Graph Neural Network on Electronic Health Records for Predicting Alzheimer's Disease

Weicheng Zhu
Center for Data Science
New York University
New York City, NY 10011
jackzhu@nyu.edu

Narges Razavian
School of Medicine
New York University
New York City, NY 10016
narges.razavian@nyulangone.org

Abstract

The cause of Alzheimer's disease (**AD**) is poorly understood, so forecasting AD remains a hard task in population health. Failure of clinical trials for AD treatments indicates that AD should be intervened at the earlier, pre-symptomatic stages. Developing an explainable method for predicting AD is critical for providing better treatment targets, better clinical trial recruitment, and better clinical care for the AD patients. In this paper, we present a novel approach for disease (AD) prediction based on Electronic Health Records (**EHR**) and graph neural network. Our method improves the performance on sparse data which is common in EHR, and obtains state-of-art results in predicting AD 12 to 24 months in advance on real-world EHR data, compared to other baseline results. Our approach also provides an insight into the structural relationship among different diagnosis, Lab values, and procedures from EHR as per graph structures learned by our model. [1]

1 Introduction

Graph neural network (**GNN**) has been considered an effective way to generalize Convolutional Neural Networks (**CNN**) in extracting signals from non-grid structured data. Electronic Health Records (**EHR**) are inherently sparse and structured data with high probability of missing values. As CNNs can extract features from images with missing pixels, learning graph structures on EHR can likewise help infer the missing entries through representation of other entries, leading to a more generalizable and explainable representations. Therefore, GNN can be a strong tool on multiple machine learning tasks on EHR, including patient representation learning, medical graph learning and disease prediction.

Alzheimer's Disease (**AD**) is a cause for majority of dementia. At the moment, all of clinical trials for reversing AD have failed [1]. It is known that AD needs to be treated before the onset of clinical symptoms, and clinical trials need to intervene on high-risk patients before advanced irreversible brain degeneration.[2]. There are several approaches to predicting AD, including using brain nuclear MRI imaging (PET scans) and cerebrospinal fluid protein tests, but predicting AD based on the EHR data remains unexplored, despite the feasibility of scaling screening to large populations, without requiring any extra exams. However, there are two major challenges in using EHR data for disease prediction. First, EHR data is highly sparse and misses documentation of many disease history or lab measurements, because most patient/physician encounters only focus on, and measure a small subset of conditions and variables. Second, a relational graph structure among all variable types within EHR is difficult to construct, because it requires considerable efforts and expertise to manually construct a reasonable graph structure for over 400K nodes, and existing ontologies only capture a small subset

[1] Code for this project is available on https://github.com/NYUMedML/GNN_for_EHR

Preprint. Under review.

of relationships among isolated groups of variables. In our paper, we use Graph Attention Network [3], a variation of GNN, to tackle these two challenges in the task of predicting outcomes of AD between 12 to 24 months into the future, for patients who do not already have it, using EHR data.

2 Related Work

A number of recent works approach representation learning on Electronic Health Records (**EHR**) for disease prediction [4] through concept embedding [5, 6] and temporal information from encounter sequences [7–11]. These methods do not take advantage of the structure inherent within and between diagnosis (**ICD-10**, **SNOMED**), lab values(**LOINC**), and procedures (**CPT**) variables, which can be expressed as a graph.

One approach to signal processing with Graph Neural Network (**GNN**) is the spectral network [12] which applies Fourier transform to graph Laplacian on graph structured data. Based on spectral network model, Graph Convolutional Network (**GCN**) generalizes translational convolution filters in standard convolutional neural networks(**CCNs**) to a non-Euclidean localized filter[13], that can be applied to various non-grid data. One application of GCN is to learn representations of node features and graph structure through semi-supervised learning on node classification [14]. This work provides a paradigm for generating labels for unknown nodes given graph structures and node features.

Self-attention, comparable to CNN in encoding features from sequential data, requires less computation time and out-preforms CNN in sequence generating tasks like machine translation[15]. Similar to replacing convolutional block with self-attention, Graph Attention Network (**GAT**) [3] attends each node in the graph on its neighbouring nodes and itself to learn localized features instead of using spectral filters. In this architecture, GAT can assign different importance to edges, which increases the model capacity and interpretability as well as learns graph structures itself by attention weights.

Inspired by these recent works, we construct a EHR graph by taking each EHR feature as a node, initially imposing a fully connected structure on them, and implicitly learning their graph structure via self-attention mechanism in GAT. Some previous research also model EHR represetation with in graph structure: MiME[16] builds tree structures on visit level representations; GRAM[17] leverages external knowledge DAG. However, a well defined graph is necessary for these methods, which is a great limitation on generality. A recent work introduced Graph Convolution Transformer on EHR [18], an architecture which learns graph structure with Transformers without graph structure and regularizes learned structures via medical graph structure, if known structure exists. This work addressed the difficulty in manually constructing graphs and exploited graph properties of EHR with an average-pooled visit representation for supervised learning tasks (readmission, morality prediction) or self-supervised learning tasks (graph reconstruction, masked diagnosis categorizing) tasks.

Our paper approaches disease prediction with latent graph structure of EHR in a different way: we introduce a 2-layer GAT on EHR data, a model that predicts labels for outcome nodes based on hidden variables computed from their connections to nodes in original EHR graph and corresponding node representations. This enables our outcomes to both attend to other nodes in the graph as well as to each other, if there are more than one outcomes.

3 Methods

3.1 Graph Attention Layer

The graph attention layer [3] is the building block of arbitrary graph attention network. For each graph attention layer, we have a directed graph that is composed of nodes $V = \{1, 2, \cdots, N\}$ and edges E that connect some node i to node j. For each node i, it has a feature vector $h_i \in \mathbf{R}^d$. We would like to produce a set of new representations for each node $h' = \{h'_i\}_{i \in V}, h'_i \in \mathbf{R}^{d'}$ that incorporate graph information.

To adjust the dimensionality of features h_i from d to d' for desired output and increase the model capacity potentially, a linear layer $W \in \mathbf{R}^{d' \times d}$ is applied to all input features. Then we use attention mechanism [19] to compute the weight α_{ij} on the edge from node i to node j, where α_{ij} is expressed

by a function of attention coefficients $e_{ip} \in \mathbf{R}$ for node i over its adjacent nodes and itself $p \in N_i$.

$$\alpha_{ij} = \text{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{p \in N_i} \exp(e_{ip})}$$

There are multiple ways of computing attention coefficients with alignments of two input vectors [19, 20, 15]. In this work, we concatenate two input vectors and apply a linear feed-forward layer $a \in \mathbf{R}^{2d' \times 1}$. Then a LeakyReLU activation at $\alpha = 0.15$ is used in our experiment to speed up training compared to Tanh non-linearity.

$$e_{ij} = \text{LeakyReLU}\left(a^T \left[Wh_i \parallel Wh_j\right]\right)$$

Multi-head attention [15] is beneficial to our attention graph as it allows the model to jointly attend to information from different representation subspaces at different positions just like multiple convolution filters in one CNN layer. Therefore, we parallel-compute K-head attentions $\{\alpha_{ij}^k\}_{(i,j)\in V}$ with feature transformations W^k and feed-forwrds a^k where $k = 1, 2, \cdots, K$. For k^{th} head, an output graph representation of node i is obtained by the weighted sum of its neighbours' and its own features over attention weights. An ELU non-linearity is used in our experiment for the output of graph attention layer.

$$h'_{k,i} = \sigma\left(\sum_{j \in N_i} \alpha_{ij}^k W^k h_j\right)$$

3.2 Predicting Outcomes with Two-layer Graph

Embedding Layer Similar to word embedding containing more semantic meanings on tokens, we use high-dimensional embedding vectors to encode binary features in EHR. Given one-hot vectors for diagnosis, procedures, lab results (binned and turned into binary variables), and demographic information, we featurize these binary variables with high dimensional embedding vectors $h_i \in \mathbf{R}^d$. Compared to the work of Veličković et al. [3] which take advantage of external features of nodes, we train embedding that contains implicit meaning of medical concepts as features of each nodes.

Input Graph Layer Given EHR data of a patient, where features $x_1, x_2, \cdots, x_N$ are positive, we take them as observed nodes of the graph and fully connect edges among these nodes. We assign features $h_{x_1}, h_{x_2}, \cdots, h_{x_N}$ for nodes $x_1, x_2, \cdots, x_N$ with embedding lookup table. We can obtain K graph representations $h'_{k,x_i} \in \mathbf{R}^{d'}$ for each node with multi-head attention. The graph representations are concatenated as the output of first layer:

$$h'_{x_i} = \|_{k=1}^K \left[\sigma\left(\sum_{j \in N_{x_i}} \alpha_{ij}^k W^k h_j\right)\right]$$

Output Graph Layer The output graph adds the nodes we would like to generate $y_1, y_2, \cdots, y_M$ besides nodes $x_1, x_2, \cdots, x_N$ in the input graph. We also fully connect new nodes with the input graph and each other. As nodes $y_1, y_2, \cdots, y_M$ are not included in input graph, we denote their representations as concatenations of inherent embedding $h'_{y_i} = \|_{k=1}^K h_{y_i}$. Then we apply graph attention layer on fully-connected graph of nodes $x_1, \cdots, x_N, y_1, \cdots, y_M$ and obtain $\tilde{K}$ graph representations $\tilde{h}_{\tilde{k},y_i} \in \mathbf{R}^{\tilde{d}}$. To generate outcome nodes, we take mean of multi-head attention and apply linear layers to transform the graph representation to classification outputs:

$$\tilde{h}_{y_i} = V_{y_i}^T \sum_{\tilde{k}=1}^{\tilde{K}} \left[\frac{1}{\tilde{K}} \sigma\left(\sum_{j \in N_{y_i}} \alpha_{ij}^{\tilde{k}} W^{\tilde{k}} h'_j\right)\right] + b_{y_i}$$

where $V_{y_i} \in \mathbf{R}^{\tilde{d} \times c_{y_i}}, b_{y_i} \in \mathbf{R}^{c_{y_i}}$, c_{y_i} is binary in this case on whether y_i disease, as a output node in new EHR graph, is positive.

Given the output of two-layer graph attention network on nodes $y_1, \cdots, y_M$, we can infer the probability distribution of outcome nodes.

$$p(y_i | x_1, \cdots, x_N) = \text{softmax}(\tilde{h}_{y_i})$$

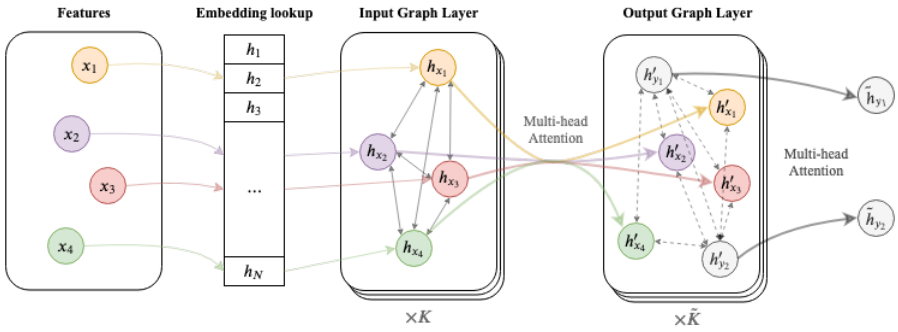


Figure 1: The architecture of GAT model for EHR. Positive features $x_i \in V_{pos}$ in EHR are encoded with embedding vectors $\{h_{x_i}\}_{x_i \in V_{pos}}$ and fully-connected as nodes in graph space. Weights on edges $\{\alpha_{x_i x_j}\}_{x_i, x_j \in V_{pos}}$ and graph representation $\{h'_{x_i}\}_{x_i \in V_{pos}}$ of first GAT layer are computed by multi-head attentions. Nodes representing target disease outcomes are connected to second graph and output representations of $\{\tilde{h}_{y_i}\}_{i \leq M}$ are computed from multi-head attentions on node representations of original nodes in first graph and linear feed-forward layer.

To learn parameters of the model with semi-supervised learning, we minimize the objective function:

$$\mathcal{L} = -\frac{1}{B} \sum_{i=1}^{M} \sum_{b=1}^{B} w_{Y_i}^B \log \left(p(y_i^b = Y_i^b | x_1^b, \cdots, x_N^b) \right)$$

where B is the number of patients in EHR, Y_i^b is true label of y_i outcome for patient b, and x_i^b, y_i^b are corresponding observations and outcome random variables of patient b. To deal with imbalance labels, the loss is weighted by inverse class weight $w_{Y_i}^B$ for each outcome nodes y_i,

$$w_{Y_i}^B = \left\{ \frac{c_{y_i} \cdot |y_i^b = Y_i^b|}{\sum_{c=1}^{c_{y_i}} |y_i^b = c|} \right\}^{-1}$$

where $|y_i^b = c|$ denotes the cardinality of class c labels in training set.

4 Experiments

In our experiments, we predict diagnosis of AD for patients who are not already diagnosed, between 12 to 24 months into the future. For this specific task, only one node (diagnosis on AD) is generated from original graph. In this section, we present details of our dataset, explain the training steps and analyze the result of experiments.

4.1 Data

We work with the EHR data corresponding to 1.64M distinct patients each with unique Medical Record Number (**MRN**), spanning from 2016 to 2019. The variables in this study include a total of 3588 diagnosis (**ICD-10**, **SNOMED**), lab values (**LOINC**), procedures (**CPT**) and demographic information. These variables were selected based on most commonly occurrences and correlation with each other. Among all the features in EHR, we eliminated redundant features that were similar to others (with correlation greater than 0.85). For each patient, we use one year of the data as input (2016.02.20 - 2017.02.19), and aggregate sequential encounter information over 365 days into a single high-dimensional feature vector. For each diagnosis and procedure, we define it positive if the count of positive exceeds some thresholds (one feature per thresholds of 0, 2, and 10). To process features of lab values, we define it positive if there exists one encounter such that the normalized lab value falls within a specific range. Each lab is normalized independently according to all observed values for that lab in the training set. For each lab, we use the ranges corresponding to -10, -3, -1, -0.5, 0.5, 1, 3, 10 standard deviation ranges for that lab. From observation, the diagnosis on ICD-10 codes are sparse as shown in Table 1, which is common nature of EHR. SNOMED features are much denser since we only select the more general SNOMED concepts in our feature set. One advantage

of GNN is that the connection between SNOMED, ICD-10 codes, labs, etc can be represented by edges. Therefore, missing variables can be inferred from graph representation of other variables. For

Table 1: Variable Statistics Summary

Type of EHR	Average Observed Counts	Total Variables
Diagnosis (**ICD-10 Codes**)	4.58	3159
Diagnosis (**SNOMED**)	21.80	195
Lab Values (**LOINC**)	5.19	220
Others (demographic, procedures)	2.40	14

the target variable of our main task, predicting AD, we collect diagnosis records from EHR including AD related diseases[2] on (2018.02.20 - 2019.02.19) and aggregate them as outcomes of AD. As our task is defined as predicting AD in 12 to 24 months, patients who have positive outcome in history window (before 2016.02.19), feature window (2016.02.20 - 2017.02.19) and gap window (2017.02.20 - 2018.02.19) are excluded to avoid data leakage, and 1.61M unique MRNs in EHR remain. For experiments, we split this data into training, validation and test set in ratio of $70\%, 20\%$ and 10% respectively by MRNs, so no patient in training set exists elsewhere.

In summary, our EHR data has properties of high sparsity, long prediction gap and imbalance in classes (only 8174 samples among 1.61M patients have positive outcomes), which makes AD prediction a difficult task.

4.2 Training

Graph attention networks can be memory intensive. In our setting, as the graph is fully-connected, for $O(|F|)$ nodes, $O(|F|^2)$ edge weights should be allocated, where F is number of features. However, for each patient graph, only a few nodes have observed values, so we implement the model in sparse form with Pytorch 1.1.0 to free the memory of unobserved edges in the graph of each patient.

To improve robustness and ability of the model inferring missing features through graph, we randomly mask 10% of nodes during training. To reduce number of epochs and learn from enough positive samples, we upsample positive samples by 50 times. We also randomly downsample 80% negative patients with age under 50 each epoch to accelerate training, as they may not contain significant signals related to AD. The resampling is only done on the training set. Validation and Test set are not adjusted to reflect real world distribution of patients. To find model with least validation loss, we tune hyperparameters: dropout rate p of graph representations at each layer GAT, number of heads K in multi-head attention, negative slope α of LeakyReLU nonlinear activation and dimension d of hidden graph representations. We choose $p = 0.4$ for regularization, $K = 3, \alpha = 0.15$ for stabilizing the attention and $d = 512$ for balance between model capacity and efficiency.

We also use a two-step optimization. We first train 40 epochs with an Adam optimizer at learning rate 1×10^{-4} and a scheduler decreasing learning rate by 60% every 10 epochs. When the model starts overfitting, we freeze the embedding layer and fine tune the model for 5 more epochs with a SGD optimizer with learning rate at 3×10^{-6}. The second training stage after freezing the embedding improves learning of attention parameters in output graph layer without fluctuation on node representations from input layer graph.

4.3 Model Performance

We evaluate performance of our model on test set with ROC curve and precision-recall curve, which are quantified with the area under ROC curve(**AUROC**) and area under precision-recall curve(**AUPRC**) as well as precision at 20% recall(**PPV@0.2NPV**). As outcomes in our dataset are imbalanced, our evaluation focus on precision and recall. Several baseline models are implemented and compared with our method. Evaluations of these models on test set are reported in Table 2.

[2]Agency for Healthcare Research and Quality(AHRQ) at United States Department of Health and Human Services defines the family of Alzhiemer's related dementia, including ICD-10 codes: F01.50, F01.51, F02.80, F02.81, F03.90, F03.91, F04, F05, F07.0, F07.81, F07.89, F07.9, F09, F48.2, G30.0, G30.1, G30.8, G30.9, G31.01, G31.09, G31.1, G31.83, R41.81, R54

Logistic Regression and Random Forest Two statistical machine learning models are introduced as benchmarks for the prediction task on AD. Logistic regression is a simple method for classification task by learning parameters of features. Random forest is an ensemble model of decision trees that takes advantage of bagging mechanism to reduce overfitting. In this experiment, we use original one-hot features as input of these two models.

Multilayer Perceptron MLP is multiple feed-forward network for outcome prediction in EHR[4]. We tuned hyperparameters and obtained best result at hidden size 1024 and dropout 0.5.

Continuous Bag of Words CBOW[21] is a neural network widely used as a language model in NLP, where every token is represented as an embedding vector. Then a sample is represented by the sum or mean of embedding of all its tokens in high-dimensional space. For EHR data, each feature can be viewed as a token, and the representation of a patient can be expressed by mean of embedding of all the positive features.

Graph Representation A simplified version of our method: A graph attention block is used to extract features from embedding in this method. Then similar to GCT [18], the graph representation of a patient is expressed by the mean of node representations that computed from graph block. Non-linearity functions and feed-forward layers are applied to predict positive or negative classes for outcomes.

Table 2: Model Evaluation

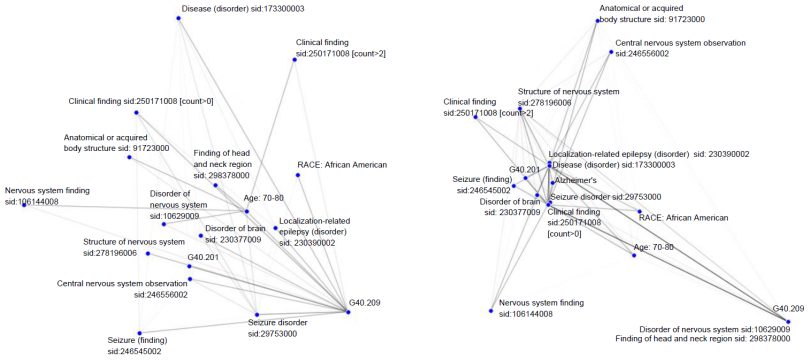
Model	AUROC	AUPRC	PPV@0.2NPV
Logistic Regression	0.766	0.054	0.03
Random Forest	0.742	0.084	0.13
MLP	0.698	0.144	0.38
CBOW	0.783	0.062	0.07
GCT + Avg. Pooling	0.790	0.139	0.40
GAT	**0.802**	**0.205**	**0.54**

4.4 Visualization

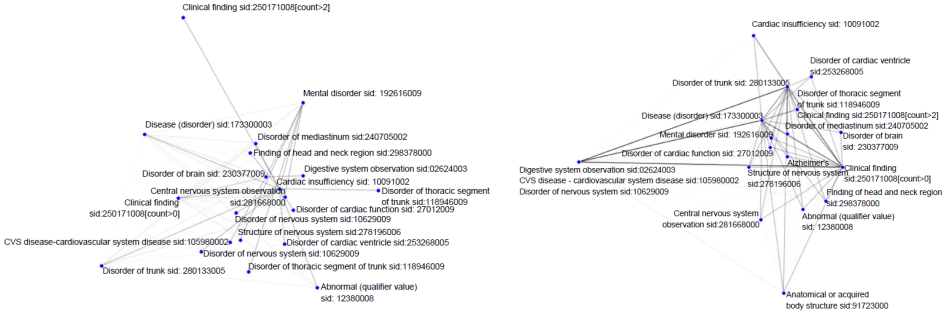
To qualitatively evaluate the model and explain the mechanism of graph network. We visualize EHR graph for an anonymous patient with positive Alzheimer's label. The embedding features and graph embedding for each node h_i and h_i' are projected to 2-D space by t-SNE algorithm [22]. The weights of edge that connecting nodes are computed by the mean of multi-head attentions among adjacent nodes and visualized through channel of opacity (the darker color is, the more weight is on the edge).

The input and output graph layers of two anonymous patients with positive AD outcomes from validation set are visualized in Figure 2. For input graph of patient (A), we can learn that most of nodes have strong attention on nodes of G40.209 (partial symptomatic epilepsy and epileptic syndromes with complex partial seizures) and age 70-80. Patient (B) is a more general case to exemplify how the graph structure deals with the data sparsity in EHR. Unlike the record for patient (A), there is neither ICD-10 diagnosis nor age information in the record of patient (B). Though SNOMED is not as deterministic as ICD-10 codes, stronger connections can still be learned by model on two SNOMED codes that are related to mental disorder and cardiac insufficiency.

After feature extraction in the first graph layer, a clear graph structure forms in each output graph layer. We can notices clear clusters in both samples (right-bottom corner for A, left side for B). Within the clusters, the medical concepts with similarities are gathered. For instance, in output graph of B, digestive system observation, disorder of nervous system and CVS disease, which are closely related in medical meaning, gather to a cluster due to the impact of graph structure. This augments the signal from representation of individual medical concepts in EHR. Also, according to our method in Section 3.2, an output node for AD is generated in this layer. In this view, we did not visualize edges that point to Alzheimer's node from other nodes, because these dominating edges make other relationships hardly visible. The topological structure of the graph suggests that the output layer functions as pooling over different features by attention. The graph representation are summarized by weighted average over other node embedding. For instance, in output graph of patient (A), the Alzheimer's and other nodes mostly attend to two general SNOMED (clinical finds and disorder), which better summarizes the graph representation in input layer than other nodes.



Anonymous Patient (A)



Anonymous Patient (B)

Figure 2: Two samples of patients with positive outcomes are visualized. **Left:** The visualization of input graph. For each node i, the position of node i is the projection of h_i on $\mathbf{R}^2$. **Right:** The visualization of output graph. For each node i, the position of node i is projection of graph representation computed in input layer h'_i on $\mathbf{R}^2$. For both graphs, the thickness of maximum weight in each row of adjacent matrix is scaled to 1; darkness of other edges is scaled by ratio to row max.

4.5 Discussion

In this section, we discuss benchmark models from different perspectives. Our method of GAT on EHR obtains best performance according to Table 2. Through comparisons among these models, we can learn the improvement of performance that different architectures contribute to the task. The logistic regression and random forest are common methods for supervised learning task on one-hot features. However, as a result of the sparse nature of EHR data, less signal are available to differentiate patients. The method of featurizing each one-hot feature with high dimensional vectors can help the model to embed more meaning within the feature. Our experiments demonstrate the ability of high dimensional embedding on feature representation, since CBOW, a simple embedding-based network, exceeds the performance of logistic regression, but it is not as effective as random forest which enhance feature selections through randomness of bagging ensemble method.

CBOW is also a benchmark to demonstrate the ability of graph structures. By comparing CBOW with graph-based models (graph representation, GAT), we learn that the graph network has stronger feature extraction ability than a single embedding layer. Also, with the graph structure, the neural network becomes explainable by visualizing position of nodes and edges connecting them, as shown in Section 4.4. Regarding sparseness of EHR data, it should be noted that more frequent features correspond to more generic conditions, and features with great specification are even more sparse than other features. This means most positive features are general features that do not contain strong signals. Graph network improves the performance of model by focusing attention on more significant features.

To quantitatively analyze the improvement of graph network on predicting sparse samples, we evaluate benchmark models on sparse EHR samples in validation set that have less than 40 observation, and the average counts of their ICD-10 observations is 0.9 (the average of observation counts is 33.97 in Table 1). We select four bins of sparse samples and evaluate the performance of models with AUPRC in Figure 3. Our method over-performed other benchmarks in all four bins, especially from 5 to 40. Within this range, the other models perform poorly, but our method still has a promising performance even in these difficult sub-tasks.

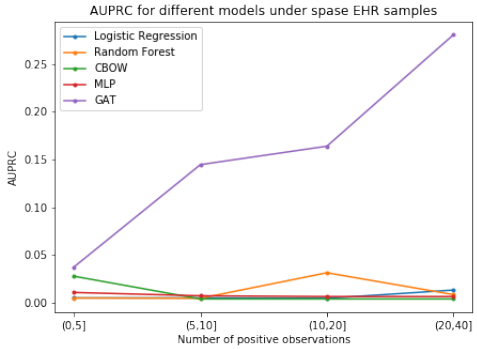


Figure 3: Performances of benchmarks on sparse EHR samples

In Table 2, two variants of our graph network models are also compared. Our GAT on EHR outperforms the model that only use graph representation for classification. Although the graph layer can infer external information through connection of nodes, graph representations are weakened by taking average or summation over node representations. Our method of generating new outcome nodes takes advantage of pooling graph representation by attending on more relevant parts in original EHR graph. As shown in Figure 2, the decision of generating output AD node leverages clusters of node and weights on edges. Therefore, with these advantages of architecture over the other benchmark models, our method achieves best performance in the task of predicting AD.

5 Conclusion

In this paper, we presented a new method for predicting AD, a graph network built upon patient-level EHR data that infers labels of AD node by learning graph structures and representations of other nodes in EHR graph. We addressed and solved two intrinsic problems for exploiting EHR data - the sparsity issue and lack of understandable connection among features in EHR. Our architecture improves feature extraction with a convolution-like graph layer, and enables the model to learn embedding features and weights on fully-connected edges. Multiple experiments of different approaches to AD prediction task are presented, demonstrating the state-of-art performance of our method, especially for sparse patient EHR samples. Explanatory visualization of graph layers presents qualitative analysis on the interpretation of graph representation, structures in each layer as well as the mechanism of classifying AD nodes. Both quantitative and qualitative studies explicate the functions of each layer in our architecture and advantages over previous methods. Our architecture also comes with great potential of generalization to a wide range of prediction tasks as well as several new research topics. Some possibilities of future research are proposed here.

Multitask Prediction For this EHR dataset, we only applied our method in the task of predicting Alzheimer's Diseases. In Section 3.2, we introduce mechanism of predicting multiple outcome nodes through semi-supervised learning on graph network. With this architecture, a potential generalization can be predicting a number of diseases simultaneously given EHR data with multiple outcomes.

Transfer Learning With graph structures and embedding for medical concepts the model obtains from semi-supervised learning, we can make use of them to summarize features of a patient for other tasks. Then for new EHR related tasks, we can freeze parameters in input graph layer as an auto-encoder and fine-tune parameters in the rest part of neural network. With this potential improvement, a large amount of computation can be saved in building models for other tasks. This method can also be applied to solve classification tasks with limited EHR data.

References

[1] K Servick. Another major drug candidate targeting the brain plaques of alzheimer's disease has failed. what's left. *Science*, 10, 2019.

[2] Dev Mehta, Robert Jackson, Gaurav Paul, Jiong Shi, and Marwan Sabbagh. Why do trials for alzheimer's disease drugs keep failing? a discontinued drug perspective for 2010-2015. *Expert opinion on investigational drugs*, 26(6):735–739, 2017.

[3] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[4] Benjamin Shickel, Patrick Tighe, Azra Bihorac, and Parisa Rashidi. Deep EHR: A survey of recent advances on deep learning techniques for electronic health record (EHR) analysis. *CoRR*, abs/1706.03446, 2017.

[5] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, and Jimeng Sun. Multi-layer representation learning for medical concepts. *CoRR*, abs/1602.05568, 2016.

[6] Riccardo Miotto, Li Li, Brian A. Kidd, and Joel T. Dudley. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific Reports*, 6:26094 EP –, May 2016. Article.

[7] Narges Razavian and David Sontag. Temporal convolutional neural networks for diagnosis from lab tests. *arXiv preprint arXiv:1511.07938*, 2015.

[8] Edward Choi, Mohammad Taha Bahadori, and Jimeng Sun. Doctor AI: predicting clinical events via recurrent neural networks. *CoRR*, abs/1511.05942, 2015.

[9] Yu Cheng, Feng Wang, Ping Zhang, and Jianying Hu. Risk prediction with electronic health records: A deep learning approach. In *SDM*, 2016.

[10] Yikuan Li, Shishir Rao, Jose Roberto Ayala Solares, Abdelaali Hassaïne, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi Khorshidi. BEHRT: transformer for electronic health records. *CoRR*, abs/1907.09538, 2019.

[11] Zachary Lipton, David Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. 11 2015.

[12] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *CoRR*, abs/1312.6203, 2013.

[13] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

[14] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[16] Edward Choi, Cao Xiao, Walter F. Stewart, and Jimeng Sun. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. *CoRR*, abs/1810.09593, 2018.

[17] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F. Stewart, and Jimeng Sun. GRAM: graph-based attention model for healthcare representation learning. *CoRR*, abs/1611.07012, 2016.

[18] Edward Choi, Zhen Xu, Yujia Li, Michael W. Dusenberry, Gerardo Flores, Yuan Xue, and Andrew M. Dai. Graph convolutional transformer: Learning the graphical structure of electronic health records. *CoRR*, abs/1906.04716, 2019.

[19] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

[20] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[21] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[22] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.