Predicting gene-disease associations via graph embedding and graph convolutional networks

Lvxing Zhu @, Zhaolin Hong @, Haoran Zheng*
School of Computer Science and Technology
University of Science and Technology of China
Hefei, China
@These two authors contribute equally to the work.
*Corresponding Author: hrzheng@ustc.edu.cn

***Abstract*—Identifying disease-related genes provides essential information for physiology, pathology and pharmacology. With the rapid growth of high-throughput sequencing technology and genome-wide association studies, a large number of gene-disease associations have been accumulated. These associations, coupled with the existing gene-related and disease-related databases, enable prediction of unknown associations by computational approaches. In this article, we proposed a new graph-based machine learning framework to predict the disease related genes. We first constructed a heterogeneous gene-disease association graph by integrating multiple biomedical knowledge bases, and subsequently processed the graph by a method that utilizes graph embedding representation and graph convolutional networks to learn the gene-disease associations. In our method, we defined a novel cluster loss function and a dropout mechanism to improve the generalization ability.We conducted experiments on DisGeNet dataset in 10-fold cross validation. The experimental results suggested that our method significantly outperformed the existing methods according to the performance of gene prioritization. The performance comparison of the variations of our method showed the efficacy of each module in our method.**

***Index Terms*—gene-disease associations, machine learning, graph embedding, graph convolutional networks**

I. Introduction

Exploring genes related to a given disease provides essential information on disease biology, pathology and pharmacology [1]. In recent decades, a large quantity of gene-disease associations were found and recorded, due to the rapid development of high-throughput sequencing technology and genome-wide association studies [2]. However, the records of gene-disease associations are still inadequate and the identification of new associations is still expensive and time-consuming [3]. Thus in silico computation approaches for predicting candidate gene-disease associations are urgently needed. The "guilt-by-association" principle [4], which indicates that similar phenotypes arise from functionally related genes, makes it feasible to infer unknown associations between genes and disease by existing gene-disease associations. Therefore, exploiting disease related genes by computational methods has becomes an attractive research area in recent years.

This work was supported by the National Key Technologies R&D Program [2017YFA0505502] and the Natural Science Foundation of Anhui Province [1508085MF128].

Existing published methods for prediction can be summarized into these categories: (1) Matrix decomposition. [5] used an inductive low-rank matrix decomposition for gene-disease associations matrix, and predicted unknown associations by matrix recovery from the decomposed matrices. The feature matrices of the genes and diseases were derived from multiple sources including microarray measurement, gene-gene network, disease similarity and gene-phenotype associations. (2) Network propagation. [6] proposed a network propagation method, which propagates gene prior scores on protein-protein network, and [7] used propagation on a heterogeneous network. A variation of network propagation by [8] employed a random walk with restart algorithm on the phenotype-gene bi-layer network. This approach constructed a bi-layer network by similarity network confusion from five individual gene (protein) similarity networks to overcome the low coverage of single PPI networks. (3) Shallow machine learning. [9] developed an approach for prediction of gene-disease associations by a boosted tree regression; the features used in the boosted tree were derived from the gene-gene mutual information from the known gene-disease associations and a known protein-protein network. Gene2DisCo [10] used disease-genes, gene interaction networks and disease similarities to provide the features and applied generalized linear models to prioritize the genes. (4) Graph embedding. HerGePred [11] embedded the genes and diseases into low-dimension vectors by Node2vec [12] on a heterogeneous network, and predicted the disease genes via a random walk with restart on a reconstructed disease-gene network, which was built by cosine similarity between the embedding vectors. The original heterogeneous network in HerGePred is built by integrating multiple disease-gene-related datasets that includes DisGeNet [13], protein-protein interactions and disease-symptom associations.

The common materials for predicting are classified as follows: (1) Gene-disease associations, such as DisGeNet [13], Malacards [14] and OMIM [15]. (2) Protein-protein interactions (PPIs), such as HIPPIE [16], STRING [17] and [18], [19]. (3) Gene functional network, such as HumanNet [20], [21]. (4) Disease-symptom associations, such as HPO [22] and Orphanet [23]. (5) Gene-phenotype associations [24], [25]. (6) Disease similarity network [26].

The success of the existing methods suggested that hetero-

978-1-7281-1867-3/19/$31.00 ©2019 IEEE
978-1-7281-1867-3/19/$31.00 ©2019 IEEE

geneous graph (network), which merges multiple data sources, is an efficient way to support the prediction of associated genes. On the other hand, various graph-based machine learning algorithms have been introduced recently, such as graph representation learning [12], [27] and graph convolutional network (GCN) [28], [29]. These graph-based models have shown the ability to handle the link prediction task in graph data (e.g. social networks, citation networks and telecommunication networks) scalably and universally. Therefore, the application of these models in gene-disease graph is expected to perform well.

Here, we propose a method that applies graph embedding representation and GCN on a heterogeneous gene-disease graph to predict the gene-disease associations, the overall process of our method is demonstrated in Figure 1. Initially, we build a heterogeneous graph that integrates multiple authority data sources that depict the gene-gene network, disease-disease network and gene-disease network. Then, graph embedding is developed to learn the representations of the genes and diseases and GCN is applied to extract the features from the representations in the graph. To ensure the generalization ability, we add *cluster loss* and *dropout of adjacent matrix* into training. Finally, we feed the feature vectors to a decoder to predict the associations between the genes and diseases. We conducted experiments on a standard dataset to verify the reasonableness and efficacy of our method.

The contributions of this article are summarized as follows:

- Describes the first attempt to integrate graph representation learning and graph convolutional networks aimed to predict gene-disease associations.
- Proposes a novel cluster loss function and dropout of adjacent matrix to enhance generalization ability.
- Significantly improves the state-of-the-art performance of gene-disease association prediction task.

II. Methods

A. Datasets

The heterogeneous graph we built contains two types of vertices representing gene and disease, and three types of edges corresponding to the gene-disease linkage, gene-gene linkage and disease-disease linkage. We collected three types of linkages from three datasets: DisGeNet [13], HumanNet [20] and Mesh vocabulary [30].

We collected the gene-disease linkages from DisGeNet. DisGeNet is a database of gene-disease associations, containing one of the largest publicly available collections of the genes and variants associated with human diseases. The records in DisGeNet are integrated from the expert-curated repositories, Genome-wide association studies (GWAS) catalogues, animal models and scientific literature. The DisGeNet associations forms gene-disease sub-graph $G^{gd} = (V^{gd}, E^{gd})$, where the set of vertices $V^{gd} = \{v_1^{gd}, v_2^{gd}, ..., v_{N+M}^{gd}\}$ represent all genes and diseases, and the set of edges E^{gd} represent all the associations. The adjacent matrix of G^{gd} is $\mathbf{A}^{gd} \in \mathbb{R}^{N\times M}$, where $\mathbf{A}_{ij}^{gd} = \mathbf{A}_{ji}^{gd} = 1$ for each $(v_i^{gd}, v_j^{gd}) \in E^{gd}$ and otherwise $\mathbf{A}_{ij}^{gd} = 0$, N and M are the numbers of the genes and diseases, respectively.

The gene-gene linkages are derived from HumanNet. HumanNet is a probabilistic functional gene network for *Homo sapiens* genes, which are constructed by a modified Bayesian integration of multiple omics data. Each interaction in HumanNet has a score that indicates the probability of the functional linkage between two genes. The functional gene network forms gene-gene sub-graph $G^{gg} = (V^{gg}, E^{gg})$, where V^{gg} is the set of all genes and E^{gg} are all linkages in the network. The adjacent matrix of G^{gg} is $\mathbf{A}^{gg} \in \mathbb{R}^{N\times N}$, where $\mathbf{A}_{ij}^{gg} = \mathbf{A}_{ji}^{gg} = w$ for each $(v_i^{gg}, v_j^{gg}) \in E^{gg}$ and w is the weight of the linkage provided by HumanNet.

The disease-disease linkages are inferred by MeSH vocabulary of diseases. MeSH provides the hierarchy-tree-structure catalogue of disease descriptors, which can be described as a directed acyclic graph (DAG). However, the linkage in the DAG only represents an "is-a" relation from a parent node to a child node. To obtain meaningful disease-disease linkages, we computed semantic similarity of two diseases according to their relative location in DAG according to the method described in [31]. Thus, we obtained a disease-disease sub-graph $G^{dd} = (V^{dd}, E^{dd})$, where V^{dd} is the set of all diseases and E^{dd} are all linkages in the network. The adjacent matrix of G^{dd} is $\mathbf{A}^{dd} \in \mathbb{R}^{M\times M}$, where $\mathbf{A}_{ij}^{dd} = \mathbf{A}_{ji}^{dd} = s$ for each $(v_i^{dd}, v_j^{dd}) \in E^{dd}$ and s is the similarity value.

B. Graph Representation

We integrated three sub-graphs into a heterogeneous gene-disease network $G = (V, E)$, where $V = V^{gd}$ and $E = E^{gd} \cup E^{gg} \cup E^{dd}$. The vertices in different sub-graphs are aligned according to the entity mapping. The adjacent matrix of G is $\mathbf{A} \in \mathbb{R}^{(N+M)\times(N+M)}$ defined as follows:

$$\mathbf{A} = \begin{bmatrix} \tilde{\mathbf{A}}^{gg} & \mathbf{A}^{gd} \\ (\mathbf{A}^{gd})^T & \tilde{\mathbf{A}}^{dd} \end{bmatrix} \tag{1}$$

where $\tilde{\mathbf{A}}^{gg}$ and $\tilde{\mathbf{A}}^{dd}$ are the reweight matrix of $\mathbf{A}^{gg}$ and $\mathbf{A}^{dd}$. The $\tilde{\mathbf{A}}^{gg}$ is computed as:

$$\tilde{\mathbf{A}}_{ij}^{gg} = \phi^i \tilde{\mathbf{A}}_{ij}^{gg} \tag{2}$$

where:

$$\phi^i = \phi \frac{\sum_j \mathbf{A}_{ij}^{gd}}{\sum_j \mathbf{A}_{ij}^{gg}} \tag{3}$$

where ϕ is normalized coefficients. $\tilde{\mathbf{A}}^{dd}$ is computed in the same manner. We then applied *Deepwalk* [27] on the heterogeneous network $G = (V, E)$ to learn the representations of the gene and disease vertices. Deepwalk generates the sequences of vertices $[v_1, v_2, ..., v_n]$ stochastically, where v_{i+1} is a vertex chosen randomly from the neighbours of vertex v_i and the probability of choosing each neighbour is proportional to the weight of the corresponding edge in $\mathbf{A}$. Since the adjacent matrix $\mathbf{A}$ is normalized, the ratio of probabilities of jumping to a homogeneous vertex and jumping to a heterogeneous vertex remains fixed. We used Deepwalk to generate λ-length sequences and to start t times at each vertex.

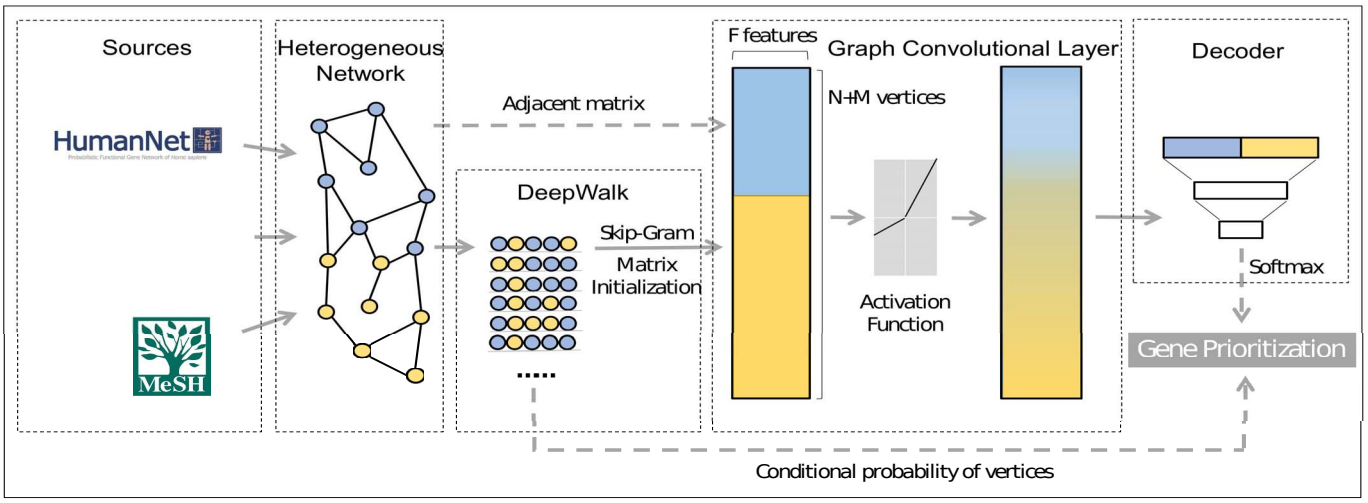

Fig. 1. The overall process of our method

Skip-gram [32], a Word2vec model, was trained on the given sequences of the vertices to obtain the embedding representation vectors and the probability distribution of the vertices. A representation vector encodes structural features of a graph into a continuous low-dimensional space, by maximizing the conditional probability $P(v_c|v_i)$, where v_c is the vertex that appear within the context window of v_i. The loss function for v_i during training is:

$$\begin{aligned}\mathcal{L}_{vi} &= -\,logP(v_{c1}, v_{c2}, ..., v_{cW}|v_i) \\ &= -\,log\prod_{j=1}^{W} P(v_{cj}|v_i) = -log\prod_{j=1}^{W}\frac{exp(\mathbf{X}_{c1}^T\cdot\mathbf{X}_j)}{\sum_{k=1}^{|v|}exp(\mathbf{X}_k^T\cdot\mathbf{X}_i)}\end{aligned} \tag{4}$$

where W is the window size, and $\mathbf{X}_i$ is a representation vector for the i-th vertex. The vectors of all the vertices compose the representation matrix $\mathbf{X}\in\mathbb{R}^{(N+M)\times C}$, where C is the vector length.

C. Graph Convolutional Network

We obtained the heterogeneous gene-disease network $G=(V,E)$ and the adjacent matrix $\mathbf{A}$; then, the graph convolutional network (GCN) was applied to extract the features for each gene and disease vertex. The graph normalized Laplacian matrix was defined as $\mathbf{L}$, where $\mathbf{L}=\mathbf{I}-\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ and $\mathbf{D}$ is the diagonal degree matrix of $\mathbf{A}$. $\mathbf{U}$ and $\mathbf{\Lambda}$ were defined as the eigenvectors and eigenvalues of $\mathbf{L}$, respectively.

Convolution operation can efficiently extract high-level features from time series and images [33]. Due to the difficulty of expressing convolution in the spatial domain of the graph, the convolution operator on graph was calculate via the Fourier domain [28]. To obtain a simplified calculation form and localized filters in the spatial domain, we used the *Chebyshev polynomial* for approximation as suggested by [34]. The convolution operation for signal $\mathbf{x}\in\mathbb{R}^{(N+M)}$ on graph G can be written as:

$$\mathbf{y}=\sum_{k=0}^{K-1}\theta_k\mathbf{U}T_k(\tilde{\mathbf{\Lambda}})\mathbf{U}^T\mathbf{x}\approx\theta\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}x \tag{5}$$

where θ_k is Chebyshev coefficient and $T_k(\tilde{\mathbf{\Lambda}})\in\mathbb{R}^{(N+M)\times(N+M)}$ is Chebyshev polynomial of order k, $\tilde{\mathbf{\Lambda}}=2\mathbf{\Lambda}/\lambda_{max}-\mathbf{I}$. The second part of Equation (5) serves to further simplify the computation form and alleviate overfitting, where we approximated $\lambda_{max}\approx 2$ and truncated Chebyshev polynomial at order $k=1$; $\tilde{\mathbf{A}}=\mathbf{A}+\mathbf{I}$ and $\tilde{\mathbf{D}}_{ij}=\sum_j\tilde{\mathbf{A}}_{ij}$ according to [29]. The signal at each vertex in our task is a vector and therefore, we generalized Equation (5) to signal $\mathbf{X}\in\mathbb{R}^{(N+M)\times C}$ with C input channels and extended the parameter θ to $\mathbf{\Theta}\in\mathbb{R}^{C\times F}$ with F filters. We also added a nonlinear activation function $\sigma()$ after convolution. Finally, a convolutional layer is computed by:

$$\mathbf{Y}=\sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\mathbf{\Theta}) \tag{6}$$

A convolution layer can be viewed as the information from the neighbouring vertices aggregated through the learnable parameter matrix [28] and then transformed by the activation function. We stacked two convolutional layers with a shortcut connection [35] from the input of the first layer. The representation matrix $\mathbf{X}$ is also a learnable parameter matrix that is initialized by Deepwalk in the previous subsection. The output of the two-layer convolutional network is defined as $\mathbf{Y}\in\mathbb{R}^{(N+M)\times F}$, where $\mathbf{Y}_i$ is the feature vector of the i-th vertex.

D. Decoding and Training

Since we obtained the feature vectors of each disease and gene, a three-layer dense layer was adopted as a decoder to predict the associations of the disease-gene pairs. For the i-th disease and the j-th gene, we concatenate their feature vectors into $\mathbf{y}^{ij}=[\mathbf{Y}_i,\mathbf{Y}_j]$ and feed the vector to the dense layers:

$$\mathbf{r}^{ij} = \mathbf{W}_3(f(\mathbf{W}_2(f(\mathbf{W}_1\mathbf{y}^{ij} + \mathbf{b}_1)) + \mathbf{b}_2)) + \mathbf{b}_3 \quad (7)$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3$ and $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ are the weight matrices and bias vectors of three layers, respectively, and $f()$ is the activation function. Finally, the vector $\mathbf{r}^{ij} \in \mathbb{R}^2$ is fed into a softmax layer as follows :

$$\mathbf{p}_k^{ij} = \frac{exp(\mathbf{r}_k^{ij})}{\sum_m exp(\mathbf{r}_m^{ij})}, \quad k = 0, 1 \quad (8)$$

where $\mathbf{p}_1^{ij}$ and $\mathbf{p}_0^{ij}$ is the probability of that the i-th disease and the j-th gene have an association or do not have an association. During training, we update the parameters of the networks by optimizing the total loss function:

$$\mathcal{L} = -\sum_{\substack{(v_i,v_j)\in \\ E_{all\_target}}} (\sum_{k\in 0,1} l_k^{ij} ln \mathbf{p}_k^{ij}) + \alpha Cluster\_Loss(\mathbf{X}) \quad (9)$$

where $E_{all\_target} = E_{target} \cup E_{nega\_target}$ is the set of all training samples. Each sample in E_{all_target} has a one-hot label l^{ij} that denotes whether it is a positive sample or a negative sample. The positive samples E_{target} are derived from the edges in E (see Chapter 2.6) while the negative samples E_{nega_target} are generated by randomly replacing the vertex of the positive samples; we generated the $nega$ samples for each positive sample. The first term of Equation (9) is the *NLLloss* for supervised learning; the second term is the cluster loss that is defined on representation matrix $\mathbf{X}$ and α is a reweight for the two terms. An end-to-end optimization was then conducted to learn the network parameters of the GCN and the decoder by gradient propagation.

E. Cluster Loss

We introduced a cluster loss into training of GCN. The goal of the cluster loss is to promote the representation vectors of the vertices with similar functions to be closer, i.e., the vertices with similar properties should form a local cluster in the representation vector space. The cluster loss is motivated by the "guilt-by-association" principle [4]. The principle indicates that diseases with similar phenotypes arise from functionally similar genes. Therefore, we let the similar diseases have close vectors and so do functionally similar genes. We expected that the principle can yield more reasonable distribution of the vectors and improve the generalization ability.

The cluster loss makes the representation vectors closer to their nearest cluster through training. Considering that the initial representation vector $\mathbf{X}$ encodes the functional information for each disease and gene, a cluster of vectors $\mathbf{X}$ can be viewed as a set of similar genes or similar diseases. Thus the cluster loss is calculated by summing the L2 distance of each vertex to its cluster center $\mathbf{Cluster}^j$ measured by the representation vectors. The loss is written as:

$$\begin{aligned} Cluster\_Loss(\mathbf{X}) &= \sum_{i\in V} \parallel \mathbf{X}_i - \mathbf{Cluster}^{j|i\in j^{th}cluster} \parallel_2^2 \\ &= \sum_{i\in V} \parallel \mathbf{X}_i - \overline{\{\mathbf{X}_k | k \in j^{th}cluster\}} \parallel_2^2 \end{aligned} \quad (10)$$

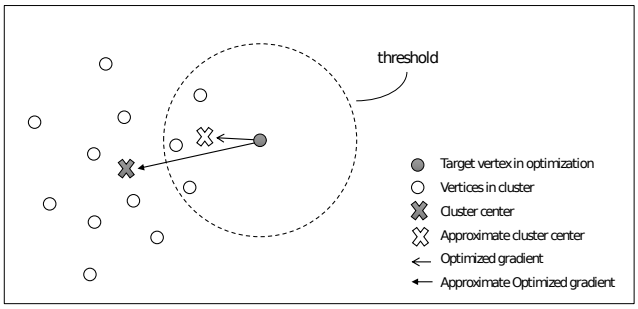

Fig. 2. The principle for approximation of cluster loss

However, the clustering across all vertices is time-consuming, so we take an approximation method here. Instead of calculating each cluster, we use the average position of the neighbour vertices within a limited distance to calculate cluster loss approximately. The approximation is demonstrated in Figure 2. The set of neighbour vertices is filtered by threshold t_s derived from cosine similarity between the representation vectors. Thus Equation (10) can be written as:

$$\begin{aligned} Cluster\_Loss(\mathbf{X}) &\approx \sum_{i\in V} \parallel \mathbf{X}_i - \overline{\{\mathbf{X}_k | Cosine(\mathbf{X}_k, \mathbf{X}_i) > t_s\}} \parallel_2^2 \\ &= \parallel \mathbf{X} - (\mathbf{S}^{cosine} > t_s)^{row\_norm}\mathbf{X} \parallel_2^2 \end{aligned} \quad (11)$$

where $\mathbf{S}^{cosine}$ is the cosine similarity matrix of $\mathbf{X}$ where $\mathbf{S}_{ij}^{cosine} = Cosine(\mathbf{X}_i, \mathbf{X}_j)$. $(\mathbf{S}^{cosine} > t_s)$ is a binary matrix consisting of 0 and 1 and $(\mathbf{S}^{cosine} > t_s)^{row\_norm}$ is the row-normalization matrix of it. Equation (11) converts the loss function into a couple of matrix computations that can be efficiently accelerated by a GPU.

F. Dropout of Adjacent Matrix

We introduced dropout of adjacent matrix to prevent the convolutional networks from overfitting. It is different from regular dropout [36], which randomly drops units of neural networks, the dropout of adjacent matrix randomly drops the edges in adjacent matrix $\mathbf{A}$.

The set of edges E is split into the training set E_{train} and the test set E_{test} by n-fold cross validation. In dropout of adjacent matrix, we further randomly split E_{train} into two subsets E_{base} and E_{target} in a fraction of $p : 1 - p$ at each training epoch. In the training phase, the adjacent matrix $\mathbf{A}$ in GCN is only composed of E_{base}, and E_{target} is used as the training samples. In the test phase, we take full training set E_{train} to build adjacent matrix $\mathbf{A}$ and the elements in $\mathbf{A}$ are reweighted by p.

The dropout of an adjacent matrix ensures that the information of the training samples is not contained in an adjacent matrix; therefore, the overfitting can be avoided. The process can be considered as training of different models on a "thinned" graph and unknown samples are predicted by the ensemble of these "thinned" models.

G. Gene Prioritization

A common demand of the gene-disease prediction task is "gene prioritization", i.e., output of a candidate gene set for a given disease. These candidate genes are associated with the disease with relatively high probability and are sorted by the probability. Here, we derived the sorted candidate genes by a two-step approach that uses the output of the GCN decoder and the probability distribution derived from Deepwalk.

Initially, we filter all genes by the output of the GCN decoder. Given the i-th disease, we enumerated all valid gene-disease pairs and recorded their log probability $\mathbf{p}^{ij}$ from Equation (8). Then, a candidate gene set was obtained by setting a threshold t_g, which is defined as $G_{set} = \{v_j|\mathbf{p}_1^{ij} > t_g\}$. Generally, the candidate set is small and contains only dozens of genes.

Then, we sorted all genes in the G_{set} by the conditional probability $P(v_j|v_i)$ from Skip-gram. The probability is derived from Equation (4) where v_i is the vertex of a given disease and $v_j \in G_{set}$. The sorted sequence for disease d was denoted as $P(d) = [v_1, v_2, ..., v_m]$.

III. Results and Discussion

A. Experimental Settings

The experimental dataset was collected from DisGeNet [13]. We followed the settings as proposed in HerGePred [11]; only curated associations are used for 10-fold cross validation that include a total of 130,821 gene-disease associations between 8,948 genes and 13,074 diseases. The rest of DisGeNet was used for an external dataset. For the gene-gene subgraph, a total of 151,302 linkages were collected from HumanNet [20]. 695,559 disease-disease linkages were derived from MeSH [30] filtered by a similarity threshold.

We have done an optimization of all the hyperparameters,the main hyper parameterswhich were tuned according to the k fold cross validation were set as follows: the similarity threshold of MeSH is 0.2, the size of the representation vectors and feature vectors were 128, the window size of Skip-gram were 10, the normalized coefficient ϕ was 0.2, the weight for cluster loss α was 1e-4, the threshold t_s in cluster loss was 0.8 and $nega$ was 10. The activation function was Relu and optimizer was Adam. We trained Skip-gram for 5 iterations; the GCN and the decoder was trained through 100 epochs, which were implemented in pytorch. The curves of total loss and cluster loss of the GCN and the decoder are shown in Figure 3.

We used "gene prioritization" evaluation metrics, which sorted all candidate genes for a given disease and truncated Top-i genes to compare to the gold standard genes in the test set. We defined $T(d)$ as the genes associated with a disease d in the test set and $P_i(d)$ as the predicted genes for d in Top-i. Assuming N as the number of diseases in test set, the average recall, precision, F1 score and AP are defined as follows:

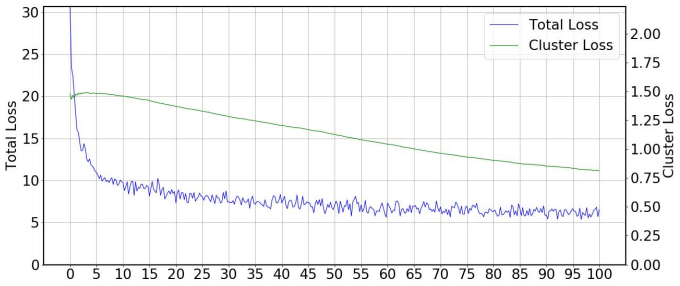

Fig. 3. Loss sequence of 100-epochs training

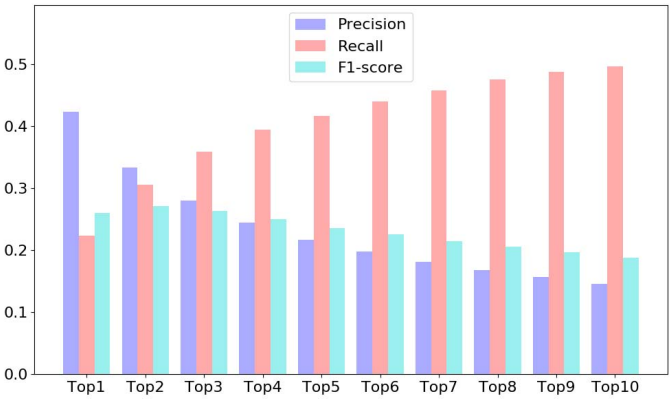

Fig. 4. Bar graph of recall, precision and F1-score for predicted Top-i genes of our method

$$
\begin{aligned}
Prec_i &= 1/N \sum_d \frac{|T(d) \cap P_i(d)|}{|P_i(d)|}, \quad i = 1, 2, .., 10 \\
Recall_i &= 1/N \sum_d \frac{|T(d) \cap P_i(d)|}{|T(d)|}, \quad i = 1, 2, .., 10 \\
F1score_i &= 1/N \sum_d \frac{2|T(d) \cap P_i(d)|}{|P_i(d)| + |T(d)|}, \quad i = 1, 2, .., 10 \\
AP &= 1/N \frac{\sum_d |T(d) \cap P_k(d)|}{\sum_d |P_k(d)|}, \quad k = |T(d)|
\end{aligned}
\tag{12}
$$

B. Performance

The average precision, recall and F1 score of Top-i predicted genes are shown in Figure 4, where i range is from 1 to 10. In Figure 4, the precision decreases with an increase in i. The maximum of precision is reached at $i = 1$ and the value is 0.424 meaning that the average probability that a Top-1 predicted gene is a real associated gene is 42.4%. Meanwhile, the recall increases with an increase in i. The recall of $i = 10$ has the maximum value 0.497, which represents that there are 49.7% real associated genes included in the Top-10 predicted genes. The maximum of F1 score is reached at $i = 2$ and the value is 0.268. It should be noted that the precision, recall and F1 score is calculated for each disease independently and then

averaged (Equation (12)); thus, the F1 score here is lower than the scores of precision and recall.

The performance of our method, variants of our method and other existing methods are shown in Table I. The methods are summarized as follows. pgWalk [7] is a network propagation method on multiple disease-gene networks. GUILD [37] is based on the topology of the protein-protein network. LVRSim-EmbDGG [11] prioritizes genes by cosine similarity of embedding vectors. RW-RDGN-EmbDGG [11] propagates the similarity on reconstructed networks. In the case of "Without Deepwalk", our proposed method lacks the initialization by the Deepwalk representation vectors. In "Without GCN", we did not use GCN and decoded the Deepwalk representation vectors directly. In "Without C&D", we removed the cluster loss and dropout of adjacent matrix from the proposed method. "Decoder Sorted" prioritized genes only by the softmax output from the decoder. "Skip-gram Sorted" is sorted by condition probability from Skip-gram.

As shown in Table I, our method outperformed other methods by a significant margin and achieved state-of-the-art performance in the AP, precision, recall and F1 score of the Top-3 genes and recall and F1 score of Top-10 genes. Our method obtained a 0.411 AP that is significantly higher than the values obtained by other methods. The Top-3 precision and recall of our method were 0.283 and 0.361, respectively, indicating that the Top-3 genes for each disease can cover 36.1% real associated genes on average and 28.3% of the Top-3 genes were real associated genes. For Top-10 genes, our method obtained a precision of 0.147 and a recall of 0.494 and the F1 score was 0.188. All performance indicators of our method had the highest scores except for the recall of Top-10, which was 0.494 and is somewhat lower than the recall of the Skip-gram Sorted, a variants of our method which scored 0.496.

In general, other variation methods did not perform better than our proposed method suggesting that all modules and characteristics added to our method efficiently enhanced the performance. The performance of "Without GCN" and "Without C&D" declined by approximately 10% in general compared to the best performance indicating that GCN contributed a positive effect and cluster loss improved the generalization ability. The scores of "Without Deepwalk" were markedly decreased suggesting that the initialization vectors derived from Deepwalk provided crucial information and GCN without initialization can hardly converge to a global minimum. Decoder Sorted and Skip-gram Sorted used single sort criteria and achieved the performance close to best method; however, the merge of two modules can perform better.

We further evaluated these methods using an external dataset of DisGeNet; the results are shown in Table II. Our method achieved the best recall and F1 score of the Top-3 genes and Top-10 genes, and the precision of Top-3 and Top-10 was slightly lower than the best performance of existing methods. Furthermore, our method was superior to the variation methods. The results demonstrated the efficacy and robustness of our method. Additionally, the performance in an external dataset was markedly lower than the performance in the test set because external dataset contained numerous unknown genes and the overlapped associations with the training and test sets have been removed.

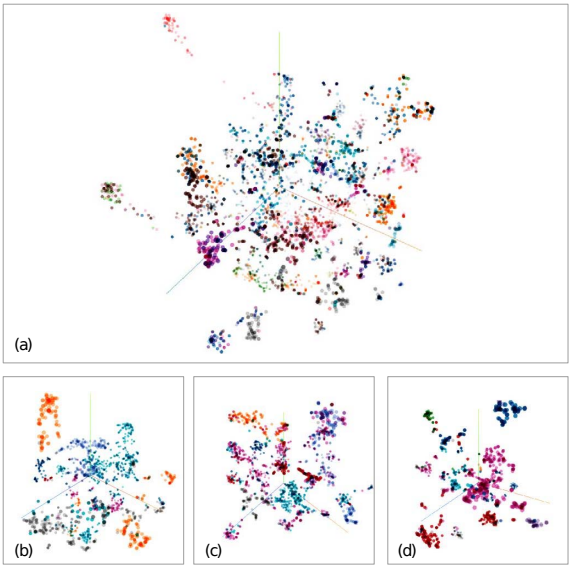

Fig. 5. Visualized scatter diagram of disease vectors, colored with category from MeSH. Figure(a) is the scatter diagram of all 26 categories diseases. To clarify, we splited 26 categories into three subsets and draw corresponding diseases in figure(b), figure(c) and figure(d), respectively

To demonstrate the rationality of cluster loss, we plotted visualized vectors of all diseases as a scatter diagram in Figure 5. The dimension of the representation vectors were reduced by t-SNE [38] into three-dimension. The vertices are coloured according to their MeSH categories with a total of 26 categories. The category information is not explicitly input into our method; however, our method learned the category features of each disease spontaneously and made them form the clusters according to the features. The results suggest that our method can embed similar diseases or genes into similar vectors. This property is in agreement with the "guilt-by-association" principle and therefore, the cluster loss can be implemented in our method.

C. Case Study

We studied a prediction case for a given disease "hypothyroidism" to illustrate the application in a real biological scenario. Table III shows the Top-10 genes for hypothyroidism along with their predicted score. There are three real associated genes in the test set including *DUOX2*, *THRA* and *FOXI1*. Two of the genes were successfully predicted in the Top-10 genes (*DUOX2* and *THRA*) and the remaining gene obtained a 0.7684 score that ranked out of Top-10. Excluding for the two genes in the gold standard, other four genes in the Top-10 predicted genes were supported by text mining or certain studies including *DUOXA2*, *NKX2-1*, *FOXE1* and *FSHB*. Only four of the Top-10 predicted genes lack evidence to support the conclusions.

TABLE I
PERFORMANCE COMPARISON ON TEST SET OF DISGENET

Approach	AP	Top-3			Top-10		
		Prec	Recall	F1 Score	Prec	Recall	F1 Score
GUILD [37]	0.091	0.010	0.021	0.013	0.004	0.030	0.007
pgWalk [7]	0.258	0.222	0.305	0.219	0.105	0.416	0.145
LVRSim-EmbDGG [11]	0.239	0.200	0.264	0.191	0.101	0.386	0.136
RW-RDGN-EmbDGG [11]	0.294	0.243	0.325	0.233	0.124	0.477	0.167
Without DeepWalk*	0.100	0.093	0.067	0.064	0.056	0.174	0.069
Without GCN*	0.344	0.251	0.242	0.233	0.137	0.447	0.167
Without C&D*	0.391	0.258	0.324	0.239	0.138	0.453	0.173
Decoder Sorted*	0.377	0.259	0.309	0.234	0.132	0.431	0.166
Skip-gram Sorted*	0.390	0.270	0.348	0.255	0.142	**0.496**	0.184
Proposed method	**0.411**	**0.283**	**0.361**	**0.266**	**0.147**	0.494	**0.188**

TABLE II
PERFORMANCE COMPARISON ON EXTERNAL SET OF DISGENET

Approach	Top-3			Top-10		
	Prec	Recall	F1 Score	Prec	Recall	F1 Score
GUILD [37]	**0.144**	0.022	-	**0.124**	0.044	-
pgWalk [7]	0.135	0.018	-	0.116	0.039	-
LVRSim-EmbDGG [11]	0.100	0.014	-	0.089	0.035	-
RW-RDGN-EmbDGG [11]	0.132	0.018	-	**0.124**	0.045	-
Without DeepWalk*	0.054	0.009	0.008	0.051	0.025	0.017
Without GCN*	0.116	0.030	0.030	0.097	0.075	0.043
Without C&D*	0.130	0.035	0.042	0.101	0.072	0.048
Proposed method	0.136	**0.038**	**0.047**	0.107	**0.076**	**0.056**

TABLE III
PREDICTED CASE: TOP-10 GENES FOR HYPOTHYROIDISM

Gene ID	Gene Name	Predicted Score	Evidence
8022	LHX3	0.9777	-
405753	DUOXA2	0.9715	BeFree
50506	DUOX2	0.9560	True
10984	KCNQ1OT1	0.9248	-
7067	THRA	0.9205	True
7080	NKX2-1	0.9068	BeFree
2304	FOXE1	0.9048	BeFree,GAD
89884	LHX4	0.9041	-
2488	FSHB	0.8735	[39]
54361	WNT4	0.8703	-

* BeFree and GAD [40] are text mining system.
* True indicates the gene is contained in the test set.

IV. CONCLUSION

In this article, we presented a machine learning framework that is integrated with graph embedding representation and graph convolutional networks to predict gene-disease associations. We constructed a heterogeneous gene-disease graph from multiple biomedical databases and embedded vertices into the representation vectors; then, the features were extracted from these vectors by graph convolution operators. Our method took advantage of the embedding vectors that encodes the local and global features of the graph, and the message passed through the edges by GCN. A novel cluster loss and dropout of adjacent matrix were used in our method to alleviate overfitting. The evaluation results showed that our method achieved state-of-the-art performance compared to the existing methods suggesting the efficacy of each module and characteristic in our method. In the future work, the prior knowledge about genes and diseases and the edge features will be used to provide additional information. Since prioritization of the disease-related genes via computation approaches is an in-demand technology for physiology, pathology and pharmacology, we believe that the proposed method can facilitate various studies in real-world scenarios.

REFERENCES

[1] O. Vanunu, O. Magger, E. Ruppin, T. Shlomi, and R. Sharan, Associating genes and protein complexes with disease via network propagation, PLoS computational biology, vol. 6, no. 1, p.e1000641, 2010.

[2] T. A. Manolio, Genomewide association studies and assessment of the risk of disease, New England journal of medicine, vol. 363, no. 2, pp.166176, 2010.

[3] R. M. Piro and F. Di Cunto, Computational approaches to disease-gene prediction: rationale, classification and successes, The FEBS journal,vol. 279, no. 5, pp. 678696, 2012.

[4] C. J. Wolfe, I. S. Kohane, and A. J. Butte, Systematic survey reveals general applicability of guilt-by-association within gene coexpression networks, BMC bioinformatics, vol. 6, no. 1, p. 227, 2005.

[5] N. Natarajan and I. S. Dhillon, Inductive matrix completion for predicting genedisease associations, Bioinformatics, vol. 30, no. 12, pp. i60i68, 2014.

[6] Y. Qian, S. Besenbacher, T. Mailund, and M. H. Schierup, Identifying disease associated genes by network propagation, in BMC systems biology, vol. 8, no. 1. BioMed Central, 2014, p. S6.

[7] R. Jiang, Walking on multiple disease-gene networks to prioritize candidate genes, Journal of molecular cell biology, vol. 7, no. 3, pp. 214230, 2015.

[8] Z. Tian, M. Guo, C. Wang, L. Xing, L. Wang, and Y. Zhang, Constructing an integrated gene similarity network for the identification of disease genes, Journal of biomedical semantics, vol. 8, no. 1, p. 32, 2017.

[9] H. Zhou and J. Skolnick, A knowledge-based approach for predicting genedisease associations, Bioinformatics, vol. 32, no. 18, pp. 28312838, 2016.

[10] M. Frasca, Gene2disco: gene to disease using disease commonalities, Artificial intelligence in medicine, vol. 82, pp. 3446, 2017.

[11] K. Yang, R. Wang, G. Liu, Z. Shu, N. Wang, R. Zhang, J. Yu, J. Chen,X. Li, and X. Zhou, Hergepred: Heterogeneous network embedding representation for disease gene prediction, IEEE Journal of Biomedical and Health Informatics, 2018.

[12] 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. ACM, 2016, pp. 855864.

[13] J. Piero, . Bravo, N. Queralt-Rosinach, A. Gutirrez-Sacristn, J. Deu-Pons, E. Centeno, J. Garca-Garca, F. Sanz, and L. I. Furlong, Disgenet: a comprehensive platform integrating information on human disease-associated genes and variants, Nucleic acids research, p. gkw943, 2016.

[14] N. Rappaport, M. Twik, I. Plaschkes, R. Nudel, T. Iny Stein, J. Levitt, M. Gershoni, C. P. Morrey, M. Safran, and D. Lancet, Malacards: an amalgamated human disease compendium with diverse clinical and genetic annotation and structured search, Nucleic acids research, vol. 45, no. D1, pp. D877D887, 2016.

[15] A. Hamosh, A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick, Online mendelian inheritance in man (omim), a knowledge-base of human genes and genetic disorders, Nucleic acids research, vol. 33, no. suppl 1, pp. D514D517, 2005.

[16] M. H. Schaefer, J.-F. Fontaine, A. Vinayagam, P. Porras, E. E. Wanker, and M. A. Andrade-Navarro, Hippie: Integrating protein interaction networks with experiment based quality scores, PloS one, vol. 7, no. 2, p. e31826, 2012.

[17] D. Szklarczyk, A. Franceschini, S. Wyder, K. Forslund, D. Heller, J. Huerta-Cepas, M. Simonovic, A. Roth, A. Santos, K. P. Tsafou et al., String v10: proteinprotein interaction networks, integrated over the tree of life, Nucleic acids research, vol. 43, no. D1, pp. D447D452, 2014.

[18] U. Stelzl, U. Worm, M. Lalowski, C. Haenig, F. H. Brembeck, H. Goehler, M. Stroedicke, M. Zenkner, A. Schoenherr, S. Koeppen et al., A human protein-protein interaction network: a resource for annotating the proteome, Cell, vol. 122, no. 6, pp. 957968, 2005.

[19] A. Chatr-Aryamontri, R. Oughtred, L. Boucher, J. Rust, C. Chang, N. K. Kolas, L. ODonnell, S. Oster, C. Theesfeld, A. Sellam et al., The biogrid interaction database: 2017 update, Nucleic acids research, vol. 45, no. D1, pp. D369D379, 2017.

[20] I. Lee, U. M. Blom, P. I. Wang, J. E. Shim, and E. M. Marcotte, Prioritizing candidate disease genes by network-based boosting of genome-wide association data, Genome research, vol. 21, no. 7, pp. 11091121, 2011.

[21] G. Wu, X. Feng, and L. Stein, A human functional protein interaction network and its application to cancer data analysis, Genome biology, vol. 11, no. 5, p. R53, 2010.

[22] S. Kohler, N. A. Vasilevsky, M. Engelstad, E. Foster, J. McMurry, S. Ayme, G. Baynam, S. M. Bello, C. F. Boerkoel, K. M. Boycott et al., The human phenotype ontology in 2017, Nucleic acids research, vol. 45, no. D1, pp. D865D876, 2016.

[23] S. S. Weinreich, R. Mangon, J. Sikkens, M. Teeuw, and M. Cornel, Orphanet: a european database for rare diseases, Nederlands tijdschrift voor geneeskunde, vol. 152, no. 9, pp. 518519, 2008.

[24] R. A. Green, H.-L. Kao, A. Audhya, S. Arur, J. R. Mayers, H. N. Fridolfsson, M. Schulman, S. Schloissnig, S. Niessen, K. Laband et al., A high-resolution c. elegans essential gene network based on phenotypic profiling of a complex tissue, Cell, vol. 145, no. 3, pp. 470482, 2011.

[25] R. J. Nichols, S. Sen, Y. J. Choo, P. Beltrao, M. Zietek, R. Chaba, S. Lee, K. M. Kazmierczak, K. J. Lee, A. Wong et al., Phenotypic landscape of a bacterial cell, Cell, vol. 144, no. 1, pp. 143156, 2011.

[26] M. A. Van Driel, J. Bruggeman, G. Vriend, H. G. Brunner, and J. A.Leunissen, A text-mining analysis of the human phenome, European journal of human genetics, vol. 14, no. 5, p. 535, 2006.

[27] B. Perozzi, R. Al-Rfou, and S. Skiena, Deepwalk: Online learning of social representations, in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014, pp. 701710.

[28] M. Defferrard, X. Bresson, and P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in Advances in neural information processing systems, 2016, pp. 38443852.

[29] T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907, 2016.

[30] C. E. Lipscomb, Medical subject headings (mesh), Bulletin of the Medical Library Association, vol. 88, no. 3, p. 265, 2000.

[31] D. Wang, J. Wang, M. Lu, F. Song, and Q. Cui, Inferring the human microrna functional similarity and functional network based on microrna-associated diseases, Bioinformatics, vol. 26, no. 13, pp. 16441650, 2010.

[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781, 2013.

[33] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, nature, vol. 521, no. 7553, p. 436, 2015.

[34] D. K. Hammond, P. Vandergheynst, and R. Gribonval, Wavelets on graphs via spectral graph theory, Applied and Computational Harmonic Analysis, vol. 30, no. 2, pp. 129150, 2011.

[35] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770778.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research, vol. 15, no. 1, pp. 19291958, 2014.

[37] E. Guney and B. Oliva, Exploiting protein-protein interaction networks for genome-wide disease-gene prioritization, PloS one, vol. 7, no. 9, p. e43557, 2012.

[38] L. v. d. Maaten and G. Hinton, Visualizing data using t-sne, Journal of machine learning research, vol. 9, no. Nov, pp. 25792605, 2008.

[39] M. Garca, R. Barrio, M. Garca-Lavandeira, A. R. Garcia-Rendueles, A. Escudero, E. Daz-Rodrguez, D. G. Del Blanco, A. Fernndez, Y. B. De Rijke, E. Vallespn et al., The syndrome of central hypothyroidism and macroorchidism: Igsf1 controls trhr and fshb expression by differential modulation of pituitary tgf and activin pathways, Scientific reports, vol. 7, p. 42937, 2017.

[40] . Bravo, J. Piero, N. Queralt-Rosinach, M. Rautschka, and L. I. Furlong, Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research, BMC bioinformatics, vol. 16, no. 1, p. 55, 2015.