Report: Disease Prediction Using Graph-based models

Minh Nguyen Dich Nhat, Dat Pham Quoc, Anh Hoang The

January 4, 2024

1 Problem Definition

EMRs, short for electronic medical records, are a commonly utilized data management scheme employed in hospitals to store comprehensive clinical information obtained from patients' visits. In recent times, advancements in information technology and machine learning have led to a more manageable volume of EMRs. Consequently, the analysis of EMRs using machine learning and data mining techniques has emerged as a prominent research direction, aiming to enhance healthcare services. One notable application of machine learning in the healthcare domain involves disease prediction, which focuses on determining whether a patient is afflicted with a specific ailment. This task typically involves training a classifier to make predictions based on the information derived from EMRs

This report show our experiments about graph-based models for disease prediction and compare it with traditional approaches

2 Survey

There have been a lot of approaches have been proposed for this task. For instance, Palaniappan and Awang employed various data mining techniques, such as Decision Trees [14], and Neural Networks [8], to construct a predictive system for heart diseases [12]. Leveraging the capabilities of Convolutional Neural Networks (CNNs), Suo et al. [17] initially identified similarities among patients based on their EMRs and subsequently conducted personalized disease predictions.

Existing deep neural networks lack the incorporation of patient interactions and associations within their architecture. However, considering these relationships can prove advantageous as it facilitates the analysis and study of similar patient cohorts. Graphs offer a natural means of representing the interactions among a population by treating patients as nodes and their associations as edges. Constructing a graph between patients based on a subset of their features allows for the summarization of these features within the graph edges, reducing feature dimensionality and mitigating overfitting caused by a large number of features [2], [18]. In recent years, GCNs have been adopted in different applications, especially in medical domains e.g., brain analysis [9], mammogram analysis [4],... The disease prediction problem has been also widely explored by GCN-based methods [1], [10]

3 Purpose

In this report, I will build various graph-based models, including both single-graph and multi-graph inputs. Additionally, I will utilize different mechanisms to update and aggregate messages between nodes. I will combine GCN layers, Multi Layer Perceptron (MLP), and mechanisms for information exchange and update, such as using Gated Recurrent Neural Network (GRU) or Attention mechanism, to build a disease prediction model. I also propose different ways to build a graph and compare to traditional approaches in this problem

4 Definitions

4.1 Graph Classifier

We consider a graph G with N nodes, denoted by $G(V, E, X)$, where V represents the set of nodes ($|V| = N$), E is the set of edges, and $X \in R^{N \times F}$ denotes the node feature matrix. The adjacency matrix $A \in R^{N \times N}$ is unweighted and undirected, reflecting the connections between nodes. Each node v_i has an associated feature vector x_i , a one-hot label vector y_i , and a true class label c_i C, where C represents the set of classes. The label information is available only for a subset of nodes, and the objective is to learn a parametric function $f_\theta(X, A)$ that takes the adjacency matrix and node features as input and predicts the true labels for the unlabeled nodes. It is important to note that a probabilistic classifier generates a probability distribution over the $|C|$ classes, and the label with the highest probability is selected. In our proposed method, q_i represents the output probability distribution of the classifier, defined

over the $|C|$ classes, for the sample x_i where the c-th element represents the classifier's confidence in assigning the label c to x_i. Thus, the problem can be formulated as follows:

$$Q = f_\theta(X, A) \tag{1}$$

where $Q \in R^{|N| \times C}$ is the prediction matrix of the classifier for all nodes, including the unlabeled ones

4.2 Neural Graph Encoder

To start, in the context of a graph G we adopt a uniform representation for disease, symptom, or patient nodes as v E for brevity. At the l-th layer of information propagation, the embedding $\mathbf{h}_v^l$ of node v is computed as follows:

$$\mathbf{h}_{N(v)}^l = AGGREGATE(\mathbf{h}_{v'}^{l-1}, \forall v' \in N(v)) \tag{2}$$

$$\mathbf{h}_v^l = \sigma(\mathbf{W}_l.[\mathbf{h}_v^{l-1}; \mathbf{h}_{N(v)}^l]) \tag{3}$$

Here, W^l denotes the weight matrix to be learned at the l-th layer, $\mathbf{h}_v^{l-1}$ represents the embedding of node v at the previous layer, and L denotes the total number of layers. The concatenation of two vectors is denoted by $[\cdot; \cdot]$, and $N(v)$ represents the set of neighbor nodes of v sampled uniformly. It is important to note that for $l = 0$, the node embedding $\mathbf{h}_v^0 \in R^d$ is initialized using either random values or side information from the data, depending on its availability. For example, in the case of a patient node, if patient demographics and medical profiles are available in the electronic medical record (EMR) data, $\mathbf{h}_v^0$ will be initialized as a dense, real-valued feature vector. Each element in $\mathbf{h}_v^0$ would then represent an observed value of a specific feature dimension (e.g., age). $\mathbf{h}N(v)$ represents the synergistic representation obtained through the aggregation function, which is designed to combine the embeddings of node v's neighbors from the ($l1$)-th layer. σ denotes a non-linear activation function such as the hyperbolic tangent (tanh), and the aggregator can be selected from options like mean, max pooling, recurrent neural networks (RNNs), and so on.

Then, we take a normalization step before reaching the final embedding for all nodes at the last layer L :

$$\mathbf{h}_v = \frac{\mathbf{h}_v^L}{||\mathbf{h}_v^L||_2}, v \in G \tag{4}$$

After passing through the GCN layers, we will feed $\mathbf{h}_v$ through an MLP layer to obtain the final embedding representation.

$$\mathbf{z}_v = \sigma(\mathbf{W}\mathbf{h}_v), \forall v \in G \tag{5}$$

where $\mathbf{W}$ is the learnable weight, and $\mathbf{z}_v$ is the final embedding for node v.

4.3 Attention mechanism

We can apply attention mechanisms to selectively encode the information from neighbors according to their importance to the target node v. This is achieved by taking a weighted sum of the representations of all v 's neighbor nodes:

$$\mathbf{h}_v^l = \sum_{v' \in N(v)} \alpha_{v'v} \mathbf{M}\mathbf{h}_{v'}^{l-1} \tag{6}$$

where M is the transformation weight matrix, and $\alpha_{v'v}$ is the attentive weights indicating the importance of neighbor node $v' \in N(v)$ when calculating h_v^l. Each $\alpha_{v'v}$ is computed via the following attention network:

$$\alpha_{v'v} = \frac{exp(LeakyReLU(a^T[\mathbf{N}\mathbf{h}_v^{l-1}||\mathbf{N}\mathbf{h}_{v'}^{l-1}])}{\sum_{k \in N(v)} exp(LeakyReLU(a^T[\mathbf{N}\mathbf{h}_v^{l-1}||\mathbf{N}\mathbf{h}_k^{l-1}])} \tag{7}$$

with a projection vector a and the weight matrix N. Essentially, the learned attentive weights allows the aggregator to lay more emphasis on neighbor nodes having more contributions to the message passing process, thus being able to generate highly expressive node embeddings.

4.4 Disease Prediction for Patients

The graph decoder in our model plays a crucial role in translating the information captured by symptom, disease, and patient node embeddings into predictions of potential diseases associated with a given patient. To elaborate, when provided with the embedding z_p of a patient p, the graph decoder transforms it into a vectorized output $\hat{c}_p \in 0, 1^{|C|}$ that serves as an approximation of the patient's multi-hot disease label $c_p \in 0, 1^{|C|}$. Specifically, during the decoding process, each element of $\hat{c}_p$ is computed as follows:

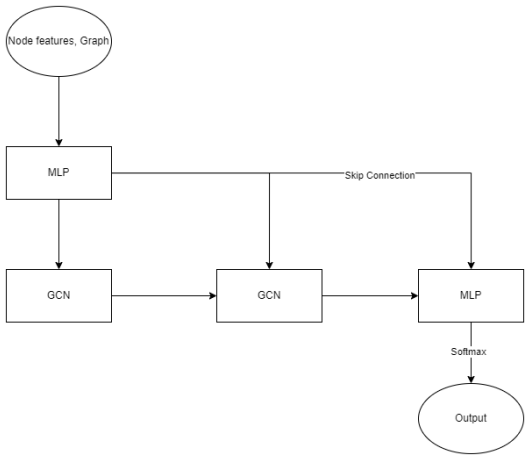
$$\hat{c}_{p,n} = sigmoid(z_p^T Q_n), \forall m_n \in \mathbf{V}_M \tag{8}$$

where $\hat{c}_{p,n}$ is the n-th element in $\hat{c}_p$, while n $|V_M|$ is used for indexing all the diseases m. The closer $\hat{c}_{p,n} \in \hat{c}_p$ is to 1 , the more likely patient p is diagnosed with disease m_n. $Q \in R^{V_M \times d}$ carries the corresponding regression weights for all diseases, and Q_n is the n-th column of it. To train our model, we quantify the prediction error via the following negative log likelihood loss function:

$$\mathcal{L} = -\sum_{n=1}^{|V_M|} c_{p,n} \log(\hat{c}_{p,n}) \tag{9}$$

5 Architectures

I will build two different types of Graph-based models, using either a single graph or multiple graphs as inputs. To avoid the over-smoothing problem, I will only use two layers of Graph Convolution Network (GCN) and incorporate skip connections to preserve information. To enhance the model's representational capacity, I will combine MLP layers, including Fully Connected and Batch Normalization layers, to extract information effectively. Moreover, during the training process, I will employ various mechanisms to aggregate information, such as SUM, MAX, MEAN, and Attention, to effectively combine the information from neighboring nodes. For updating node representation, I will utilize MLP layers to learn isomorphic or non-isomorphic sub-graphs and employ a GRU network to learn sequential representations of neighboring node



Hình 1: Figure 1: One Input Graph

6 Experiment

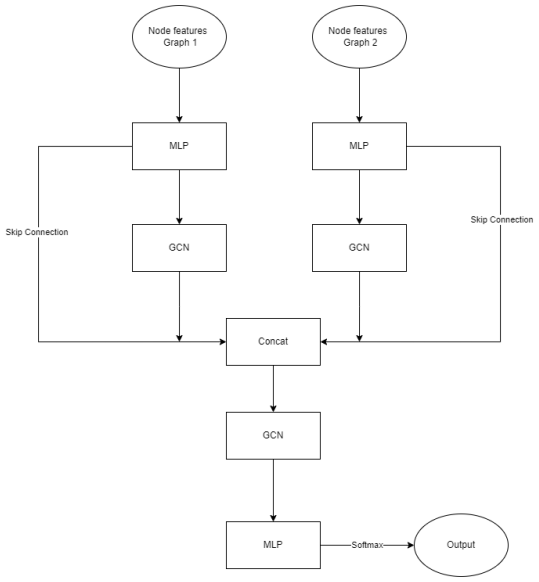
6.1 Environment

We will using Google Colab with Python programming language as our experimental environment.

- GPU: T4
- RAM: 16GB

6.2 Data

I utilize two datasets for disease prediction based on relevant symptoms to construct multi-class classification and binary classification problems.



Hình 2: Two Input Graphs

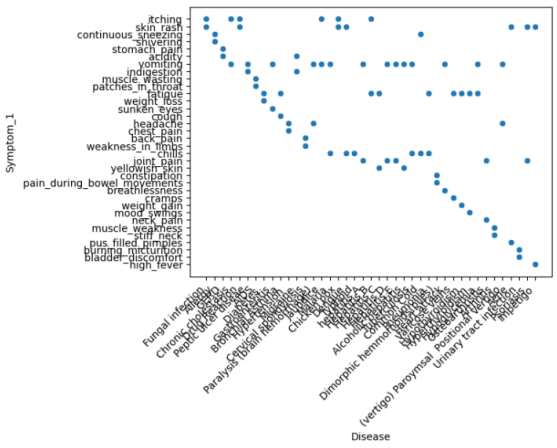
6.2.1 Disease Symptom Prediction

This dataset consists of 4920 records with 18 different fields representing disease symptoms. There are 41 distinct diseases, with each disease having 120 records. After conducting the Exploratory Data Analysis (EDA), I observed that each disease exhibits 2-3 distinctive symptoms. There are also common symptoms shared among different diseases, such as headaches and fatigue. Additionally, patients infected with the same disease tend to have 1-2 similar symptoms. These findings inspire the idea of constructing graphs for different disease types and patient graphs based on shared symptoms. These graphs serve to generalize the information of the patients, and by utilizing Graph Neural Networks (GNN), we can extract latent features from the data.

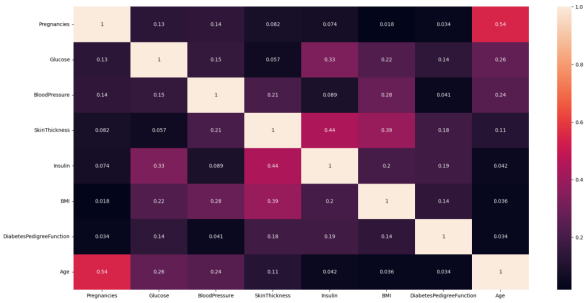
6.2.2 Pima Indian Diabetes (Diabetes)

The dataset is produced by the "National Institute of Diabetes and Digestive and Kidney Diseases". The goal of this dataset is to recognize the diabetic status of patients (binary classification). Every patient has 7 numeric features which show the diagnostic measurements of diabetes including the number of pregnancies, plasma glucose, blood pressure, skin thickness, Body Mass Index (BMI), insulin level, diabetes pedigree function, and age.

Based on the data, we can observe a relatively high correlation between the age field and other fields in the dataset. Therefore, I perform binning on the age field to create different age groups and



Hình 3: Relation between Symptom and Disease



Hình 4: Correlation between feature columns

utilize them to construct graph connections between these age groups.

6.3 Graph Construction

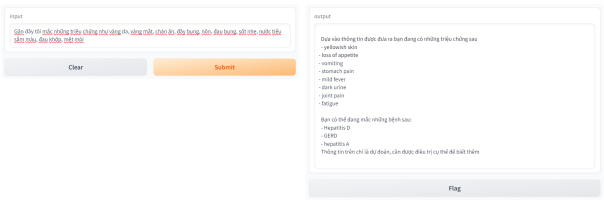
I proposed some strategy to construct 2 type of graph: Disease Graph, Patient Graph Graph. For establishing connections between two patients, we can:

- Strategy 1: Create an edge between two patients if they have the same type of disease (To avoid data leakage, only use the training set to construct the graph).
- Strategy 2: Create an edge between two patients if they share two similar symptoms. (This approach can be combined with Strategy 1 to create a new graph).
- Strategy 3: Create an edge between two patients if their cosine similarity, measured by a threshold γ exceeds a certain threshold. $a_{ij} = 1$ if cosine-similarity$(\mathrm{x}_i, x_j) > \gamma$ and $= 0$ otherwise where a_{ij} is indicator for representing the existence of an edge connecting patients i-th and j-th and x_i is sample of patient i-th.

For establishing connections between two diseases, we can create a connection between two diseases if they share the top two prominent symptom

7 Demo

For the demo, I will create a Chatbot that take the question from patient and predict the disease based on the information provided. I use API from OpenAI to get access to ChatGPT and use it to extract symptoms from the input provided by user, and use them to predict the disease using our model. I also use Gradio library to build an interface for users



Hình 5: Demo Interface

8 Result

In this report, to evaluate the classification performance of the model, I optimize the parameters based on the F1-macro, Recallmacro, and

Precision-macro metrics. This approach is chosen because in reality, there are numerous rare iseases with very few data records. Optimizing parameters using the macro strategy helps to address the issue of data imbalance to some extent

Binary Classification: We proceeded to build a binary classification model using the Diabetes dataset, I performed age binning to form age groups and established connections between patient age group using Strategy 3

The table below show results between different combination

Model	Accuracy
Attention + GRU	0.76
Mean + GRU	0.76
Sum + GRU	0.77
Attention + Concat	0.81
Mean + Concat	0.75
Sum + Concat	0.75

Compare our model with traditional approaches:

Model	Accuracy
Decision Tree	0.69
XG Boost	0.75
Graph	0.81

Multi-class Classification: After conducting experiments on the Kaggle dataset, we obtained the following results for each graph construction strategy and methods of aggregating information from neighboring nodes and updating node representation

Model	Accuracy	F1
Random Forest	0.91	0.90
One Graph Input	0.98	0.98
Multi Graph Input	1	1

We can observe that the Graph-based models all outperform the baseline model, Random Forest, demonstrating their effectiveness on this dataset. Furthermore, the training method that reduces the size of the input graph and utilizes the Multi Head Attention layer helps the model focus on important connections within the graph, resulting in excellent classification result

9 References

[1]Rushil Anirudh and Jayaraman J Thiagarajan. Bootstrapping graph convolutional neural networks for autism spectrum disorder classification. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3197–3201. IEEE, 2019.

[2] Christopher M Bishop and Nasser M Nasrabadi. Pattern recognition and machine learning, volume 4. Springer, 2006.

[3] Luca Cosmo, Anees Kazi, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein. Latent-graph learning for disease prediction. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23, pages 643–653. Springer, 2020.

[4] Hao Du, Jiashi Feng, and Mengling Feng. Zoom in to where it matters: a hierarchical graph based model for mammogram analysis. arXiv preprint arXiv:1912.07517, 2019.

[5] Richard Hillestad, James Bigelow, Anthony Bower, Federico Girosi, Robin Meili, Richard Scoville, and Roger Taylor. Can electronic medical record systems transform health care? potential health benefits, savings, and costs. Health affairs, 24(5):1103–1117, 2005.

[6] Yongxiang Huang and Albert CS Chung. Edge-variational graph convolutional networks for uncertainty-aware disease prediction. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VII 23, pages 562–572. Springer, 2020.

[7] Anees Kazi, Shayan Shekarforoush, S Arvind Krishna, Hendrik Burwinkel, Gerome Vivar, Karsten Kortüm, Seyed-Ahmad Ahmadi, Shadi Albarqouni, and Nassir Navab. Inceptiongcn: receptive field aware graph convolutional network for disease prediction. In Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2–7, 2019, Proceedings 26, pages 73–85. Springer, 2019.

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553):436–444, 2015.

[9] Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H Staib, Pamela Ventola, and James S Duncan. Braingnn: Interpretable brain graph neural network for fmri analysis. Medical Image Analysis, 74:102233, 2021.

[10] Yang Li, Buyue Qian, Xianli Zhang, and Hui Liu. Graph neural network-based diagnosis prediction. Big Data, 8(5):379–390, 2020.

[11] Fenglong Ma, Jing Gao, Qiuling Suo, Quanzeng You, Jing Zhou, and Aidong Zhang. Risk prediction on electronic health records with prior medical knowledge. In Proceedings of the

24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, pages 1910–1919, 2018.

[12] Sellappan Palaniappan and Rafiah Awang. Intelligent heart disease prediction system using data mining techniques. In 2008 IEEE/ACS international conference on computer systems and applications, pages 108–115. IEEE, 2008.

[13] Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrerro Moreno, Ben Glocker, and Daniel Rueckert. Spectral graph convolutions for population-based disease prediction. In Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III 20, pages 177–185. Springer, 2017.

[14] J. Ross Quinlan. Simplifying decision trees. International journal of man-machine studies, 27(3):221–234, 1987.

[15] Jack W Smith, James E Everhart, WC Dickson, William C Knowler, and Robert Scott Johannes. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the annual symposium on computer application in medical care, page 261. American Medical Informatics Association, 1988.

[16] Xuegang Song, Feng Zhou, Alejandro F Frangi, Jiuwen Cao, Xiaohua Xiao, Yi Lei, Tianfu Wang, and Baiying Lei. Graph convolution network with similarity awareness and adaptive calibration for disease-induced deterioration prediction. Medical Image Analysis, 69:101947, 2021.

[17] Qiuling Suo, Fenglong Ma, Ye Yuan, Mengdi Huai, Weida Zhong, Aidong Zhang, and Jing Gao. Personalized disease prediction using a cnn-based similarity learning method. In 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 811–816. IEEE, 2017.

[18] Honglei Zhang and Mancef Gabbouj. Feature dimensionality reduction with graph embedding and generalized hamming distance. In 2018 25th IEEE Inte