The Thirty-Third International
FLAIRS Conference (FLAIRS-33)

COPD Disease Classification Using Network Embedding with Synthetic Relationships

Anak Wannaphaschaiyong, Xingquan Zhu
Dept. of Computer and Electrical Engineering and Computer Science,
Florida Atlantic University, Boca Raton, USA
{awannaphasch2016, xzhu3}@fau.edu;

Abstract

Chronic obstructive pulmonary disease (COPD), a progressive and non-reversible lung disease causing obstructed airflow from the lungs, often occurs with other diseases not restricted to the respiratory system. Therefore, it is important to understand interaction between genes and diseases to uncover the real causes of a disease. In this paper, we propose to automatically classify COPD diseases, using network of gene disease relationships. We simplify interaction between COPD, COPD multimorbidities, and related genes as a bipartite network, and apply network embedding together with machine learning classifiers to classify diseases into different categories. Our experiments confirm that adding synthetic edges in a strategic way statistically enhances quality of node embedding and improve COPD disease classification performance.

Introduction

COPD is a progressive and non-reversible lung disease. It dramatically decreases patient's quality of life. Over 16 million people in the US have been diagnosed with COPD. In many occasions, COPD is a direct cause of COPD multimorbidities, including systemic venous multimorbidities and anxiety among others (Divo et al. 2015; Grosdidier et al. 2014). COPD multimorbidities are closely related to a variety of diseases classes, such as mental health disease, and cardiovascular disease among other.

Recently, machine learning tools are used to investigate interactions between biological units such as gene and disease interactions. Failure of these interactions cause malfunctions within biological systems which cause many diseases. Genes and diseases have complicated interaction relationships, yet the current knowledge are incapable of identifying all existed relationships between biological entities due to uncertainty or unobserved information. Therefore, instead of purely relying on given edge-relationships, we propose to add synthetic edges between disease nodes, and systematically investigate different ways to form synthetic edges and predict COPD into different disease families (*i.e.* categories or classes).

By employing a network representation tools, such as Node2vec (Grover and Leskovec 2016), each node were represented as a lower dimensional vector to be learn by machine learning model. Furthermore, we propose strategically adding synthetic edges between disease nodes to improve COPD classification performance.

Definition & Problem Statement

Motivation

Graph represents an intuitive way to connect abstract concepts together which are represented as nodes and their relation as edges. In this paper, we use bipartite gene-disease graph for COPD disease classification. In our research, we assume that edges represent partially observed node relationship. By adapting structure of graph in appropriate ways, it will provide useful information for disease classification. As we will show in our experiments, changing a graph's structure by applying appropriate rules will certainly enhance the quality of network node classification.

Definition

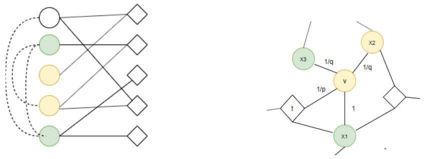
A graph is defined as $G = (V, E)$ where V represents nodes and E represents edges. In this paper, we represent genes and diseases as nodes, where V_d and V_g denote disease nodes and gene nodes, respectively. An edge connects a pair of nodes. A graph is considered bipartite if the vertex set V can be partitioned into two disjoint subsets V_g and V_d such that no edge in E has both endpoints in the same set, *i.e.* $V = V_g \cup V_d$; $V_g \cap V_d = \emptyset$; $E \in V_g \times V_d$. An illustration of COPD gene-disease bipartite network is shown on the left panel of Figure 1a.

Problem Statement

In this paper, we formulate COPD disease classification as a network node classification problem. Given a COPD gene-disease network with some labeled diseases nodes, our **goal** is to correctly predict class labels of unlabeled disease nodes.

Proposed Approach

Our paper demonstrates that adding synthetic edges between disease nodes improve the COPD disease classification results. Firstly, synthetic edges were added to the graph according to selected strategy. Then, we apply Node2vec to



(a) COPD bipartite network (b) Node2vec random walk

Figure 1: Left panel: A conceptual view of COPD gene-disease bipartite network. Circles and diamonds represent disease and gene nodes, respectively. Colors represent node labels, so no color means no label. Solid line denotes observed connections between nodes, and dashed lines denoted synthetic edges. Right panel: A conceptual view of Node2vec random walk (from node t to node v) on the COPD gene-disease bipartite network. p controls degree of exploitation and q controls degree of exploration.

convert each node as a vector and fed it to classifier to predict diseases' classes. The detailed procedures of the proposed framework described in Algorithm 1.

Network Embedding Learning: Node2vec

Node2vec is a node embedding algortihm generating informative numerical representations for nodes in the network to preserve network structure (Zhang et al. 2020), such that similar nodes in network are close to each other in the vector space.

Node2vec (Grover and Leskovec 2016) uses random walks on the network. As shown on the right panel of Figure 1b, node2vec utilized two parameters, p and q, for bias random walk. p defines probability of exploitation and exploration as probability of returning to previous nodes and probability that q will explore unvisited nodes, respectively. Using explore and exploit dilemma, Node2vec could recognize larger diversity of connectivity patterns.

Formally, given a source node u and a fixed length, we simulate a random walk started from node u to its neighbor connected by an edge. Let c_i denote the i^th node in the walk starting with c_0. Nodes c_i are generated as follows:

$$P(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if (v,x)} \in \text{E} \\ 0 & \text{otherwise} \end{cases}$$

Where π_{vx} is the unnormalized transition probability between nodes v and x, and Z is a normalization constant.

Next step is applying skip gram and negative sampling to sample paths with objective to maximize log-probability of observing nodes neighbour *i.e.* maximizing log-probability of observing N_s(u) for $u \in V$ where V are a set of all vertices and $N(u)$ are neighbors of u. Let f: V $\rightarrow \mathbb{R}^d$ be a mapping from node to its feature representation. Skip-gram aims to learn mapping function f through the following objective function:

$$\max_f \sum_{u \in V} [-logZ_u + \sum_{n_i \in N_s(u)} f(n_i) \cdot f(u)] \qquad (1)$$

where $Z_u = \sum_{v \in V} \exp(f(u) \cdot f(v))$

Node Similarity Assessment

Jaccard coefficient calculates the intersection of neighborhoods between v_i and v_j, divided by the union of the neighborhood of both nodes.

$$J(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|}$$

where $N(v_i)$ denotes neighbours of node v_i.

Nodes sharing more neighbors have a higher Jaccard similarity coefficient, and if a node has a large number of neighbors, its Jaccard similarity to other nodes will be decreased.

Disease-disease Synthetic Edges

We use Jaccard coefficient as similarity metrics between each node pairs. In this paper, we design eight synthetic edge strategies as follows:

- **Bottom Alpha Deterministic (BAD):** Jaccard coefficients scores is calculated. Then, all pairs are sorted in an ascending order of their Jaccard scores. The highest alpha (α) percent of disease node pairs are picked and a synthetic edge are formed between disease pairs.
- **Bottom Alpha Random (BAR):** Similar to BAD, except synthetic edges are randomly formed depending on probability inversely proportional to the coefficient score.
- **Top Alpha Deterministic (TAD):** Similar to the BAD, but disease pairs are sorted in a descending order.
- **Top Alpha Random (TAR):** Similar to the BAR, but the probability value is proportional to the coefficient scores.
- **Top Bottom Alpha Deterministic (TBD):** This approach combines BAD and TAD. Jaccard coefficient scores are sorted in an ascending order. A $\frac{\alpha}{2}$ of synthetic edges are selected from the top and bottom of the list, respectively.
- **Top Bottom Alpha Random (TBR):** This approach selects $\frac{\alpha}{2}$ of edges using BAR and TAR, respectively.
- **All Nodes Random (ANR):** Collecting all disease pairs. Uniform distribution is assigned to the pairs. Randomly select edges from uniform distribution. Same amount of edges are added for the same value of alpha used in other strategies.
- **Shared Nodes Random (SNR):** Similar to ANR, but only account for disease pairs sharing at least a gene.

Ensemble Prediction

After we adding synthetic edges to the network, node2vec is applied to generate embedding features. To reduce stochastic effect from random walk, we repeat the process multiple times (controlled by parameter N in the Algorithm 1). After that, majority voting ensemble, $\boldsymbol{\mathcal{E}}(\boldsymbol{\mathcal{M}})$, is applied to predict unlabeled COPD disease nodes.

Algorithm 1 Algorithm Procedures

```
Require:
    (1) $\mathcal{D}$: Gene-Disease Relationship Dataset
    (2) $\mathcal{S}$: Edge Addition Strategy
    (3) $\alpha$: Alpha ($\alpha$) for edge addition
    (4) $\mathcal{C}$: Classifier
    (5) $\mathcal{N}$: Ensemble Size
    (6) $\mathcal{E}$: Voting Ensemble
Ensure:
    $\mathcal{E}(\mathcal{M})$: Voting ensemble's prediction
$\mathcal{G} \leftarrow$ Create COPD gene-disease network from gene-disease relation dataset $\mathcal{D}$
$traning\_set$,$test\_set$ = split($\mathcal{G}$). Split network into training set vs. test set.
$\mathcal{M} \leftarrow \emptyset$. A list of models' predictions
for $i$=1 to N do
    $\hat{\mathcal{G}} \leftarrow$ AddSyntheticEdges($\mathcal{G}$, $S$, $\alpha$ ). Add edges between disease nodes using edge addition strategy $\mathcal{S}$
    $node\_emb \leftarrow$ NODE2VEC($\hat{\mathcal{G}}$). Apply Node2vec to generate node embedding
    $\mathcal{C}$.train($training\_set$,node_emb). Train classifier
    $pred \leftarrow \mathcal{C}$.predict($test\_set$,node_emb). A classifier predicts diseases labels.
    $\mathcal{M}$.append($pred$)
end for
return $\mathcal{E}(\mathcal{M})$
```

Descriptions	Values
Number of edges	4,715
Number of nodes	2,975
Number of COPD disease nodes	101
Number of gene nodes	2,874
# of COPD disease classes (only disease is labeled)	5
Network diameter	8
Diameter between disease nodes	8

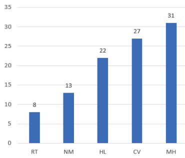


Figure 2: Left panel: A summary of the COPD gene-disease network. Right panel: class distributions of COPD disease labels. From left to right Respiratory Tract disease (RT), Nutritional and Metabolic Disease (NM), Hemic and Lymphatic Disease (HL), Cardiovascular Diseases (CV), and Mental Health disease (MH).

Experiments

COPD Gene-Disease Network

Gene-disease edges of COPD and its multimorbidities are extracted from DisGeNET. There are 4,715 edges and 2,975 gene nodes, including 101 COPD disease nodes and 2,874 gene nodes. Properties of COPD gene-disease network is summarized in figure 2 (right). Our dataset can be downloaded from [1]. Properties of our network is shown in figure 2 (left panel). In our experiments, only disease nodes are labelled, and all gene nodes are unlabelled.

We use Disease Ontology to label disease nodes. Disease ontology is constructed as a tree like structure without crossover between branches. It consists of seven distinct branches including: disease by infectious agent, disease of anatomical entity, disease of cellular proliferation, disease of mental health, disease of metabolism, genetic disease, physical disorder, and syndrome. In disease ontology, each disease node has exactly one parent node. To obtain COPD disease labels, we start from each of the 101 COPD diseases and traverse to their parents, grandparents, and so on. The walk is repeated, until there are five categories (classes). The COPD disease network and class distributions are reported in Figure 2.

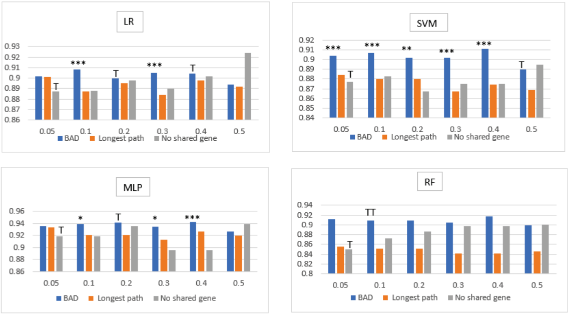


Figure 3: Comparing BAD to LPA and NSG strategy. Confidence of permutation test are represented as superscript. T is 50-80 percent confidence, * is 80 to 90 percent confidence, and ** is 90 to 98 percent confidence, and *** is more than 98 percent confidence

Experiment Settings

In Algorithm 1, α denotes the percentage of existing edges to be added as synthetic edges. We use six α values: 0.05, 0.1, 0.2, 0.3, 0.4, and 0.5 and four classifiers: Random Forest (RF), Linear Regression (LR), Multilayer Perceptron (MLP), and Support Vector Machines (SVM). 60-40 split is used where 60 % is training set and 40 % is test set. Node2vec compress features to 64 dimensions. Size of voting ensemble is set to 10.

Baseline Method and Performance Metrics We report performance of the eight synthetic edge approaches (BAD, BAR, TAD, TAR, TBD, TBR, ANR, and SNR), and compare them with the baseline, graph with no adding synthetic edges. The tables show AUC values using five classifiers with different synthetic edge approaches.

Statistical Test We use permutation test (10,000 runs) to test statistically significant performance of differences. Permutation test is applied between models and their benchmarks. For example, TAD + Random Forest will be compared against original set of edges with Random Forest. These comparisons measure effectiveness of edge adding strategies.

Results & Analysis

Impact of Synthetic Edges Table 1 reports AUC values for COPD disease classification. MLP always outperforms except four times where its performance seconds to LR. BAD is shown to be the best strategy considering number of outperforms models. According to table 1, Excluding 0.5, BAD yields better performance 16 out of 20 times. Adding synthetic edges to RF causes performance deterioration both at individual and ensemble case. LR outperforms 34 out of 48 times. Excluding BAR and TAD, LR outperforms 31 out

[1] http://eng.fau.edu/research/kmelin/resources/COPD.Network.html

Table 1: Bold indicates better performance than the benchmark, underline indicates worse performance and italic indicates a model performance within 0.5 percent of the benchmark. Superscript characters denote models with an improved performance. Type of superscript are determined by confidence of permutation test. *** indicates confidence more than 98 confidence. ** indicates 90-95 percent confidence. * indicates 80-90 percent confidence. $^{\mathrm{T}}$ indicate 50-80 percent confidence.$^{\mathrm{TT}}$ indicate confidence less than 80.

		LR	SVM	MLP	RF
No Added Edges		0.900	0.896	0.929	0.910
Bottom Alpha Deterministic (BAD)	0.05	**0.902**	**0.904*****	**0.936**	**0.911**
	0.1	**0.908*****	**0.907*****	**0.939***	*0.909*$^{\mathrm{TT}}$
	0.2	*0.900*$^{\mathrm{T}}$	**0.902****	**0.942**$^{\mathrm{T}}$	*0.909*
	0.3	**0.905****	**0.902*****	**0.935***	0.904
	0.4	**0.904**$^{\mathrm{T}}$	**0.911*****	**0.943****	**0.917**
	0.5	0.894	0.890$^{\mathrm{T}}$	0.927	0.899
Bottom Alpha Random(BAR)	0.05	**0.908**	**0.907**$^{\mathrm{T}}$	*0.929**	0.900$^{\mathrm{TT}}$
	0.1	0.893	0.903	*0.928*	0.895
	0.2	0.883	0.884$^{\mathrm{T}}$	0.915	0.891
	0.3	0.891	0.885	**0.931**	0.891
	0.4	0.870	0.868	0.901	0.879
	0.5	**0.899**	0.883	0.914	0.889
Top Alpha Deterministic (TAD)	0.05	0.880$^{\mathrm{T}}$	0.885	0.906$^{\mathrm{TT}}$	0.893
	0.1	**0.902**$^{\mathrm{T}}$	0.893$^{\mathrm{TT}}$	*0.928*	**0.916**
	0.2	0.880	0.879	**0.922**	0.892
	0.3	0.884	0.872	0.889	0.888
	0.4	0.896	0.884$^{\mathrm{T}}$	**0.919**	0.905
	0.5	0.893$^{\mathrm{TT}}$	0.891*	**0.917**$^{\mathrm{TT}}$	0.888
Top Alpha Random (TAR)	0.05	**0.903**$^{\mathrm{TT}}$	**0.905*****	**0.937**$^{\mathrm{T}}$	0.908$^{\mathrm{TT}}$
	0.1	**0.909**$^{\mathrm{T}}$	**0.897*****	**0.934*****	0.898
	0.2	0.897	0.880*	0.918	0.886
	0.3	**0.902**$^{\mathrm{TT}}$	0.877	*0.928*$^{\mathrm{TT}}$	0.893
	0.4	**0.901**	0.886$^{\mathrm{TT}}$	0.923	0.887
	0.5	**0.903**$^{\mathrm{T}}$	*0.896*	**0.938**$^{\mathrm{TT}}$	0.905
Top Bottom Alpha Deterministic (TBD)	0.05	**0.910*****	**0.901**	**0.935*****	*0.909*
	0.1	**0.910**$^{\mathrm{TT}}$	0.887	**0.933**$^{\mathrm{TT}}$	0.883
	0.2	**0.913**$^{\mathrm{TT}}$	0.891$^{\mathrm{TT}}$	**0.939****	0.899
	0.3	**0.909**$^{\mathrm{T}}$	0.883	**0.941***	0.894
	0.4	**0.904**	0.879	0.925	0.882
	0.5	0.891	0.888	**0.928**	0.902
Top Bottom Alpha Random (TBR)	0.05	**0.903*****	**0.901**	**0.948*****	0.896
	0.1	**0.917***	**0.910**$^{\mathrm{TT}}$	**0.933***	**0.919**$^{\mathrm{TT}}$
	0.2	**0.906**$^{\mathrm{TT}}$	**0.906**$^{\mathrm{TT}}$	0.923$^{\mathrm{TT}}$	0.897
	0.3	**0.909**$^{\mathrm{T}}$	0.889	**0.933*****	0.897
	0.4	**0.921**	0.858	0.901	0.872
	0.5	**0.919**	0.864$^{\mathrm{TT}}$	0.903	0.867
All Nodes Random (ANR)	0.05	**0.930**	0.887$^{\mathrm{T}}$	**0.943**$^{\mathrm{T}}$	0.890
	0.1	**0.904****	**0.904**	0.918*	0.906
	0.2	**0.916*****	**0.901**$^{\mathrm{T}}$	**0.934**	0.896
	0.3	**0.903***	0.893	**0.931**$^{\mathrm{TT}}$	0.897
	0.4	**0.915**	0.889	**0.943**	0.880
	0.5	0.886	0.881	0.919	0.880
Shared Nodes Random (SNR)	0.05	**0.920**	0.885*	0.914	0.893
	0.1	**0.920***	*0.889**	**0.931***	0.898
	0.2	**0.912**$^{\mathrm{TT}}$	0.886$^{\mathrm{TT}}$	*0.929*$^{\mathrm{TT}}$	0.907
	0.3	**0.919**	0.877$^{\mathrm{TT}}$	0.926	0.884
	0.4	**0.906**	0.879	0.925	0.885
	0.5	**0.906**	0.871	0.896	0.875

of 36 time. Adding synthetic edges, RF performance worsen 44 out of 48 times against benchmark.Furthermore, using BAD, with LR, SVM, and MLP, 14 out of 18 models yield better performances.

BAD and TBR improve performance with over 90 percent confidence for α value greater than 0.2. With α greater than 0.2, only eight out of 40 models outperform with more than 80 percent confidence, and five of them are from BAD. This demonstrates effectiveness of BAD strategies and demonstrates that edges that are added by alpha more than 0.2 are mostly noise. Furthermore, with threshold less than 0.3, with BAD, TBD, TBR, ANR, and SNR and exclude RF classifier, outperforms 34 out of 45 times and 22 time outperforms with more than 80 percent.

Comparing permutation test score between BAD and BAR, random process reduces its consistency to outperform from 16 times to four times. In contrast, comparing TAD to TAR, random process improves its consistency to outperforms from five times to 10 times. The results imply that, synthetic edges with low Jaccard coefficient improves disease classification prediction.

Rationale of BAD We compare BAD to other two approaches: Longest Path Alpha (LPA) and No Shared Gene Alpha (NSG). Number of added edges are always the same for each alpha.

- **Longest Path Alpha (LPA):** For all disease pairs, LPA assigns higher probability to nodes that are further apart. Paths with the same distance are assigned uniform distribution.
- **No Shared Gene Alpha NSG:** Collecting diseases with no common genes. Uniform distribution are assigned to all nodes.

We design LPA to test whether there is useful information between nodes that are separated by longest shortest path. One could think of the synthetic edges as a concept of disease pairs partially caused by the same malfunction gene. Therefore, it is natural to test the hypothesis against NSG. In Figure 3, none of the models utilizing NSG out-performed benchmark. The result confirms that the rationale of BAD and are responsible by adapting graph structure to shorten shortest path connecting nodes with shared genes.

Conclusion

In this paper, we proposed to use bipartite COPD multimorbidities gene-disease network for disease classification. We implemented eight strategies and compared their performance. Our experiments confirmed that connecting disease node using Bottom Alpha Deterministic (BAD) result in the best performance. Among all classifiers, neural network (MLP) performs the best, and synthetic edge addition is more beneficial to Linear Regression (LR) classifier.

Acknowledgment

This work is supported by the U.S. National Science Foundation under grant Nos. IIS-1763452 & CNS-1828181.

References

Divo, M.; Casanova, C.; Marin, J.; Pinto-Plata, V.; de Torres, J.; Zulueta, J.; Cabrera, C.; et al. 2015. Copd comorbidities network. *European Respiratory J.* 46(3):640–650.

Grosdidier, S.; Ferrer, A.; Faner, R.; Piñero, J.; Roca, J.; Cosío, B.; Agustí, A.; Gea, J.; Sanz, F.; and Furlong, L. I. 2014. Network medicine analysis of copd multimorbidities. *Respiratory research* 15(1):111.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proc. of the 22nd ACM SIGKDD Conf.*, 855–864. ACM.

Zhang, D.; Yin, J.; Zhu, X.; and Zhang, C. 2020. Network representation learning: A survey. *IEEE Trans. on Big Data* 6(1).