness in the generation of walks or permutations, thus preventing the effective use of all information in graphs to address real world problems.

In this work, we propose a novel approach to reduce noise in the node's representation produced by three algorithms: DeepWalk [3], Node2Vec [4] and NBNE [5]. We use a denoising autoencoder to learn a more effective node feature representation. We evaluate our approach using eight datasets in the traditional link prediction task and show that noise removal effectively improves the quality of node's representation in seven datasets and presents no statistical difference on the last one.

2 Denoising Architecture

Denoising autoencoders (dA) are used for unsupervised learning of latent features [6, 7] and the training process reconstructs an original input from a corrupted version of it [8, 9]. In our work, we use state-of-the-art NE algorithms to generate initial versions of nodes' representations and afterwards apply denoising autoencoders to make a denoised version of them. We changed the typical denoising autoencoder's structure by using *tanh* (instead of *sigmoid*) as the activation function and added noise to the input instead of dropping out part of its features. By using *tanh* we preserve the representation's value learned by NE algorithms between $[-1, 1]$ and by adding noise we simulate real world noise that is introduced during the process of representation learning.

The denoising steps of our approach are: (i) first corrupt the original input x by adding noises, $\tilde{x} := x + noises$, (ii) map the corrupted input into a latent space using an encoder multi layer perceptron (MLP), generating a more efficient and robust hidden representation of the input, $h := f(\tilde{x}) = tanh(W_f^n ... tanh(W_f^0 \tilde{x} + b_f^0) + b_f^n)$, where W_f^i are weight matrices and b_f^i are bias vectors, both learned during the training process, (iii) reconstruct the original input from the hidden representation using a decoder MLP, $\hat{x} := g(h) = tanh(W_g^n ... tanh(W_g^0 h + b_g^0) + b_g^n)$, where W_g^i are weight matrices, and b_g^i are bias vectors. Note that processing time increases linearly as the number of nodes rises.

3 Experiments and Results

We evaluate our denoising approach extrinsicaly by performing link prediction produced by: DeepWalk (DW), Node2Vec (NV) and NBNE. Each graph represents a network: (i) Amazon product co-purchasing network (*amazon0302*) [10]; (ii) Astro-Physics collaboration network (*astro*) [11]; (iii) High energy physics phenomenology citation graph (*cit-HepPh*) [12]; (iv), (v) and (vi) Deezer music streaming - three countries (*Deezer-HU, Deezer-HR, Deezer-RO)* [13]; (vii) Enron email network (*email-Enron*) [14]; (viii) Facebook (*facebook*) [15].

We predict which links between nodes will be created based on already known links. Five-fold cross-validation [16] with training and test sets of selected graph links was performed. We report effectiveness in terms of AUC [17], using cosine similarity (value between $[-1, 1]$) as binary classifier to reflect the intrinsic quality of the representations.

We start by doing a random search with 25 runs to find the best hyper-parameters for each baseline, only running our dA on top of the best one (as chosen by results on a validation set). We then do a grid search to tune our dA architecture on top of these embeddings. We use *Batch Normalization* and *Dropout* to reduce overfitting and our architecture uses an equal number of layers in the encoder and decoder MLPs, keeping the number of nodes constant through them, equal to the input NE size. We search through the hyper-parameters: $noise \sim [N(0, 0.05), N(0, 0.1), N(0, 0.2)]$, $dropout \in [0.1, 0.3, 0.5]$ and $num\_layers \in [2, 3]$.

From Table 1, we observe that our approach effectively reduces noise, improving the quality of node's representation in almost all cases. With exception of *amazon0302* (using DW or NV) and *cit-HelPh* (using DW), our denoising approach improved the representation quality, and consequently providing significant gains in terms of AUC. Also, in many cases our approach achieved significant improvements, specially in *Deezer*, and also in *email-Enron* using NBNE, where removing noise improved the result from **15.27** to **3.94**.

Table 1: Results are given in terms of 100−AUC for easier comparison. A t-test with p-value of 0.01 was performed for all pairs (NE, Ours) and statistically significant values are shown in bold.

Dataset	DW	Ours	NV	Ours	NBNE	Ours
amazon0302	1.22±0.01	1.1±0.14	0.94±0.02	0.97±0.05	1.75±0.02	**1.39±0.15**
astro	1.48±0.02	**1.28±0.04**	2.25±0.07	**1.35±0.04**	8.77±0.11	**2.42±0.04**
cit-HepPh	1.2±0.01	1.18±0.03	2.71±0.06	**1.45±0.02**	3.82±0.04	**2.13±0.04**
Deezer-HU	10.29±0.13	**7.96±0.09**	10.71±0.32	**8.24±0.08**	12.47±0.58	**9.32±0.18**
Deezer-HR	4.88±0.06	**4.19±0.04**	5.41±0.12	**4.15±0.04**	8.27±0.16	**4.86±0.07**
Deezer-RO	11.53±0.12	**8.78±0.42**	12.94±0.81	**8.84±0.22**	14.3±0.95	**11.3±0.27**
email-Enron	2.6±0.09	**2.35±0.16**	4.89±0.1	**2.37±0.1**	15.27±0.66	**3.94±0.24**
facebook	0.45±0.03	**0.34±0.02**	0.78±0.03	**0.56±0.07**	1.54±0.04	**1.3±0.11**

4 Conclusion and Future Work

Experimental results attest the effectiveness of our denoising approach in performing link prediction, with reductions in error of up to 74.19%. As future work, we intend to evaluate our approach in different network topologies, NE architectures, tasks, and also exploit dA in natural language processing with word embeddings.

References

[1] HongYun Cai, Vincent Zheng, and Kevin Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE TKDE*, 30(9):1616–1637, 2018.

[2] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowl.-Based Syst.*, 151:78–94, 2018.

[3] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of KDD'14*, pages 701–710, 2014.

[4] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of KDD'16*, pages 855–864, 2016.

[5] Tiago Pimentel, Adriano Veloso, and Nivio Ziviani. Fast node embeddings: Learning egocentric representations. In *Proceedings of ICLR'18 Workshop Track*, 2018.

[6] Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014.

[7] Cheng-Yuan Liou, Jau-Chi Huang, and Wen-Chie Yang. Modeling word perception using the elman network. *Neurocomputing*, 71(16-18):3150–3157, 2008.

[8] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

[9] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of ICML'08*, pages 1096–1103, 2008.

[10] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *TWEB*, 1(1):5, 2007.

[11] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *TKDD*, 1(1):2, 2007.

[12] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of KDD'05*, pages 177–187, 2005.

[13] Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles A. Sutton. GEMSEC: graph embedding with self clustering. *CoRR*, abs/1802.03997, 2018.

[14] Bryan Klimt and Yiming Yang. Introducing the Enron corpus. In *Proceedings of CEAS'04*, 2004.

[15] Julian J. McAuley and Jure Leskovec. Learning to discover social circles in ego networks. In *Proceedings of NIPS'12*, pages 548–556, 2012.

[16] Raj Jain. *The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling.* Wiley-Interscience, New York, 1991.

[17] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.