Paper Review For EmbeddingGCN
by Anurag Banerjee

Predicting gene-disease associations via graph embedding and graph convolutional networks[1]

Lvxing Zhu et. al.

1 Jargon Glossary

DAG - or, *Directed Acyclic Graph*, a graph with each edge having a direction, such that there are NO directed cycles, or closed loops

DNA Sequencing - The general laboratory technique for determining the exact sequence/pattern of nucleotides/bases (**A**denine, **C**ytosine, **G**uanine, and **T**hymine) in a DNA molecule

FFNN - Feed Forward Neural Network

GCN - Graph Convolution Network [2]

HMDD - or, *Human MicroRNA-associated Disease Database*, has experimentally supported associations for human micro-RNA and diseases

LDA - or, *Latent Dirichlet Allocation*, given multiple *documents* that contain permutations of *words* in a vocabulary, create topic (representative word) clusters, such that doc. belongs to some topic; an unsupervised, probabilistic model

Pathology - The science of causes and effects of diseases; specifically involves laboratory examination of samples of body tissue

PCA - or, *Principal Components Analysis*, find eigen vectors of the covariance matrix such that highly correlated variable set may be reduced to fewer un-correlated independent variables, in a co-ordinate system whose axes capture all the variance of the original variables; an unsupervised statistical model

Pharmacology - The scientific study of effect of drugs and chemicals on living organisms

Physiology - Physiology is the scientific study of functions and mechanisms in a living system

2 Problem Description

Identifying genes that contribute to a disease can further our understanding in human **physiology**, **pathology** and thus enable better research in **pharmacology**. In this paper, the authors have setup a heterogeneous graph of *genes* and *diseases* and utilised the techniques of graph embedding (DeepWalk + word2vec) followed by GCN [2] to learn embeddings for both the *gene nodes* and the *diseases nodes*. This is followed by a decoder module which learns to predict the probability of association between any *gene-disease* pair. An overview of all the components of the paper is summarised in the *mindmap* in Figure 1.

3 Problem Relevance

Most *in-silico* methods attempt to leverage the *guilt-by-association* principle wherein new gene-disease associations are predicted considering the functional similarity of the genes to known causative genes. Some of the common methods for this prediction are:

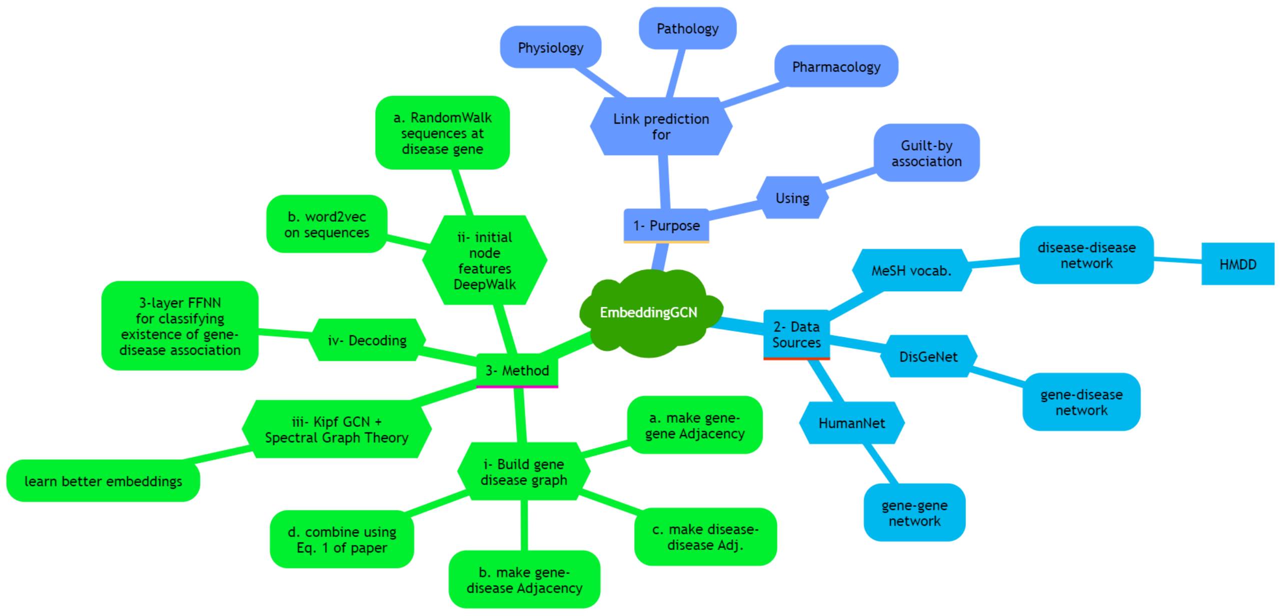

Physiology
Pathology
Pharmacology
Link prediction for
Guilt-by association
Using
1- Purpose
a. RandomWalk sequences at disease gene
b. word2vec on sequences
ii- initial node features DeepWalk
EmbeddingGCN
MeSH vocab.
disease-disease network
HMDD
2- Data Sources
3-layer FFNN for classifying existence of gene-disease association
iv- Decoding
3- Method
DisGeNet
gene-disease network
iii- Kipf GCN + Spectral Graph Theory
a. make gene-gene Adjacency
HumanNet
learn better embeddings
i- Build gene disease graph
gene-gene network
d. combine using Eq. 1 of paper
c. make disease-disease Adj.
b. make gene-disease Adjacency

Figure 1: Mindmap

1. *Matrix Decomposition*
2. *Network Propagation*
3. *Shallow Machine Learning*
4. *Graph Embedding*

Matrix Decomposition is one of the most common approaches where after decomposing a matrix of known gene-disease association, new associations are recovered by arranging the decompositions for unknown gene- disease pairs. *Network Propagation* relies on PPI graphs and *Shallow ML* methods involve boosted regression trees, again using PPI. *Graph Embedding* approaches are newer, and can incorporate heterogeneous graphs.

A heterogenous graph can simultaneously contain nodes of type **gene**, **disease**, *etc.* and the learning task can be setup directly as a *link prediction* problem, between the genes and diseases.

4 Proposed Solution

The authors have setup the downstream task as a link prediction problem, which roughly has 4 stages:

1. Build gene-disease graph
2. Initial node features
3. GCN
4. Decoding

4.1 Building Heterogeneous gene-disease Graph

As depicted in Figure 2 the primary data sources are **DisGeNet** for **gene-disease** linkages, **HumanNet** for **gene-gene** linkages and **MeSH** $+$ **HMDD** for creating the **disease-disease** linkages from the DAG of MeSH. The DAG is used in a process called MISIM [3] that generates scores that can be used for making the disease-only graph.

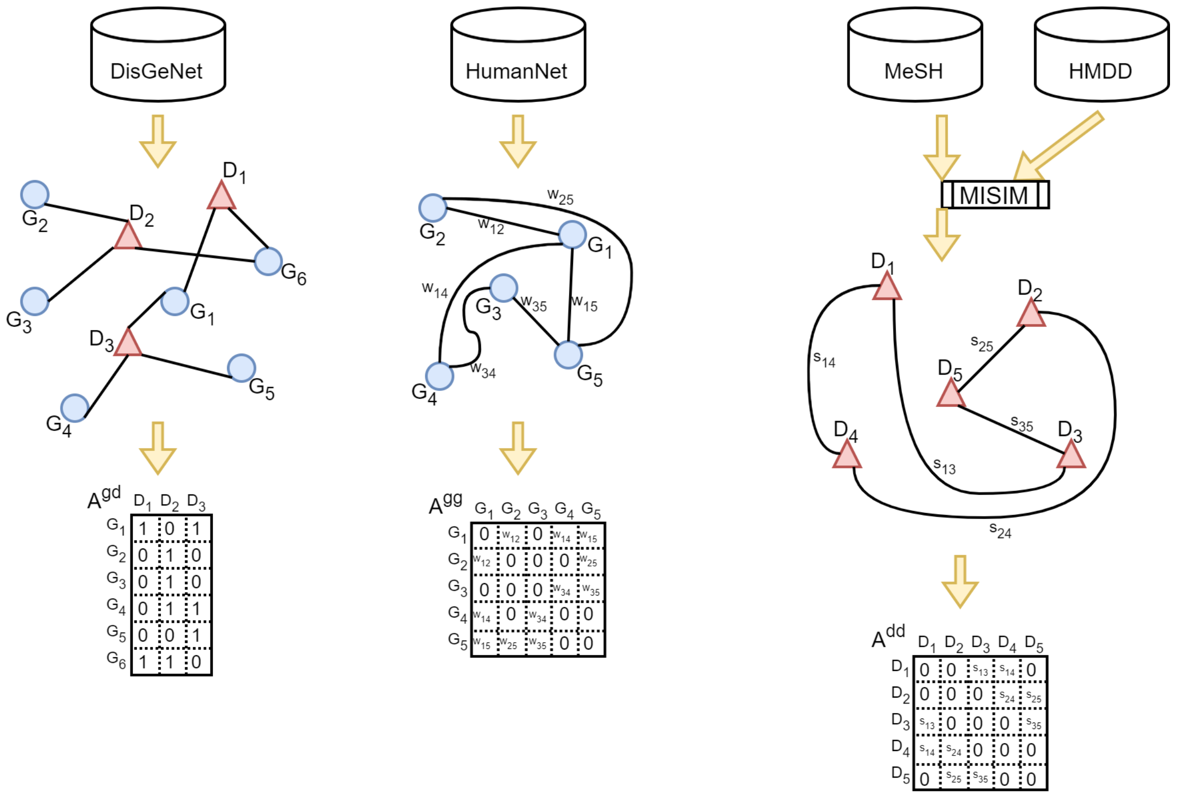
Once we have the above information, the final heterogenous graph is constructed as $G = (V, E)$, where $V = V^{gd}$ and $E = E^{gd} \cup E^{gg} \cup E^{dd}$. The adjacency matrix is constructed as follows (*Eq. 1, 2, 3 of the paper*):

Disclaimer: *The equations that follow need reconsideration. I have modified a term.*

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

where,

$$\tilde{A}^{gg}_{ij} = \phi^i A^{gg}_{ij}$$
$$\phi^i = \phi \frac{\sum_j A^{gd}_{ij}}{\sum_j A^{gg}_{ij}} \tag{2}$$


DisGeNet
HumanNet
MeSH
HMDD
MISIM
G2 D2 D1 G6 G3 G1 D3 G5 G4
G2 w25 w12 G1 w14 G3 w35 w15 G4 w34 G5
D1 D2 s25 s14 D5 s35 D4 D3 s13 s24

$$A^{gd} \quad \begin{array}{c|ccc} & D_1 & D_2 & D_3 \\ \hline G_1 & 1 & 0 & 1 \\ G_2 & 0 & 1 & 0 \\ G_3 & 0 & 1 & 0 \\ G_4 & 0 & 1 & 1 \\ G_5 & 0 & 0 & 1 \\ G_6 & 1 & 1 & 0 \end{array}$$

$$A^{gg} \quad \begin{array}{c|ccccc} & G_1 & G_2 & G_3 & G_4 & G_5 \\ \hline G_1 & 0 & w_{12} & 0 & w_{14} & w_{15} \\ G_2 & w_{12} & 0 & 0 & 0 & w_{25} \\ G_3 & 0 & 0 & 0 & w_{34} & w_{35} \\ G_4 & w_{14} & 0 & w_{34} & 0 & 0 \\ G_5 & w_{15} & w_{25} & w_{35} & 0 & 0 \end{array}$$

$$A^{dd} \quad \begin{array}{c|ccccc} & D_1 & D_2 & D_3 & D_4 & D_5 \\ \hline D_1 & 0 & 0 & s_{13} & s_{14} & 0 \\ D_2 & 0 & 0 & 0 & s_{24} & s_{25} \\ D_3 & s_{13} & 0 & 0 & 0 & s_{35} \\ D_4 & s_{14} & s_{24} & 0 & 0 & 0 \\ D_5 & 0 & s_{25} & s_{35} & 0 & 0 \end{array}$$

Figure 2: The adjacency matrices for the various edge types. The MISIM [3] method generates the scores for the disease-disease edges. In the final heterogenous graph structure, the vertex set is retained from the DisGeNet while the edge set is a union of all the above three Adjacencies as per Eq. 1, 2, 3 in the paper.

$$\begin{aligned} \tilde{A}^{dd}_{ij} &= \phi^i A^{dd}_{ij} \\ \phi^i &= \phi \frac{\sum_j A^{gd}_{ij}}{\sum_j A^{dd}_{ij}} \end{aligned} \tag{3}$$

where, ϕ is normalized co-efficients (*authors have not elaborated on the meaning*).

4.2 DeepWalk: generating initial node features

RandomWalk - After the graph is built, each node is assigned an index. For each node, DeepWalk selects λ neighbours based on edge weights from the adjacency matrix A. These λ-length, index sequences form the input to word2vec.

Word2Vec - The index sequences act as documents for a skip-gram version of the word2vec training model. With all the nodes (gene or disease) forming the vocabulary, the index sequences act as sentences. A one-hot vector is assigned to each node (based on its index). Then, for each node, context words are predicted for a fixed window size. The loss function (*Eq. 4 in paper*) compares the prediction against the actual context words in the sequences (*obtained from RandomWalk*)

The above two steps generate the initial feature vectors for each node in the heterogenous graph. The point to note is that these features only capture the ***structural*** information of the graph.

> The GCN network by itself can not learn anything, unless there is a down- stream task. My understanding is that the decoder layer is essentially the downstream task for the GCN.

4.3 Node embeddings via Graph Convolution Network

At this point, we have a graph (heterogeneous) and initial node-features. Kipf's GCN [2] has a good application in this scenario to learn even better connectivity embeddings for the nodes. Through message passing via the *symmetric degree normalized* adjacency matrix, each node eventually learns about all other nodes in the graph (over multiple iterations, all hop information is spread). All that is needed is a downstream task.

$$embeddings = \sigma(\tilde{D}^{\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta) \tag{4}$$

4.4 Decoding novel gene-disease pairs

For the downstream task, the authors concatenate embeddings of a gene and a disease and let a 3-layer FFNN predict the probabilities for a valid association or an invalid association (essentially a binary classification). For this, the set of positive connections and negative connections are drawn from the graph G. The loss function is a combination of **negative log-likelihood** and a novel **cluster loss** (*which looks at approximate clusters based on cosine similarity of the embedding vectors*). The clusters in question are approximate in the sense that a node is assigned to a cluster based on a *cluster centre* calculated within a distance threshold.

Gene Prioritization - It essentially means assigning ranks to genes based on weighted associations to a particular disease. The first step is to list all genes with **valid** associations (from section 4.4) for some disease. Next, filter the list based on a threshold on the association probability score. The final step is to sort the filtered gene list according to their DeepWalk probability score. The result is the final output (*Table III in the paper is an example*).

5 Positive Points

- One of the most straight-forward application of Graph Neural Networks for the task of gene prioritization
- For the most part, the paper is well described

6 Negative Points

- The construction of the combined adjacency matrix (Eq. 1 in the paper) follows some logic, however, it is not revealed in the paper
- The candidate gene list is selected from the decoding module, but, the sorting is done based on the DeepWalk scores. Authors do not provide a justification.
- The authors have mentioned that they used **dropout on Adjacency matrix** as a means to overcome over-fitting of the GCN. This approach doesn't seem right as random edge dropping breaks the domain context.

7 Questions

1. The cluster loss uses a threshold; apparently it is based on distance but, how is it selected?
2. How is the threshold during gene prioritization selected?

References

[1] L. Zhu, Z. Hong, and H. Zheng, "Predicting gene-disease associations via graph embedding and graph convolutional networks," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 382–389, 2019.

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[3] 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, pp. 1644–1650, 05 2010.