NÖHÜ Müh. Bilim. Derg. / NOHU J. Eng. Sci., 2022; 11(3), 477-483

Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences

Araştırma makalesi / Research article

www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

A high order proximity measure for linear network embedding

Ağ gömülümü için yüksek boyutlu yakınsaklık ölçüsü

Mustafa Coşkun[1,*]

[1] *Abdullah Gul University, Computer Engineering Department, 38080, Kayseri Turkey*

Abstract

Graph representationion learning (network embedding) is at the heart of network analytics techniques to reveal and examine the complex dependencies among nodes. Owing its importance, many computational methods have been proposed to solve a large volume of learning tasks on graphs, such as node classification, link prediction and clustering. Among various network embedding techniques, linear Matrix Factorization-based (MF) network embedding approaches have demonstrated to be very effective and efficient as they can be stated as singular value decomposition (SVD) problem, which can be efficiently solved by off-the-shelf eigen-solvers, such as Lanczos method. Despite the effectiveness of these linear methods, they rely on high order proximity measures, i.e., random walk restarts (RWR) and/or Katz, which have their own limitations, such as degree biasness, hyper-parameter dependency. In this paper, to alleviate the RWR and Katz depended high proximity usage in the linear embedding methods, we propose an algorithm that uses label propagation and shift-and-invert approach to resort RWR and Katz related problems. Testing our methods on real-networks for link prediction task, we show that our algorithm drastically improves link prediction performance of network embedding comparing against an embedding approach that uses RWR and Katz high order proximity measures.

Keywords: Graph representation learning, Node embedding, Linear embedding

Öz

Ağ gömülümü öğrenme problemi bir çok ağ analizi gerektiren problemin ifade ve çözümlenmesi için çok büyük önem arz etmektedir. Bu bağlamda, ağ içerisinde bulunan düğümlerin birbirleri ile olan gizli ilişkilerini açığa çıkarmak için, son yıllarda ağ gömülümü öğrenme problemi çokça çalışılmaktadır. Bu gizli ilişkinin açığa çıkarılması, bağlantı tahminleme, öbekleme ve sınıflandırma gibi öğreme problemlerinin daha iyi çözümlenmesinde kullanılmaktadır. Ağ gömülümünü öğrenmek için, farklı yaklaşım ve algoritmalar geliştirilmiş olsada, matris ayrışımı bazlı algoritmalar hızlı olmasından dolayı araştırmacılar tarafından büyük ilgi görmekteler. Matris ayraşım bazlı ağ gömülümü öğrenmede genel anlamı ile yüksek dereceli yakınlık ölçüleri kullanılmaktadır, örneğin random walk with restart (RWR) ve Katz ölçüleri. Ancak, bu ölçülerle yapılan ağ benzerlik ölçüleri matris ayrışımında sıfıra karşılık gelen eigenvectors (özvektörler) üretebilmektedir. Bu ise öğrenilen ağ gömülümün yanlış olmasına sebeb olmaktadır. Bu prolemi aşmak için, bu makalede shift-and-invert (kaydır ve tersini al) yaklaşımına dayanarak bir yaklaşım önerdik. Bağlantı tahimini baz problemi alarak, geliştirdiğimiz algoritmayı üç gerçek veride kullanık ve sonuçların var olan matris ayrışımlı algoritmasını bütün metrik değerlendirmelerinde var olan algoritmanın performansını ciddi miktarda artırdığını gözlemledik.

Anahtar kelimeler: Ağ gömülümü, Düğüm gömülümü, Lineer ağ gömülümü

1 Introduction

Networks (graphs) are ubiquitous in the real-world applications to represent the relationships (edges) among entities (nodes), such as social networks are used to represent the social relationship among people and protein-protein interaction networks are formed from genetic or physical associations among genes. To understand and examine the complex dependencies among the nodes, networks' association information might not be enough. Thus, we need to state and encode networks' topological structural information (association information) in a continues latent space, which could then be used in the off-the-shelf machine learning algorithms as features. There are at least two valid reasons to use the latent representations of a network instead of its plain representation: sparseness of networks and computational complexity of underlying machine learning algorithms.

Graph representation learning (network embedding) is an emerging research topic that is used to learn the structural representations of networks. Specifically, the representation learning aims at mapping the topological structure of the networks into a continuous lower-dimension so that off-the-shelf machine learning algorithms can readily be applied on continuous vector representation for various learning tasks, such as node classification and clustering. Here the basic premise behind the graph representation learning is to map a graph into a k-dimensional continues vector space, such that if two nodes are close in the network, they should be close to each other in the vector representation.

There are a plethora of papers and methods for network embedding, which we can broadly catagorize into three groups: Matrix Factorization (MF)-based, random walk-based and neural network-based [1]. Among these categories, random walk-based methods rely on computing

* Sorumlu yazar / Corresponding author, e-posta / e-mail: mustafa.coskun@agu.edu.tr (M. Çoşkun)
Geliş / Recieved: 25.06.2021 Kabul / Accepted: 22.04.2022 Yayımlanma / Published: 18.07.2022
doi: 10.28948/ngmuh.9574881

latent representation via random walk restart procedure, such as DeepWalk [2] and LINE [3], while neural network-based methods, such as GAE [4] and DGI [5] require additional attribute information which might not be available for every network. On the other hand, MF-based approaches have been soaring many research attentions due to their easy implementation and low cost computational complexity. As such, one of the well-known MF-based methods, HOPE [6], has been used for many learning tasks [1].

Despite the effectiveness of HOPE network embedding method [6], it computes an embedding using the idea of high order proximity preservation, which relies on personalized random walk (i.e., RWR) or Katz measures. However, these high order proximity measures do not take the ill-conditioning problem into account. More specifically, the very first step of RWR, degree normalization, can cause close-to-zero diagonal entries for low degree nodes which result in wrong matrix inversion, which is known as ill-conditioning problem [7] in computational mathematics.

In this paper, to circumvent the ill-conditioning problem introduced by HOPE method [6] in its high order proximity measure computation, we use the idea of shift-and-invert to take the ill-conditioning problem into account [7]. This way, we eliminate the adversely effects of the low degree nodes and define a new high order proximity measure, HOPE++, for computing network embeddings. To test our proposed method's performance against HOPE method, computed by RWR and Katz measures, we use the link prediction as a benchmark problem on three real-world datasets. Experimental results on these three datasets show that our approach drastically improves performance of HOPE method across all evaluation metrics: Accuracy, Area Under Precision Curve, Area Under ROC Curve and Macro-F1 scores, used in this paper.

2 Related work

We can trace back the original idea of network embedding to the early 2000s, including but not limited to Isomap [8] and Locally Linear Embedding [9]. These ideas revolve around the use of a linear system of equation solution and singular value decomposition (SVD) of a matrix, usually graph Laplacian, created by the adjacency matrix of a graph [10]. These methods aim at learning an embedding with the constraint: local manifold structure must be preserved.

More recently, with the advent of the large-scale graphs, a variety of scalable graph representation learning methods, [2,3,5,6,11,12] which rely on random walk procedure to capture the global properties of nodes in the graph have been proposed. While powerful, random walk-based methods suffer from known limitations, such as degree biasness [13,14].

In addition to random walk-based embedding approaches, inspired by the remarkable success of Deep Learning methods in the field of computer vision, neural network based approaches have been utilized for graph embedding [4,5,15]. These methods use both attribute of nodes and graph topology in their propagation step. As using many layers of neural network causes feature mix-matching problem [16], these methods limit their feature propagation step into 1-hop proximity. Recently, Coskun and Kuyuturk [17] show that usage of 2-hop proximity-based graph convolutional networks can improve performance link prediction. However, all the above methods do not consider the high-order proximity measure in their propagation. There exist high-order proximity-based propagation neural network methods, such as [18] however these methods still rely on random walk procedure.

In this paper, we present a simple yet very effective network embedding method that consider high-order proximity of a graft while it eliminates ill-conditioning problem introduced by random walk procedure. As our proposed method is a linear approach, it is computationally very efficient comparing to neural network-based approaches [4,5,17,18,19,20]. Furthermore, our approach does not require add-on information, such as node attributes as in neural network-based methods [4,5,17,18,19,20]. Last but not the least, our method eliminates adversely effects of low-degree nodes which has not been considered in other linear methods [2,3,6].

3 Material and methods

In this section, we first define the link prediction problem [21] in the context of graph representation learning and then we introduce our proposed network embedding approach to solve the link prediction problem.

3.1 Link prediction

The problem of link prediction can be loosely defined as follows: given a graph, predict the missing links among the nodes. More formally, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes set of n nodes in the graph and $\mathcal{E}$ reptesents the set of m edges among n nodes, the task is to predict some of the missing edges in graph. The link prediction problem has many real-world applications ranging from co-authorship prediction [13] to drug response prediction [14].

Earlier algorithms for solving link prediction problem have focused on local graph topological informations, such as commong neighbor, adamic-adar, and etc [21] while recent studies have shown that global graph structural information, which can be seen as toplogical graph representations and can be more informative to determine the missing links [13]. Very recently, graph representation learning algorithms have been applied to link prediction problem to circumvent the curse of dimensionality problem, high dimensionality problem [13] pronounced in global representations of the underlying graph [2,3,4,5,6].

These embedding techniques first map the graph structural information into a lower dimensional continues space, such that each node is represented by a k-dimensional vector. Then, for a given pair of nodes, their corresponding k-dimensional latent vector are fused to generate a feature vector for learning. To be more specific, the fused networks are used in training with their associated labels (1 if two nodes are connected in the graph and 0 otherwise). Finally, any supervised machine learning algoritm, such as Support Vector Machine (SVD), is used to train and test the existence of links.

3.2 Network embedding

Graph representation learning (Network Embedding) aims at learning a lower continues representation of each node in a given graph. More formally, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, network embedding maps the graph's structural information into an embedding matrix, $H \in \mathbb{R}^{n \times d}$, where $d \ll n$, such that if given two nodes i and j are "close" to each other in the graph, their representations h_i and h_j, respectively, should be "close" to each other.

The abstract term "close" is concreted by the different mean of proximity measures. For example, random walk-based embedding techniques [2,3] rely on random walk-based closeness global measures, such as random walk with restarts (RWR), while neural network-based approaches use more local closeness measures, see e.g., [4,5].

Despite the effectiveness of these methods, random walk-based approaches [2,3] suffer from random-walk related limitations, such as degree biasness [4], and Neural Network-based methods [4,5] are limited to small network as training time of these methods are extremely costly. On the other hand, MF-based approaches, such as singular value decomposition (SVD) and HOPE [6] enjoy efficient algorithms from numerical linear algebra literature, such as Lanczos [7].

Inspired by the effectiveness and efficiency of the MF-based approaches, in this paper, we use HOPE [6] algorithm as our baseline algorithm. In the following subsections, we first define HOPE algorithm [6] in a formal framework and then present our approach to overcome some limitations of HOPE algorithm, rooted in random walk-based closeness measure.

3.2.1 HOPE [6] Algorithm

Ou et. al., [6] has presented HOPE algorithm as one of the MF-based approaches. In essence, they obtain a graph representation, $\mathrm{H} \in \mathbb{R}^{\mathrm{n} \times \mathrm{d}}$, by using various form of closeness measures. To be more specific, they define following objective function to obtain the embedding matrix [6]:

$$min\|\boldsymbol{S} - \boldsymbol{H}\boldsymbol{H}^{\boldsymbol{T}}\|^2 \tag{1}$$

where $\boldsymbol{S}$ is a proximity matrix (encodes closeness of two nodes by using various proximity measures) and H denotes embedding matrix, which best represents $\boldsymbol{S}$ matrix in the latent space. Here, the most important part of the objective function is that how we construct the proximity matrix from which we learn the representations of nodes. To do so, Ou et. al., [6] use four well-known proximity measures: Common Neighbour (CN), Adamic-Adar (AA), RWR and Katz [8], respectively:

$$\boldsymbol{S}^{CN} = \boldsymbol{A}^{\boldsymbol{2}} \tag{2}$$

$$\boldsymbol{S}^{AA} = \boldsymbol{A}\,\boldsymbol{D}^{-\boldsymbol{1}}\boldsymbol{A} \tag{3}$$

$$\boldsymbol{S}^{RWR} = (1-\alpha)(\boldsymbol{I} - \alpha\boldsymbol{P})^{-1} \tag{4}$$

$$\boldsymbol{S}^{Katz} = (\boldsymbol{I} - \beta\boldsymbol{A})^{-1}\beta\boldsymbol{A} \tag{5}$$

where $\boldsymbol{A}$ denotes adjacency matrix of the graph, $\boldsymbol{D}$ denotes diagonal degree matrix, $\boldsymbol{I}$ denotes the identity matrix, $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{A}$ represents degree normalized adjacency matrix, i.e., transition probability matrix, and α and β are hyper-parameters.

By using various form of $\boldsymbol{S}$ matrix, Ou et. al., [6] solve, Equation (1) as singular value decomposition and return the top singular vectors corresponding to the top singular values as embedding matrix. Furthermore, they show that high-order proximity measures presented in Equation (4) and Equation (5) outperform lower-order proximity measures in Equation (2) and Equation (3) on link prediction tasks [6].

3.2.2 Our proposed method

Despite the effectiveness and efficiency of HOPE method [6], they do not take the ill-conditioned problem [7] of high-order proximity measures employed in their algorithm. More specifically, proximity measures in Equation (4) and Equation (5) might be ill-conditioned, which result in 0 singular value corresponding singular values because of "dangling nodes", i.e., nodes that do not have any connected nodes or close-zero values, smaller than the machine epsilon, due to normalization for small degree nodes. To see this, pay attention to inner part of Equation (4) , $(\boldsymbol{I} - \alpha\boldsymbol{P})$. When α is set to a large constant, which is usually preferred to rely on network structural information, inversion of the matrix faces with round-off error [7] which might result in wrong embedding matrix.

In this paper, to alleviate this problem, we regularize the matrix, $(\boldsymbol{I} - \alpha\boldsymbol{P})$, with a sparsity constraint, considering "dangling node" and small degree nodes, to attain a better and more correct embedding matrix. First, we write general assumption of network embedding: if nodes are close to each other in the graph, they must be close in the latent space, that can mathematically be states as follows:

$$\boldsymbol{\mathcal{J}} = \min \sum_{\boldsymbol{ij}} \boldsymbol{A_{ij}} \left\|\boldsymbol{h_i} - \boldsymbol{h_j}\right\| \tag{6}$$

where $\boldsymbol{h_i}$ and $\boldsymbol{h_j}$are embedding vectors of nodes i and j. It is well-known that Equation (6) can be stated in the matrix form as follows [11]:

$$\boldsymbol{\mathcal{J}} = \min \boldsymbol{h}^{\boldsymbol{T}}\,(\boldsymbol{D} - \boldsymbol{A})\boldsymbol{h} \tag{7}$$

Now, by scaling $\boldsymbol{A}$ with α and multiplying with $\boldsymbol{D^{-1}}$, we obtain the same equation in Equation (6). However, as aforementioned this matrix does not consider "dangling nodes" and/or small degree nodes. To overcome this limitation, we propose regularize Equation (7) with $\sigma\boldsymbol{I}$ and rewrite objective function as follows:

$$\boldsymbol{\mathcal{J}_{ours}} = \min \boldsymbol{h}^{\boldsymbol{T}}\,(\boldsymbol{I} - \alpha\boldsymbol{P} + \sigma\boldsymbol{I})\boldsymbol{h} \tag{8}$$

where, $\sigma = 0.5$ is a hyper-parameter. Finally, we present a new high-order proximity measure in HOPE as follows:

$$\boldsymbol{S}^{ours} = (1-\alpha)((1+\sigma)\boldsymbol{I} - \alpha\boldsymbol{P})^{-1} \tag{9}$$

In this paper, with this simple change, we are able to create a better proximity measures which leads significantly better network embedding in link prediction task than that of HOPE algorithm offers. Since we are using HOPE algorithm as our baseline method, we named our method as HOPE++.

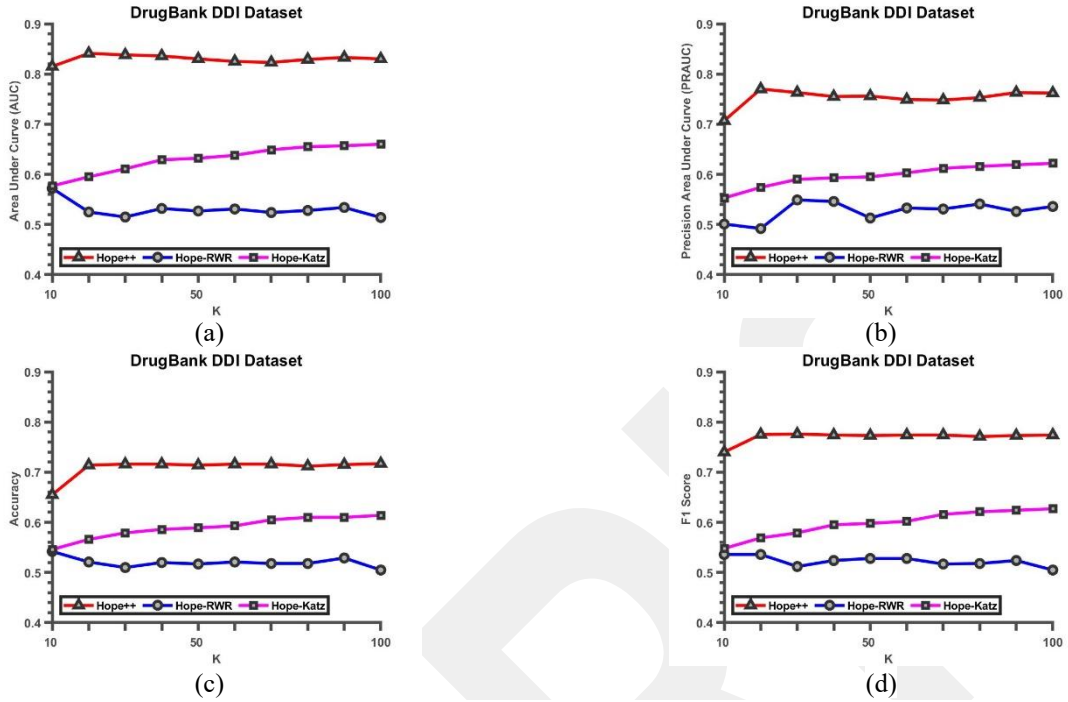

Figure. 1. Link prediction performance of our method, HOPE++, and a state-of-the-art method, HOPE with RWR and Katz high-order proximity on Drug Bank DDI dataset (a) x-axes are varying embedding dimension and y-axes is AUC score (b) x-axes are varying embedding dimension and y-axes is AUCPR score (c) x-axes are varying embedding dimension and y-axes is Accuracy score (d) x-axes are varying embedding dimension and y-axes is Macro-F1 score

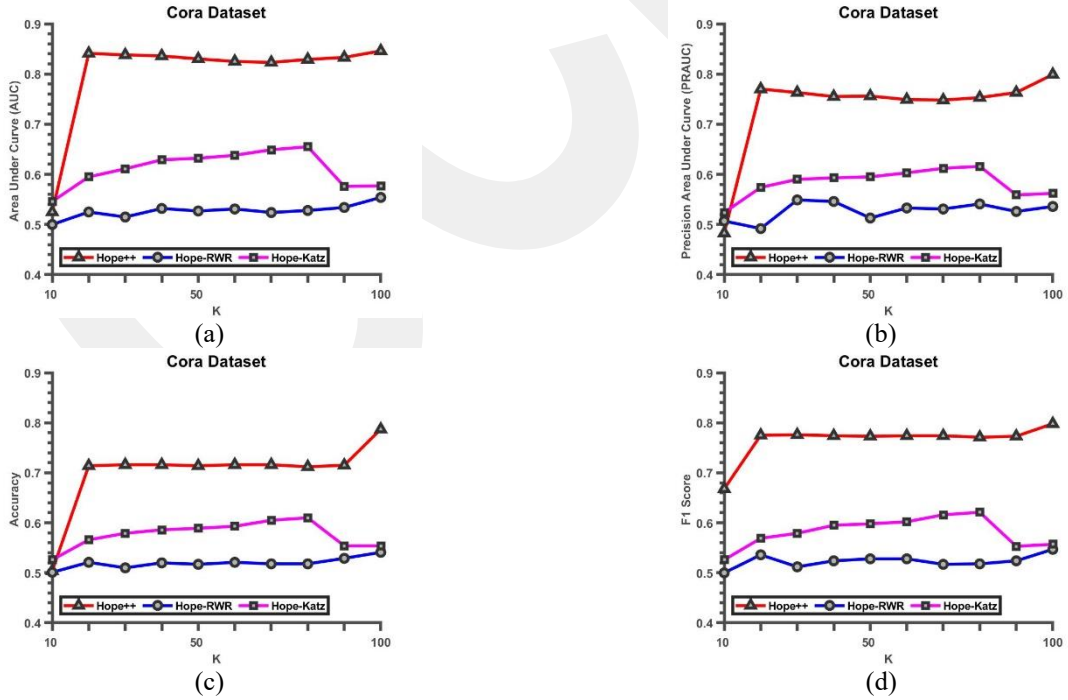

Figure. 2. Link prediction performance of our method, HOPE++, and a state-of-the-art method, HOPE with RWR and Katz high-order proximity on Cora dataset (a) x-axes are varying embedding dimension and y-axes is AUC score (b) x-axes are varying embedding dimension and y-axes is AUCPR score (c) x-axes are varying embedding dimension and y-axes is Accuracy score (d) x-axes are varying embedding dimension and y-axes is Macro-F1 score

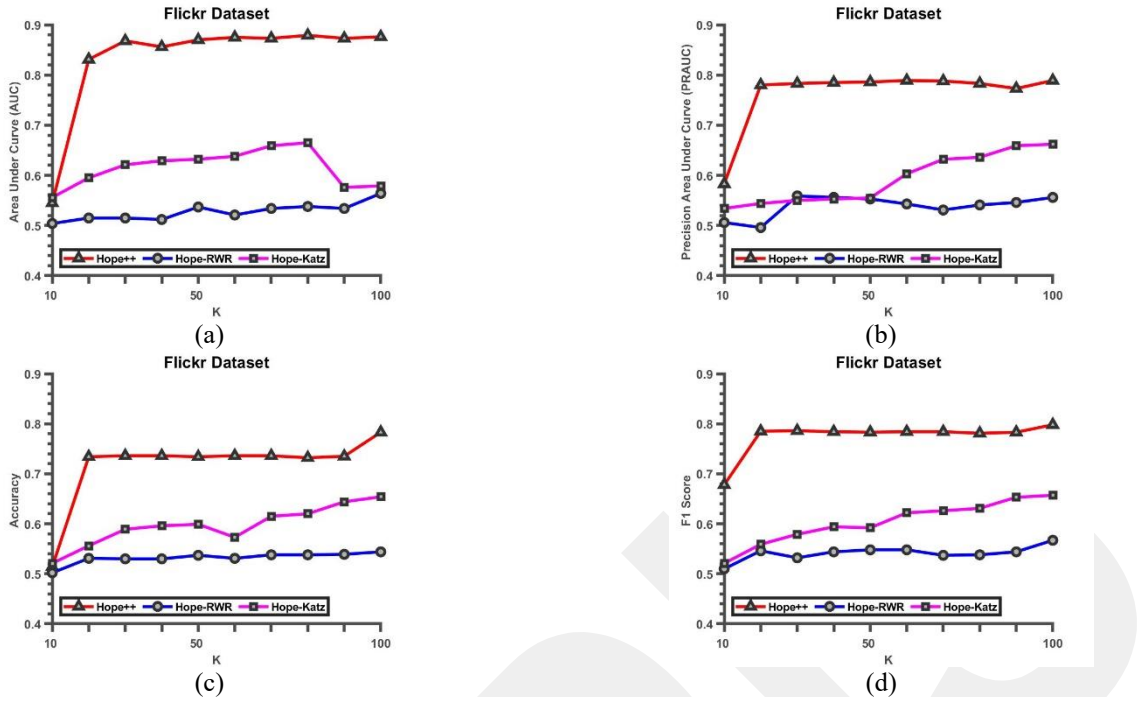

Figure. 3. Link prediction performance of our method, HOPE++, and a state-of-the-art method, HOPE with RWR and Katz high-order proximity on Flickr dataset (a) x-axes are varying embedding dimension and y-axes is AUC score (b) x-axes are varying embedding dimension and y-axes is AUCPR score (c) x-axes are varying embedding dimension and y-axes is Accuracy score (d) x-axes are varying embedding dimension and y-axes is Macro-F1 score

4 Results and discussions

In this section, we systematically evaluate the link prediction performance of our proposed method, HOPE++. Since HOPE++ uses HOPE algorithm [6] as a workhorse method, we compare our method against HOPE algorithm's high order proximity measures, namely RWR and Katz.

We start our discussion by describing the datasets and experimental setup used in this paper. We then give performance evaluation of embeddings attained by our methods and HOPE with RWR and Katz methods on link prediction task as a function embedding dimension. To this end, we use Area Under Curve (AUC), Area Under Precision Curve (AUPR), Accuracy and Macro-F1 scores evaluation metric.

4.1 Datasets and experimental setup

We use three publicly available real-world datasets whose static is summarized in Table 1. These datasets are obtained from [1], [4] and [6].

Table 1. Descriptive static of networks used in this paper

Datasets	# of Nodes	# of Edges
Drug_Bank_DDI	2.191	240.027
Cora	2.708	10.556
Flickr	7.575	239.738

Drug Bank Drug-Drug Interaction: This dataset represents the association/similarity among various drugs, which are the node and associations are edges.

Cora Dataset: This dataset represents the citation networks, where nodes are papers and links are citation relationship among the papers.

Flickr Dataset: This dataset represents an online community, where people are represented as nodes and their common interests denote edges in the graph.

In our experiments, we use Python code provided [1] and implement HOPE++ on top of this Python code. We evaluate performance of HOPE++ and HOPE algorithm that used RWR and Katz high-order proximity with default hyper-parameters on link prediction task as a function of varying embedding dimension.

4.2 Performance evaluation

To assess the performance of the proposed method, HOPE++, and original HOPE algorithm, we use link prediction problem as a benchmark problem. To this end, we first divide the networks into training and testing by 80% and 20% respectively. We use 80% of links for embedding purpose. We then tread 20% links as positive test links. Furthermore, we randomly sample 20% negative links by checking if two nodes are connected in both training and testing networks. Subsequently, we use embedding matrices attained by HOPE++, HOPE-RWR and HOPE-Katz on training networks, and positive and negative pairs' corresponding embeddings are fused by the dot (Hadamard) product so that we obtain a single score for each pair. Finally, we use these single scores and associated labels (1 if we are evaluating positive pairs; 0 otherwise) and feed them to Logistic Regression Classifier with 80% training and 20% testing.

We repeat experimental process for ten times and report means of evaluation metrics for each algorithm. Result of these analyses are depicted in Figures 1, 2 and 3. More specifically, red lines in the figures are our proposed methods, HOPE++, that uses $\boldsymbol{S}^{ours} = (1-\alpha)((1+\sigma)\boldsymbol{I} - \alpha\boldsymbol{P})^{-1}$ while blue and magenta lines represent HOPE algorithm that uses $\boldsymbol{S}^{RWR} = (1-\alpha)(\boldsymbol{I} - \alpha\boldsymbol{P})^{-1}$ and $\boldsymbol{S}^{Katz} = (\boldsymbol{I} - \beta\boldsymbol{A})^{-1}\beta\boldsymbol{A}$, respectively. As seen in the figures, across all datasets and all evaluation metrics, HOPE++, drastically improve performance of baseline method, HOPE algorithm [6], suggesting that a simple shift-and-invert based approach can drastically boost the performance of exiting linear embedding methods.

From Figures 1, 2 and 3, we can observe that our proposed method can deliver better results than HOPE with small dimension. This observation suggests that leading singular vectors in our method can capture the general connectivity structure better than that of HOPE method [6]. Furthermore, across all figures, we can observe performance oscillation for RWR-based embedding, suggesting wrong singular vector computation due to round-off error. On the other hand, our method delivers a smooth curve performance, hinting the importance of regularization approach we propose in this paper.

5 Conclusions

In this paper, we propose an alternative linear MF-based network embedding methods by capitalizing on the shift-and-invert approach. The idea of using shift-and-invert regularization is based on the premise that low degree nodes in a graph can cause round-off error in the inversion of proximity matrices. To eliminate this adversely effects of small degree nodes, we regularize the Graph Laplacian by identity matrix so that we can increase the diagonally dominancy. In order to evaluate our proposed approach, we use link prediction task as a benchmark problem on three real-world datasets. Extensive experimental evaluations for the link predict task demonstrate that our approach highly renders to improve MF-based embedding approach that uses well-known high-order proximity measures, such as random walk with restarts and Katz. The future effort in this direction would include incorporation of other learning tasks, such as node classification and their bioinformatics applications.

Conflict of interest

The author declares that there is no conflict of interest.

Similarity rate (iThenticate): 8%

References

[1] X. Yue, Z. Wang, J. Huang, S. Parthasarathy, S. Moosavinasab, Y. Huang, S. M. Lin, W. Zhang, P. Zhang and H. Sun, Graph embedding on biomedical networks: methods, applications and evaluations. Bioinformatics, 36(4), 1241–1251, 2020. https://doi.org/10.1093/bioinformatics/btz718

[2] B. Perozzi, R. Al-Rfou and S. Skiena, Deepwalk: online learning of social representations. In The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710. ACM, New York, NY, 2014.

[3] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan and Q. Mei, Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1067–1077, ACM, Florence, Italy, 2015.

[4] T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks. 4th International Conference on Learning Representations (ICLR), 2016.

[5] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio and R. D. Hjelm, Deep graph infomax. 7th International Conference on Learning Representations (ICLR), 2019

[6] M. Ou, P. Cui, J. Pei, Z. Zhang and W. Zhu, Symmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACMSIGKDD International Conference on Knowledge Discovery and DataMining, pp. 1105–1114, ACM, San Francisco, CA, 2016.

[7] Y. Saad, Iterative Methods for Sparse Linear Systems. PWS Publishing Co., Boston, 1996.

[8] M. Balasubramanian, et al., The isomap algorithm and topological stability. Science 295.5552, pp. 7–7, 2002. doi: 10.1126/science.295.5552.7a

[9] L. K. Saul and S. T. Roweis, An introduction to locally linear embedding. In:unpublished. Available at: http://www.cs.toronto.edu/roweis/lle/publications, 2000. Accessed 27 May 2022.

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

[11] A. Grover and J. Leskovec, node2vec: Scalable feature learning for networks. In:Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864, ACM, San Francisco, CA, 2016

[12] L. FR. Ribeiro, P. HP. Saverese and D. R. Figueiredo, struc2vec: Learningnode representations from structural identity. In:Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 385–394, 2017.

[13] M. Coskun and M. Koyutürk, Link prediction in large networks by comparing the global view of nodes in the network. In *2015* IEEE International Conference on Data Mining Workshop (ICDMW), pages 485–492. IEEE, NJ, USA, 2015.

[14] Z. Stanfield, M. Coskun and M. Koyuturk, Drug response prediction problem as link prediction problem. Scientific Reports, 7, 40321, 2017. https://doi.org/10.1038/srep40321.

[15] X. Huang, J. Li and X. Hu, Label informed attributed network embedding. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. 731–739, 2017.

[16] L. FR. Ribeiro, P. HP. Saverese and D. R. Figueiredo, struc2vec: Learningnode representations from structural identity. In:Proceedings of the 23rd ACM

SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 385–394, 2017.

[17] M. Coskun and M. Koyutürk, Node similarity-based graph convolution for link prediction in biological networks. Bioinformatics, *37*(23), 4501-4508, 2021. https://doi.org/10.1093/bioinformatics/btab464

[18] J. Klicpera, A. Bojchevski and S. Günnemann, Predict then propagate: Graph neural networks meet personalized pagerank. International Conference on Learning Representations, 2018.

[19] J. Gilmer, et al., Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning, Volume 70, pages 1263–1272. JMLR., 2017.

[20] Q. Li, et al., Deeper insights into graph convolutional networks for semi-supervised learning. In Thirty-Second AAAI Conference on Artificial Intelligence, April 29, 2018

[21] D. Liben-Nowell and J. Kleinberg, The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology, 58(7), 1019–1031, 2007. https://doi.org/10.1002/asi.20591