Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

Node Similarity Based Graph Convolution for Link Prediction in Biological Networks

Mustafa Coşkun [1,*] and Mehmet Koyutürk [2, 3]

[1]Department of Computer Engineering, Abdullah Gül University, Kayseri, 38080, Turkey and
[2]Department of Computer and Data Sciences, [3]Center for Proteomics and Bioinformatics, Case Western Reserve University, Cleveland, OH, 44106, USA.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Background: Link prediction is an important and well-studied problem in network biology. Recently, graph representation learning methods, which aim to learn low-dimensional representations of node attributes and topological characteristics, have drawn increasing attention in link prediction. In particular, Graph Convolutional Network (GCN)-based network embedding methods demonstrate promise in link prediction due to their ability to represent topological relationships in a network using non-linear functions.
Motivation: An important component of GCN-based network embedding is the convolution matrix, which is used to propagate features across the network. Existing algorithms use a graph Laplacian-based matrix for this purpose. In parallel, it has been shown that GCNs with a single layer tend to generate more robust embeddings by reducing the number of parameters. Laplacian-based convolution is not well suited to single layered GCNs, as it limits the propagation of information to immediate neighbors of a node.
Results: Capitalizing on the rich literature on unsupervised link prediction, we propose using node similarity based convolution matrices to compute node embeddings using GCNs. We select eight representative node similarity measures (Common Neighbors, Jaccard Index, Adamic-Adar, Resource Allocation, Hub Depressed Index, Hub Promoted Index, Sorenson Index, Salton Index) and use these as convolution matrices within a single-layered GCN to compute node embeddings. We systematically compare the performance of the resulting algorithms against GCNs that use the Laplacian-based matrix for convolution, as well as other link prediction algorithms, on three important biomedical link prediction tasks: drug-disease association (DDA) prediction, drug–drug interaction (DDI) prediction, protein–protein interaction (PPI) prediction. Our experimental results show that the node similarity-based convolution matrices considerably improve the performance of GCN-based embedding algorithms in link prediction.
Conclusions: As sophisticated machine learning frameworks are increasingly employed in biological applications, historically well-established methods can be used to reduce the complexity and improve the robustness of resulting models.
Availability: Our method is implemented as a Python library and is available at githublink
Contact: mustafa.coskun@agu.edu.tr
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Graphs (networks) are commonly used to represent a broad range of interactions and associations (as edges) among biomedical entities (as nodes)(Cowen *et al.*, 2017). Developing computational methods to analyze and understand these networks is one of the major research challenges in bioinformatics. A common problem that arises in the analysis of biomedical networks is the prediction of new associations or interactions using existing information on the network(s). This problem is often

abstracted in the form of "link prediction", a commonly studied problem in data mining and machine learning (Lü and Zhou, 2011). A considerable amount of research has been devoted to developing computational methods to predict/identify the missing/spurious links in various biomedical networks, such as disease gene prioritization (Erten *et al.*, 2011a), prediction of drug-disease associations (DDA) networks (Liang *et al.*, 2017), functional annotation of long non-coding RNAs (lncRNA) (Zhang *et al.*, 2018), denoising of protein interaction networks (Yoo *et al.*, 2015), and prediction of drug response in cancer cell lines (Stanfield *et al.*, 2017).

Earlier approaches to link prediction aim to assess the similarity between pairs of nodes based on local topological features (Zhou *et al.*, 2009). These local features usually focus on the shared neighborhood of node pairs and differ from each other in terms of how they evaluate the size of the overlap and the individual nodes in the overlap. Post-genomic developments in network biology establish the relevance of global network topology in delineating the functional relationships between biomolecules (Cowen *et al.*, 2017; Pandey *et al.*, 2008). Motivated by these insights, network proximity quantified via random-walk based algorithms is commonly utilized for link prediction (Valdeolivas *et al.*, 2019).

While powerful in capturing global network topology, random walk based methods have several limitations, including degree bias (Erten *et al.*, 2011a; Coşkun and Koyutürk, 2015), over-emphasis of proximity information at the expense of structural information (Devkota *et al.*, 2020; Ribeiro *et al.*, 2017), and dependency on the choice of hyperparameters (usually, the damping factor) (Perozzi *et al.*, 2014; Grover and Leskovec, 2016). Topological similarity based algorithms aim to circumvent these issues by using random walk based proximity scores as topological features (Cao *et al.*, 2014; Erten *et al.*, 2011b; Lei and Ruan, 2013). The concept of topological similarity is further generalized by node embeddings, which provide representations of nodes in a multi-dimensional latent feature space (Perozzi *et al.*, 2014; Grover and Leskovec, 2016). The objective of node embedding is to optimize the embedding space and the mapping of nodes to this space in such a way that nodes that are "similar" in the network are "close" to each other in the embedding space. By representing nodes as vectors in multi-dimensional feature space, node embeddings enable use of of-the-shelf representation learning algorithms for link prediction (Perozzi *et al.*, 2014).

Earlier algorithms for node embedding utilize random walk-based objectives to define node "similarity" (Tang *et al.*, 2015; Perozzi *et al.*, 2014; Grover and Leskovec, 2016; Hamilton *et al.*, 2019). With the advent of deep learning, neural network based algorithms are also applied to the computation of node embeddings (Kipf and Welling, 2016b; Gilmer *et al.*, 2017). Graph Auto-Encoder (GAE) is a direct application of Graph Convolutional Networks (GCNs) to the computation of node embeddings (Kipf and Welling, 2016b; Gilmer *et al.*, 2017). GAE uses a loss function that aims to reconstruct the adjacency matrix of the network using a dot product decoder. Veličković *et al.* (2019) propose an improved objective function for GCNs which aims at learn node embeddings that maximize the mutual information between the embedding vectors and the global representation of the entire network. Veličković *et al.* (2019)'s algorithm, Deep Graph Infomax (DGI), uses this improved objective function to limit the neural network to a single layer, thereby reducing the number of parameters to be learned. Despite DGI's Veličković *et al.* (2019) effectiveness, its use of the graph Laplacian as the convolution matrix limits its ability to propagate features across the network.

In this paper, we aim to develop an effective method for the computation of node embeddings in biological networks by integrating three key insights: 1) GCNs are potentially effective in computing powerful node embeddings for biological networks. 2) Reduced number of layers in GCNs renders the computation of node embeddings more stable and robust. 3) Local measures of node similarity, which demonstrated effectiveness in early applications of unsupervised link prediction, can provide "shortcuts" for shallow neural networks to propagate features across the network in a way that is useful for link prediction. In other words, we propose using node similarity matrices (computed using local measures of node similarity) as convolution matrices GCNs that are used to compute node embeddings for link prediction. To comprehensively investigate the promise of this idea, we explore the effectiveness of the node similarity measures as convolution matrices in DGI's single-layered GCN encoder by focusing on eight representative measures of node similarity Zhou *et al.* (2009).

To systematically investigate the potential of node similarity matrices as convolution matrices for link prediction, we use BIONEV, a framework developed by Yue *et al.* (2020) to benchmark link prediction algorithms in biomedical applications. Using BIONEV, We focus on three link prediction tasks: (i) prediction of drug-disease associations (DDAs) (Gottlieb *et al.*, 2011), (ii) prediction of drug-drug interactions (DDIs) (Zhang *et al.*, 2018), and (iii) prediction of protein-protein interactions (PPIs) prediction Wang *et al.* (2017). Our results show that GCN encoders equipped with node similarity based convolution matrices consistently outperform those that utilize the Laplacian-based convolution matrix. Interestingly, we observe that the Laplacian-based convolution matrix is useful when the network is very sparse and its eigenvalue decay is slow. These results show that insights provided by established techniques in unsupervised link prediction can help improve the accuracy of new machine learning techniques.

2 Methods

In this section, we first define biomedical link prediction problem and how it can be solved via graph embedding methods. Next, we present a brief background on two graph convolution network Kipf and Welling (2016a) based approaches, namely Graph Autoencoder Kipf and Welling (2016b) and Deep Graph Infomax Veličković *et al.* (2019) and explain how they aim to solve network embedding problem by setting different objective functions. We then present an overview about how these two embedding techniques, GAE and DGI, facilitate Laplacian-based convolution matrix in their GCN encoders Kipf and Welling (2016b); Veličković *et al.* (2019). Subsequently, we show that this Laplacian-based convolution matrix within the GCN encoder can replaced by various node similarity measures to avoid Laplacian-based convolution matrix reliance of these algorithms and to comprehensively characterize the effect of the convolution matrix on algorithm performance for link prediction in biomedical networks.

2.1 Link Prediction and Network Embedding

The task of predicting new links that are likely to emerge/disappear in a network is a fundamental problem in network analysis (Lü and Zhou, 2011). In the context of biomedical networks, link prediction is useful in discovering previously unknown associations or interactions, as well as identifying missing or spurious interactions (Yue *et al.*, 2020). Motivated by widespread application of network models in computational biology, significant effort has been devoted to the development of algorithms for link prediction in various types of biomedical networks, including protein-protein interactions (PPIs) (Wang *et al.*, 2017), drug-drug associations (Liang *et al.*, 2017), drug-disease associations (Zhang *et al.*, 2018), disease-gene associations (Erten *et al.*, 2011a), and drug response in cancers (Stanfield *et al.*, 2017).

In a general setting, the link prediction problem can be stated as follows: Given a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of n entities (e.g., genes/proteins, biological processes, functions, diseases, drugs etc.) and $\mathcal{E}$ denotes a set of m interactions/associations among these entities, predict pairs of entities that may also be interacting or associated with each other (Yue *et al.*, 2020). Link prediction can be supervised or unsupervised, where unsupervised link prediction aims to directly score and rank pairs of nodes using features derived from network topology. Supervised link

prediction, on the other hand, uses a set of "training" edges and non-edges to learn the parameters of a function that relates these topological features to the likelihood of the existence of an edge.

To extract features that represent network topology, graph representation learning techniques are used to embed the nodes of the network into a multi-dimensional feature space. For a given biomedical network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a network embedding is defined as a matrix $\mathbf{H} \in \mathbb{R}^{n \times d}$, where $n = |\mathcal{V}|$ and d is a parameter that defines the number of dimensions in the embedding space. Each row of this matrix represents, for each biomedical entity $u \in \mathcal{V}$, the embedding of u as $\mathbf{h}_u \in \mathbb{R}^d$.

To facilitate supervised link prediction using node embeddings as features, a given number of edges are randomly sampled from $\mathcal{E}$. To generate a set of "negative" samples, the same number of node pairs from the set $\mathcal{V} \times \mathcal{V} - \mathcal{E}$ are also randomly sampled. Next, for a given pair of nodes $(u, v) \in \mathcal{V} \times \mathcal{V}$, their corresponding embeddings, $\mathbf{h}_u, \mathbf{h}_v \in \mathbb{R}^d$ are concatenated with the label 1 or 0 depending on whether (u, v) represents a positive (extant edge) or negative (non-extant edge) sample. Finally, these combined latent features with their labels are fed into a supervised machine learning algorithm (e.g. support vector machine (SVM), Random Forest), to train a classifier for link prediction (Yue *et al.*, 2020).

In a recent study, Yue *et al.* (2020) extensively investigate the effectiveness of network embedding techniques in the context of *supervised* link prediction on a broad range of biomedical networks. Their results show that the accuracy of link prediction depends highly on the technique used to compute network embeddings and different embedding techniques can be effective for different applications and networks. Among various embedding techniques, graph convolutional network (GCN) based embedding approaches deliver encouraging results for most of the biomedical link prediction tasks (Yue *et al.*, 2020).

2.2 Network Embedding via Graph Convolutional Networks

Graph Convolutional Networks (GCNs) are simplified versions of Graph Convolutional Neural Networks (GCNNs), which are generalizations of conventional Convolutional Neural Networks (CNNs) on graphs (Li *et al.*, 2018). In the context of various machine learning tasks, GCNs are used to facilitate use of network topology in computing latent features from input features associated with network nodes. GCNs are also used to compute node embeddings, i.e., features that represent network topology, by setting the loss function appropriately to capture the correspondence between the embeddings and network topology.

In GCNs, each graph convolution layer involves three steps: 1) feature propagation, 2) linear transformation, and 3) application of a non-linear activation function (Wu *et al.*, 2019). Feature propagation is accomplished by using a convolution matrix that is computed from graph topology. In the context of computing network embeddings, the choice of convolution matrix is critical as it defines the relationship between network topology and computed embeddings. The parameters of linear transformation are learned by training the GCN to minimize loss function and standard non-linear functions are used for activation (.e.g, sigmoid or ReLU). Thus, the key ingredients of a GCN-based network embedding technique are the choice of the convolution matrix and the loss function.

Graph Autoencoder (GAE): Kipf and Welling (2016b) propose GAE as a direct application of their GCN model (Kipf and Welling, 2016a) to the computation of node embeddings in a network. GAE uses the graph Laplacian as the convolution matrix in a two-layer neural network. In this context, the graph Laplacian-based convoltion matrix is defined as

$$\hat{\mathbf{L}}_{sym} = \tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2}, \tag{1}$$

(Note that normally, graph Laplacian is defined as $\mathcal{L} = \mathbf{I} - \mathbf{L}_{sym}$, for name convenience, we call $\hat{\mathbf{L}}_{sym}$ as graph Laplacian-based convolution matrix) where $\mathbf{A}$ is the adjacency matrix of the network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\tilde{\mathbf{A}} = \mathbf{I} + \mathbf{A}$ is the adjacency matrix with self loops added, $\mathbf{D} = diag(d_1, d_2, ..., d_n)$ is the degree matrix, and $\tilde{\mathbf{D}} = \mathbf{I} + \mathbf{D}$. Using $\hat{\mathbf{L}}_{sym}$ as the convolution matrix, GAE defines the network embedding matrix $\mathbf{H}_{GAE}$ as:

$$\mathbf{H}_{GAE} = ReLU(\hat{\mathbf{L}}_{sym} ReLU(\hat{\mathbf{L}}_{sym} \mathbf{I} \Theta^{(0)}) \Theta^{(1)}), \tag{2}$$

where $\Theta^{(0)}$ and $\Theta^{(1)}$ are trainable weight matrices. These weight parameters are trained using the following loss function:

$$\mathcal{L}_{GAE} = \underset{\Theta^{(0)}, \Theta^{(1)}}{\text{minimize}} \left\| \mathbf{A} - \sigma(\mathbf{H}_{GAE} \times \mathbf{H}_{GAE}{}^T) \right\|, \tag{3}$$

where σ denotes logistic sigmoid function.

Deep Graph Infomax (DGI): Veličković *et al.* (2019) develop DGI using the infomax principle (Linsker, 1988) to define a loss function that can be used in various learning settings. In the context of neural networks, DGI computes the embedding matrix $\mathbf{H}_{DGI}$ using a single layered neural network:

$$\mathbf{H}_{DGI} = PReLU(\hat{\mathbf{L}}_{sym} \mathbf{I} \Theta^{(0)}), \tag{4}$$

where PReLU denotes parametric ReLU (Veličković *et al.*, 2019) as the non-linear activation function and $\Theta^{(0)}$ is a trainable weight matrix. The loss function used to train $\Theta^{(0)}$ is defined as binary cross entropy loss:

$$\mathcal{L}_{DGI} = \sum_{u \in \mathcal{V}} \log \sigma(\mathbf{h}_u^T \mathbf{M} \mathbf{s}) + \sum_{i=1}^{n} \log \left(1 - \sigma(\tilde{\mathbf{h}}_i^T \mathbf{M} \mathbf{s})\right), \tag{5}$$

where $\mathbf{s} = \sigma(\frac{1}{n} \sum_{u \in \mathcal{V}} \mathbf{h}_i)$ represents the global graph-level summary, $\tilde{\mathbf{h}}_i$ for $1 \leq i \leq n$ denote the corrupted embedding vectors that are obtained by shuffling the nodes (randomly permuting the rows of $\mathbf{I}$), and $\mathbf{M} \in \mathbb{R}^{d \times d}$ is a trainable scoring matrix.

Although DGI has not yet been implemented in biological applications, it has demonstrated great potential in other applications Veličković *et al.* (2019). DGI owes its promising results to two factors: (i) capturing the global information of the network by incorporating node summaries and corrupted embeddings in its loss function, and (ii) utilizing the power of this loss function to reduce the number of layers, thereby the number of parameters to be learned. However, the single-layered nature of DGI also limits its ability to diffuse information across the network. In the context of link prediction, node embeddings are utilized to assess the similarity between pairs of nodes. Motivated by this consideration, we hypothesize that coupling of DGI's neural network architecture and loss function with convolution matrices that are based on node similarities can deliver superior link prediction performance as compared to convolution matrices that directly incorporate the adjacency matrix of the network.

2.3 Node Similarity Measures as Convolution Matrices

Algorithms for GCN-based network embedding demonstrate great promise in link prediction (Kipf and Welling, 2016a; Veličković *et al.*, 2019; Hamilton *et al.*, 2019; Wu *et al.*, 2019; Coskun, 2019). Effective application of these methods to link prediction tasks in biology requires careful consideration of the design choices that are encountered in the context of a specific problem. An important design choice in GCN-based network embedding is the choice of the convolution matrix. As discussed above, most of the existing algorithms use the Laplacian-based convolution matrix. To date, the effect of the convolution matrix on algorithm performance has not been comprehensively characterized in the context of link prediction in biomedical networks.

The benefit of a single-layered neural network (as in DGI) over a multi-layered neural network (as in GAE) is that, this reduces the number of parameters in the model and avoids over-smoothing. However, it also limits

the convolution to a single step in the network. Based on this observation, we stipulate that network similarity measures that capture local network information can be effective as convolution matrices in conjunction with a single-layered neural network. Such measures include those that have demonstrated success in earlier applications of link prediction, including Common Neighbors, Adamic-Adar, and others (Liben-Nowell and Kleinberg, 2007), Below, we describe these measures and discuss how they can be adopted into the framework of DGI as convolution matrices. In other words, we consider the following formulation for computing node embeddings (where $\Theta^{(0)}$ is optimized using the loss function in (5)):

$$\mathbf{H} = PReLU(\mathbf{CI}\Theta^{(0)}) \tag{6}$$

and discuss various options for the convolution matrix $\mathbf{C}$ based on the rich literature on unsupervised link prediction. Observe that, for both DGI and GAE, $\mathbf{C} = \hat{\mathbf{L}}_{sym}$.

(i) *Common Neighbors (CN):* For a given node $u \in \mathcal{V}$, let $\Gamma(u) \subseteq \mathcal{V}$ be the set of neighbors of u. Then, the count of common neighbors of nodes $u \in \mathcal{V}$ and $v \in \mathcal{V}$ is defined as (Zhou *et al.*, 2009):

$$s_{CN}(u,v) = |\Gamma(u) \cap \Gamma(v)| = |\{w \in \mathcal{V} | (v,w) \wedge (u,w) \in \mathcal{E}\}| \tag{7}$$

Since $(\hat{A}^2)_{u,u} = d_u + 1$ and for $u \neq v$, $(\hat{A}^2)_{u,v} = s(u,v)$, the convolution matrix representing count of common neighbors can be computed as:

$$\mathbf{C}_{CN} = \hat{\mathbf{A}}^2 \tag{8}$$

(ii) *Jaccard Index:* This measure assesses the overlap between two sets by normalizing the size of their intersections by the size of their union. In the context of link prediction, Jaccard Index is used to assess the degree of overlap between the neighbors of two nodes in a network Zhou *et al.* (2009):

$$s_{JI}(u,v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}. \tag{9}$$

In matrix form, Jaccard Index can be formulated as a convolution matrix as follows:

$$\mathbf{C}_{JI} = \hat{\mathbf{A}}^2 \oslash (\hat{\mathbf{A}}\mathbf{N} + \mathbf{N}\hat{\mathbf{A}} - \hat{\mathbf{A}}^2). \tag{10}$$

Here, $\mathbf{N}$ denotes an all-ones matrix with the same size as $\mathbf{A}$ and $\oslash$ denotes element-wise (Hadamard) division.

(iii) *Adamic-Adar (AA):* This commonly utilized measure of node similarity refines the notion of common neighbors by assigning more weight to less-connected common neighbors (Adamic and Adar, 2003):

$$s_{AA}(u,v) = \sum_{w \in |\Gamma(u) \cap \Gamma(v)|} \frac{1}{\log(d_w)}, \tag{11}$$

where d_w denotes the degree of node $w \in \mathcal{V}$. This notion of node similarity can be formulated as a convolution matrix as (Liben-Nowell and Kleinberg, 2007):

$$\mathbf{C}_{AA} = \hat{\mathbf{A}} \log(\hat{\mathbf{D}}^{-1}) \hat{\mathbf{A}} \tag{12}$$

(iv) *Resource Allocation (RA):* As Adamic-Adar, this measure also aims to reduce the effect of highly-connected common neighbors, but does so more agressively by normalizing with the degree of the neighbor (Zhou *et al.*, 2009). Thus Resource Allocation based convolution matrix can be formulated as:

$$\mathbf{C}_{RA} = \hat{\mathbf{A}} \hat{\mathbf{D}}^{-1} \hat{\mathbf{A}} \tag{13}$$

(v) *Hub Depressed Index (HDI):* Similar to Jaccard Index, HDI aims to normalize the overlap between neighbors of two nodes based on the degrees of the nodes, but does so by focusing on the node with higher degree Zhou *et al.* (2009), i.e.:

$$s_{HDI}(u,v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{\max(d_u, d_v)} \tag{14}$$

Using the notation introduced above, HDI-based convolution matrix can be formulated as:

$$\mathbf{C}_{HDI} = \hat{\mathbf{A}}^2 \oslash \max(\hat{\mathbf{A}}\mathbf{N}, \mathbf{N}\hat{\mathbf{A}}) \tag{15}$$

(vi) *Hub Promoted Index (HPI):* In contrast to Hub-Depressed Index, HPI normalizes the size of the overlap of the neighbors of two nodes by the degree of the less-connected node, thereby promoting hubs. This index can be represented as a convolution matrix as:

$$\mathbf{C}_{HPI} = \hat{\mathbf{A}}^2 \oslash \min(\hat{\mathbf{A}}\mathbf{N}, \mathbf{N}\hat{\mathbf{A}}) \tag{16}$$

(vii) *Sørenson Index (SI).* Similar to Jaccard Index, SI normalizes the size of the overlap of the two nodes by taking into account the degree of the two nodes, but uses the sum of the degrees instead of the size of the union:

$$s_{SI}(u,v) = 2\frac{|\Gamma(u) \cap \Gamma(v)|}{(d_u + d_v)} \tag{17}$$

Thus, compared to JC, SI is more conservative toward high-degree nodes as the common neighborhood is counted twice in the denominator. SI can be formulated as a convolution matrix as follows:

$$\mathbf{C}_{SI} = 2\hat{\mathbf{A}}^2 \oslash (\hat{\mathbf{A}}\mathbf{N} + \mathbf{N}\hat{\mathbf{A}}) \tag{18}$$

(viii) *Salton Index (ST):* ST also normalizes the size of the overlap by the degrees of the two nodes, but uses the geometric mean instead of the arithmetic mean for this purpose:

$$s_{ST}(u,v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{\sqrt{(d_u \times d_v)}} \tag{19}$$

ST can be formulated as a convolution matrix as follows:

$$\mathbf{C}_{ST} = \hat{\mathbf{A}}^2 \oslash \sqrt{(\tilde{\mathbf{D}}\tilde{\mathbf{D}}^T)} \tag{20}$$

where the square-root operation is applied element-wise to the $n \times n$ matrix $\tilde{\mathbf{D}}\tilde{\mathbf{D}}^T$.

To summarize our approach, given the adjacency matrix $\hat{\mathbf{A}}$ of a network, we use the node similarity measures to compute node embeddings for all nodes in the network as follows:

1. Compute the convolution matrix $\mathbf{C}$ based on the specified node similarity index (CN, JI, AA, RA, HDI, HPI, SI, or ST).
2. Compute embeddings $\mathbf{H} = PReLU(\mathbf{CI}\Theta^{(0)})$ using the single-layered GCN encoder
3. Randomly row-wise shuffle $\mathbf{I}$ to obtain corrupted node identities $\hat{\mathbf{I}}$.
4. Compute corrupted embeddings $\hat{\mathbf{H}} = PReLU(\mathbf{C}\hat{\mathbf{I}}\Theta^{(0)})$, using the single-layered GCN encoder.
5. Compute network-level summary of embeddings $\mathbf{s} = \sigma(\frac{1}{n}\sum_{i=1}^{n}\mathbf{h}_i)$.
6. Update $\Theta^{(0)}$ and $\mathbf{M}$ using gradient descent to minimize $\mathcal{L}_{DGI}$ (5).

Once the node embeddings are computed using the above (unsupervised) procedure, we feed these embeddings into BIONEV, the supervised link prediction algorithm implemented by Yue *et al.* (2020). BIONEV takes as input a training network and node embeddings, uses these embeddings to train supervised link prediction models, and uses a test dataset to evaluate the performance of the embeddings.

Table 1. **Descriptive statics of the networks used in computational experiments.** Avg. degree is defined as $\frac{|\mathcal{E}|}{|\mathcal{V}|}$, density is defined as $\frac{2|\mathcal{E}|}{|\mathcal{V}|^2}$.

	# Nodes ($\|\mathcal{V}\|$)	**# Edges ($\|\mathcal{E}\|$)**	**Avg. degree**	**Density**
Datasets				
DrugBank DDI	2191	242027	110.5	0.1
CTD DDA	12765	92813	7.3	0.0011
NDFRT DDA	13545	56515	4.2	0.0006
STRING PPI	15131	359776	23.8	0.0031

3 Results

In this section, we systematically compare the performance of graph Laplacian-based vs. node similarity measures as convolution matrices for graph embedding in the context of various link prediction tasks in biology. We start our discussion by describing the datasets and the experimental setup. We then compare the performance of Laplacian-based and node similarity measures using Area Under Curve (AUC) as a performance criterion for link prediction. Subsequently, we investigate the underlying reasons for the differences in the performance of these methods. In particular, we investigate the effect of graph density (which can be interpreted as the size of training data) and eigenvalue decay on the performance of the methods. Finally, we compare the link prediction performance of the best performing node similarity based convolution matrices against nine other state-of-the-art embedding methods.

3.1 Datasets and Experimental Setup

In our experiments, we use four biomedical networks compiled by Yue *et al.* (2020). The statistics of these four networks are shown on Table 1. These networks represent link prediction tasks in the context of three different biomedical applications:

- **DrugBank Drug-Drug Interactions (DDIs):** The DrugBank-DDI network is composed of verified pairwise interactions between chemical compounds used as drugs, obtained from DrugBank (Wishart *et al.*, 2018). DrugBank is a freely accessible online database that contains detailed information about drugs and drug interactions (Yue *et al.*, 2020).
- **CTD Drug-Disease Associations (DDA):** Comparative Toxicogenomics Database (CTD) is a database that catalogues the effects of environmental exposures. It contains associations between chemicals and diseases, representing toxic effects of chemicals (Davis *et al.*, 2019). We refer to this dataset as the CTD-DDA network.
- **NDFRT Drug-Disease Associations (DDA):** This dataset contains drug-disease associations based on National Drug File Reference Terminology (NDFRT) in the unified medical language system, in which there is an edge between a disease and drug if the drug is used for the treatment of the disease (Bodenreider, 2004). We refer to this dataset as the NDFRT-DDA network.
- **Protein-Protein Interactions (PPIs):** The PPI network contains *Homo sapiens* PPIs extracted from the STRING database (Szklarczyk *et al.*, 2015).

For GAE and DGI algorithms, we use the Python implementation provided respectively by Kipf and Welling (2016a) and Veličković *et al.* (2019). For other state-of-the-art network embedding methods (Table 2), we use OpenNE (https://github.com/thunlp/OpenNE), Python source code implementation. We implement our node similarity measure-based embedding methods on top of PyTorch implementation provided by Veličković *et al.* (2019).

We assess the performance of the algorithms using randomized test and training tests, where the randomized tests are repeated 10 times for each algorithm/parameter setting. For each randomized test, we select a certain fraction (referred to as *test ratio*) of the edges in the network uniformly at random, remove these edges from the network, and reserve them as the positive test set. We then compute node embeddings and perform training on the remaining network. For training, we concatenate the embeddings to construct a feature set for each pair of nodes and associated label (1 if the pair has an edge in the training network, 0 otherwise), and use these data to train a Logistic Regression based binary classifier. Subsequently, we make predictions for the pairs of nodes in the positive and negative test sets and compute the area under the ROC curve (AUC) accordingly. For the test ratio, we use %10, %30, and %50 as the fraction of edges removed from the network. For the neural networks used in computing node embeddings, we use default hyper-parameters suggested by the baseline papers and embedding dimension $d = 100$, training epoch 200.

3.2 Link Prediction Performance

We compare the link prediction performance of node similarity-based convolution matrices (using DGI's single-layered GCN encoder) against encoders that use the Laplacian matrix for convolution. As representative methods, we use the following methods for comparison: (i) Deep Graph infomax (DGI, which uses the single-layered GCN encoder we also use for the convolution matrices), (ii) Graph Auto-encoder (GAE, which uses a double-layered GCN encoder). Selection of these two methods for comparison enables assessment of the effect of the convolution matrix (similarity-based vs. DGI), as well as effect of the architecture of the GCN (DGI vs. GAE).

The comparison of the link prediction performance of node similarity based convolution matrices against that of Laplacian-based matrix on four different datasets is shown in Figure 1. The results shown in the figure suggest the following observations:

1. When graph Laplacian is used for convolution, DGI's single-layered neural network delivers superior prediction performance over GAE's two-layered neural network.
2. The accuracy of GAE's two-layered neural network tends to be more variable, particularly as the training data gets smaller.
3. The link prediction performance of DGI and all node-similarity based convolution matrices (all using single-layer neural network) is robust to the (smaller) size of training data.
4. The accuracy provided by node-similarity based convolution matrices tend to depend on the dataset.
5. For all but one dataset (NDFRT-DDA), most of the node similarity based convolution matrices deliver more accurate predictions as compared to DGI (6 on DrugBank-DDI, all 8 on CRD-DDA, all 8 on STRING-PPI).
6. Among the node similarity based convolution matrices, Resource Allocation (RA), Hub-deprived Index (HDI), and Hub-Promoted Index consistently deliver better accuracy than other node similarity measures.

These results demonstrate that node similarity based convolution matrices can be more effective than the graph Laplacian-based in computing node embeddings for various link prediction tasks on biological networks. As suggested by the superior performance and low variance of DGI as compared to GAE, the use of a single-layered neural network improves and stabilizes predictive performance. The use of node similarity matrices in a single-layered network adds to this improvement by enabling the network to take multiple steps during convolution. Node similarity based convolution accomplishes this by using established "features" that are proven to be useful in link prediction to guide convolution.

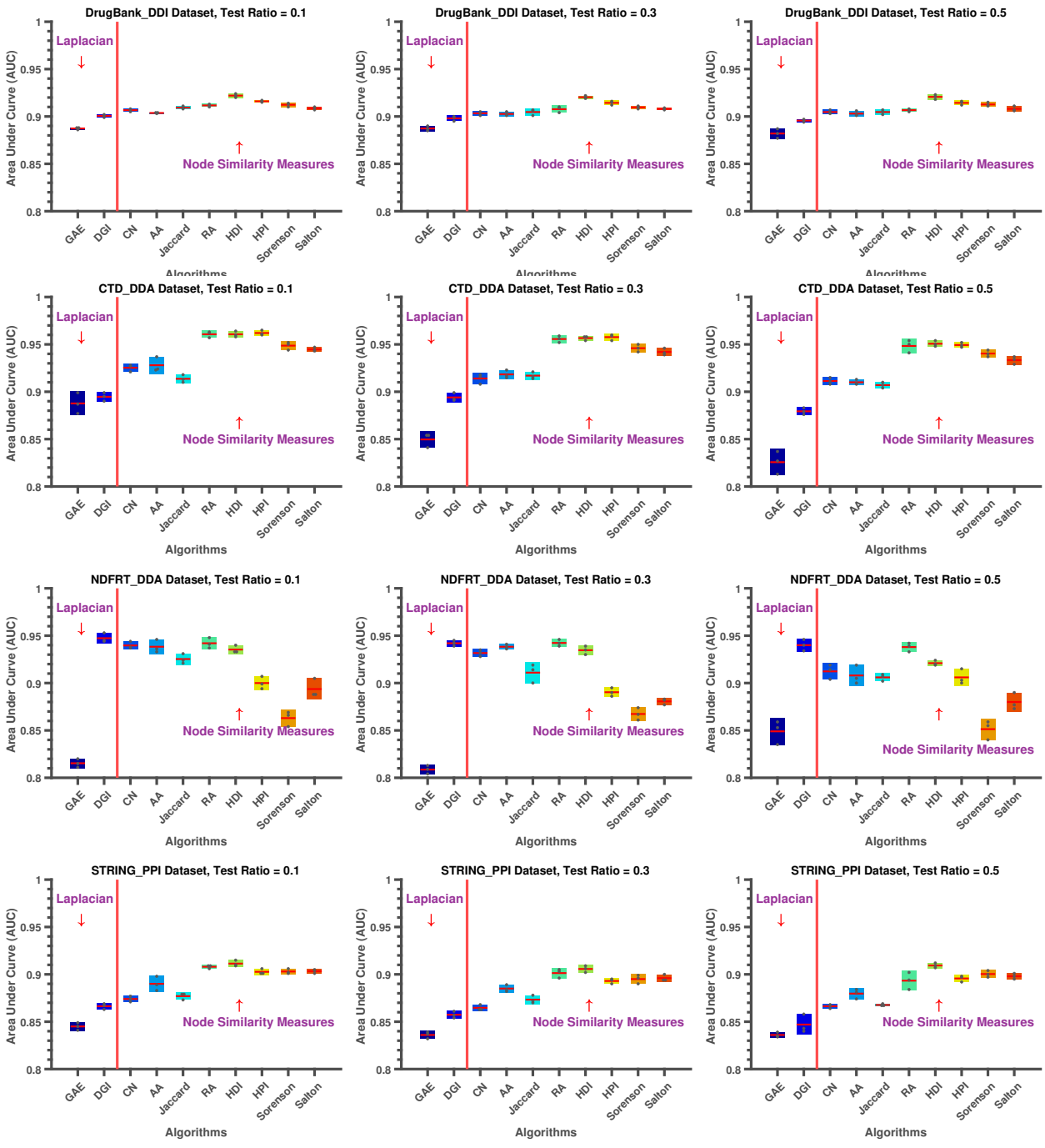
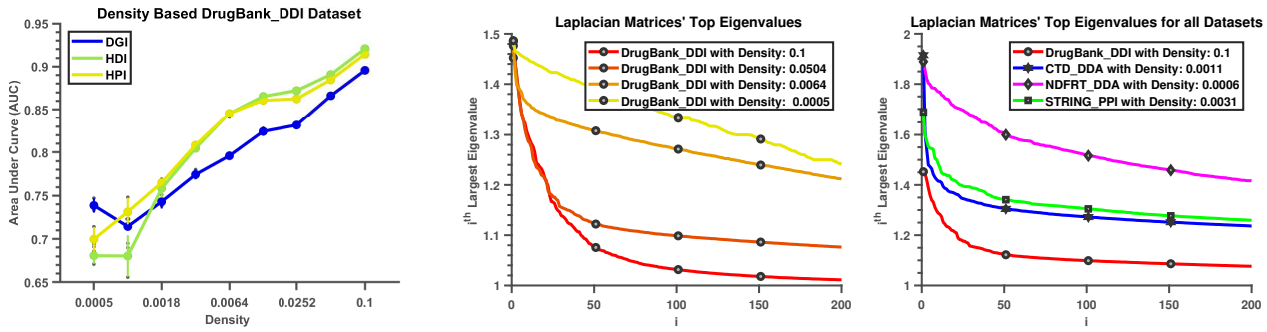


Fig. 1: **Link prediction performance of node embeddings computed using different convolution matrices.** In each figure, the x-axis shows the neural network architecture (on the left of the red line; these methods use the graph Laplacian as the convolution matrix) or convolution matrix (on the right of the red line; these methods use a node similarity based convolution matrix on a single layered neural network as in DGI), the y-axis shows the area under ROC curve (AUC) for link prediction. Each row corresponds to a different dataset, each column represents different test ratios (e.g., on the left-most column 10% of the edges in the network are deleted and used as positive test samples, the remaining 90% of the edges are used for training), where the training data gets smaller as we move from left to right. GAE: Graph Autoencoder, DGI: Deep Graph Infomax, CN: Number of common neighbors, AA: Adamic-Adar, RA: Resource Allocation, HDI: Hub-deprived index, HPI: Hub-promoted index.

3.3 Effect of Graph Density and Spectra

Node similarity based convolution matrices perform substantially better than DGI and GAE (using the graph Laplacian-based convolution matrix) on three out of four biomedical networks on link prediction tasks. They perform slightly worse than DGI on the NDFRT-DDA network. This network is the most sparse among the four networks we consider in our experiments. Motivated this observation, we further investigate the effect of network density on the accuracy of link prediction. For this purpose,



(a) (b) (c)

Fig. 2: **The relationship between the topological properties of input graphs and the link prediction performance of convolution matrices.** (a) Area under ROC curve for the single-layered neural network using graph Laplacian (DGI), Hub-deprived Index (HDI), and Hub-promoted Index (HPI) as a function of network density. Networks with density ranging from 0.0005 to 0.1 are generated by randomly sampling edges from the Drugbank-DDI dataset. (b) The spectra of four networks with different levels of density sampled from Drugbank-DDI. (c) The spectra of the four networks that are considered in our experiments.

Table 2. **Comparison of the link prediction performance of neural network based node embeddings and other state-of-the-art algorithms on four biological networks.** Similarity-Based-NN refers to node embeddings computed using node similarity based convolution matrices (RA: Resource Allocation, HDI: Hub-deprived Index, HPI: Hub-promoted Index), Laplacian-Based-NN refers to node embeddings using the graph Laplacian-based convolution matrix, RandomWalk refers to methods that use random walk based proximity measures to predict links, and MatrixFactorization refers to methods that represent link prediction as a matrix completion problem and use matrix factorization based algorithms to solve this problem. For each dataset, the best performing method(s) is (are) **underlined and shown in bold.**

	$Similarity-Based-NN$			$Laplacian-Based-NN$			$RandomWalk$			$MatrixFactorization$		
Datasets	RA	HDI	HPI	GAE	DGI	LINE	DeepWalk	node2vec	struc2vec	SVD	HOPE	GraRep
DrugBank_DDI	0.912	**0.924**	0.918	0.891	0.896	0.909	0.917	0.889	0.901	0.912	0.910	0.913
CTD_DDA	0.961	0.960	**0.962**	0.866	0.889	0.958	0.926	0.903	0.959	0.934	0.950	0.959
NDFRT_DDA	0.944	0.933	0.907	0.715	**0.953**	0.944	0.783	0.736	0.951	0.774	0.949	**0.953**
STRING_PPI	0.909	**0.915**	0.906	0.845	0.863	0.855	0.885	0.798	0.902	0.867	0.838	0.883

we use the densest network in our datasets, namely DrugBank_
randomly sample edges from the DrugBank_DDI to construct
with density ranging from 0.0005 to 0.1 (the network's original
We then perform cross-validation on these sampled networks with
ratio. Observe that as the network gets sparser, the availability o
data declines drastically.

The results of our density analysis are shown in Figure 2(a)(a)
in the figure, the network's density plays a major role on the perf
of link prediction algorithms. While the accuracy provided by all
declines steadily as density goes down, the accuracy of node s
based convolution matrices stays above that of the graph Laplacian
graph becomes extremely sparse. When graph density goes down
we observe that the accuracy of node similarity based convolution
becomes more variable and worse than that of the graph Laplacia

Since network density by itself may not be a useful measu
"information content" of a network, we further investigate the
top eigenvalues of the (self-loop added) Laplacian-based matrix
purpose, we visualize the eigenvalue decay of four sparsified versions of the NDFRT-DDA network are shown in Figure 2(b) (b). Similarly, we visualize the eigenvalue decay for all of the four networks considered in our experiements Figure 2(b) (c). Interestingly, for both original and sampled networks, the performance of link prediction algorithms and convolution matrices appear to strongly depend on the decay rate of the eigenvalues of the graph Laplacian-based matrix. Namely, for networks for which the eigenvalues decay rapidly, node similarity based convolution matrices appear to deliver improved performance. In contrast, for networks whose eigenvalues exhibit slower decay, Laplacian-based convolution matrices appear to work better.

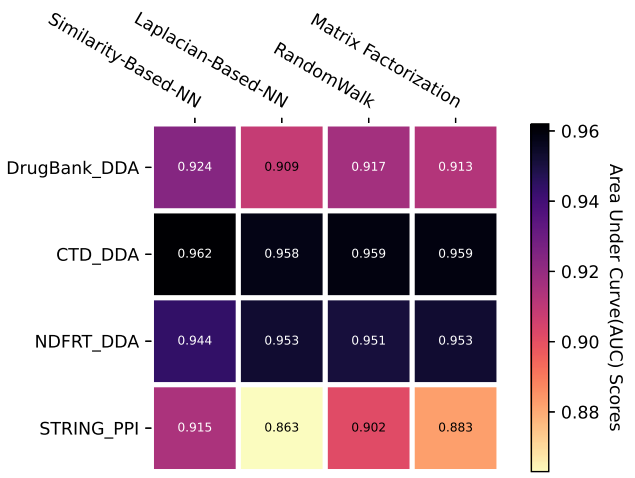


Fig. 3: **Comparison of the link prediction performance of neural network based node embeddings and other state-of-the-art algorithms.** The heatmap shows the accuracy of best performing method from each categories in Table 2 accross all datasets (e.g., on the left-most column of the plot, the cell entries represent the best AUC scores shown in Table 2 for the Similarity-Based-NN embedding category)

3.4 Comparison to Other Link Prediction Algorithms

Algorithms that are used for link prediction in biological networks are not limited to those that utilize neural network based node embeddings.

Many other approaches exist, including unsupervised methods that use node similarity (which we use as convolution matrices in this work), random walk based algorithms (which use random walks to compute node embeddings), and matrix factorization based methods (which formulate link prediction as one of a matrix completion problem). To further evaluate the performance of the node similarity-based methods against state-of-the-art methods in link prediction, we consider three different categories of approaches: (i) Neural Network based algorithms, (ii) Random Walk based algorithms, and (iii) Matrix Factorization based algorithms. For each of these three categories, we select three algorithms that are reported to perform best on biological networks Yue *et al.* (2020) and compare the link prediction performance of these algorithms on the four networks we use in our experiments. The results of these experiments are shown on Table 2.

We observe that the CTD-DDA dataset presents the "easiest" instance of link prediction for all algorithms, but two neural network-based algorithms that use the Laplacian-based matrix (GAE and DGI) do not perform as well as other algorithms on this instance. We also observe that the task of predicting protein-protein interactions (PPIs) using existing PPIs in the String-PPI dataset presents the most difficult instance for most of the algorithms. For this instance, neural network based node embedding with node-similarity based convolution matrices clearly outperform other algorithms. Neural network based algorithms' relative performance against other algorithms is worst for the NDFRT-DDA network, which is the most sparse network among all datasets we consider. In general, many other algorithms (inlcuding GAE, DeepWalk, node2vec, and SVD) also have difficulty in link prediction on this network.

These results are summarized for the four categories of algorithms (similarity-based neural network embeddings, Laplacian-based neural network embeddings, random walk based algorithms, matrix factorization based algorithms) in Figure 3. As seen in the figure, neural network based embeddings that use node similarity-based convolution matrices mostly outperform other algorithms.

4 Conclusion

In this paper, by capitalizing on the rich literature on unsupervised link prediction, we propose using node similarity based convolution matrices to learn condensed representation of a given biological network using GCNs. For this purpose, we select eight representative node similarity measures (Common Neighbors, Jaccard Index, Adamic-Adar, ResourceAllocation, Hub Depressed Index, Hub Promoted Index, Sorenson Index, Salton Index) and use these as convolution matrices within a single-layered GCN to compute node embeddings. Extensive experimental evaluation on three important biomedical link predictiontasks: drug-disease association (DDA) prediction, drug–druginteraction (DDI) prediction, protein–proteininteraction (PPI) prediction indicates that node similarity based convolution matrices in the single-layerd GCN encoder deliver much better link prediction results comparing to conventional graph Laplacian-based convolution matrix in the encoder. Future efforts in this direction would include incorporation of other similarity measures into our framework, consensus learning of these proximity measures all together, and their applications, such as node classification and clustering.

Acknowledgements

We would like to thank Kaan Yorgancıoğlu and Serhan Yılmaz from Case Western Reserve University for their valuable discussion in this paper.

Funding

Not declared yet .

References

Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. *Social networks*, **25**(3), 211–230.

Bodenreider, O. (2004). The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, **32**(suppl_1), D267–D270.

Cao, M. *et al.* (2014). New directions for diffusion-based network prediction of protein function: incorporating pathways with confidence. *Bioinformatics*, **30**(12), i219–i227.

Coşkun, M. and Koyutürk, M. (2015). 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.

Coskun, M. (2019). Graph convolutional networks meet with high dimensionality reduction. *arXiv preprint arXiv:1911.02928*.

Cowen, L. *et al.* (2017). Network propagation: a universal amplifier of genetic associations. *Nature Reviews Genetics*, **18**(9), 551.

Davis, A. P. *et al.* (2019). The comparative toxicogenomics database: update 2019. *Nucleic acids research*, **47**(D1), D948–D954.

Devkota, K. *et al.* (2020). Glide: combining local methods and diffusion state embeddings to predict missing interactions in biological networks. *Bioinformatics*, **36**(Supplement_1), i464–i473.

Erten, S. *et al.* (2011a). D a d a: Degree-aware algorithms for network-based disease gene prioritization. *BioData mining*, **4**(1), 19.

Erten, S. *et al.* (2011b). Vavien: an algorithm for prioritizing candidate disease genes based on topological similarity of proteins in interaction networks. *Journal of computational biology*, **18**(11), 1561–1574.

Gilmer, J. *et al.* (2017). Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org.

Gottlieb, A. *et al.* (2011). Predict: a method for inferring novel drug indications with application to personalized medicine. *Molecular systems biology*, **7**(1).

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Hamilton, W. L. *et al.* (2019). Representation learning on graphs: methods and applications (2017). *arXiv preprint arXiv:1709.05584*.

Kipf, T. N. and Welling, M. (2016a). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kipf, T. N. and Welling, M. (2016b). Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Lei, C. and Ruan, J. (2013). A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity. *Bioinformatics*, **29**(3), 355–364.

Li, Q. *et al.* (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Liang, X. *et al.* (2017). Lrssl: predict and interpret drug–disease associations based on data integration using sparse subspace learning. *Bioinformatics*, **33**(8), 1187–1196.

Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, **58**(7), 1019–1031.

Linsker, R. (1988). Self-organization in a perceptual network. *Computer*, **21**(3), 105–117.

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

Pandey, J. *et al.* (2008). Functional coherence in domain interaction networks. *Bioinformatics*, **24**(16), i28–i34.

Perozzi, B. *et al.* (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Ribeiro, L. F. *et al.* (2017). struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 385–394.

Stanfield, Z. *et al.* (2017). Drug response prediction as a link prediction problem. *Scientific reports*, **7**, 40321.

Szklarczyk, D. *et al.* (2015). String v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic acids research*, **43**(D1), D447–D452.

Tang, J. *et al.* (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077.

Valdeolivas, A. *et al.* (2019). Random walk with restart on multiplex and heterogeneous biological networks. *Bioinformatics*, **35**(3), 497–505.

Veličković, P. *et al.* (2019). Deep graph infomax. *7th International Conference on Learning Representations (ICLR 2019)*.

Wang, Y.-B. *et al.* (2017). Predicting protein–protein interactions from protein sequences by a stacked sparse autoencoder deep neural network. *Molecular BioSystems*, **13**(7), 1336–1344.

Wishart, D. S. *et al.* (2018). Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, **46**(D1), D1074–D1082.

Wu, F. *et al.* (2019). Simplifying graph convolutional networks. In *ICML*.

Yoo, B. *et al.* (2015). Improving identification of key players in aging via network de-noising and core inference. *IEEE/ACM transactions on computational biology and bioinformatics*, **14**(5), 1056–1069.

Yue, X. *et al.* (2020). Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics*, **36**(4), 1241–1251.

Zhang, W. *et al.* (2018). Manifold regularized matrix factorization for drug-drug interaction prediction. *Journal of biomedical informatics*, **88**, 90–97.

Zhou, T. *et al.* (2009). Predicting missing links via local information. *The European Physical Journal B*, **71**(4), 623–630.