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

Modeling Latent Comorbidity for Health Risk Prediction Using Graph Convolutional Network

Rui Wang,[†] **Ming-Ching Chang,**[†] **Marleen Radigan,**[‡]
[†]University at Albany, State University of New York, NY, USA
[‡]New York State Office of Mental Health, NY, USA
rwang3@albany.edu mchang2@albany.edu marleen.radigan@omh.ny.gov

Abstract

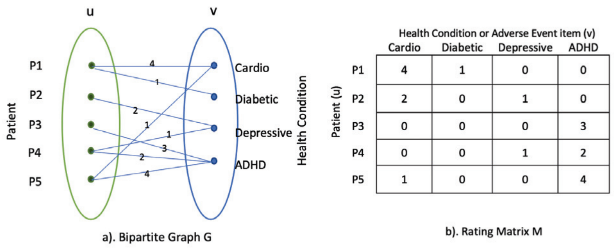
We propose to apply deep Graph Convolutional Network (GCN) for the analysis and prediction of patient health comorbidity from sparse health records. Patient health data are represented in a powerful graph structure. Specifically, *healthcare conditions* including health diagnosis categories, hospitalizations, injury incidents are represented as a type of graph nodes, and *patient attributes* including demographics, aid categories are represented as another type of nodes. Health records for individuals including diagnostic results, hospital visits are represented as graph links connecting the two node types, such that the whole record forms as a sparse *bipartite graph*. Our hypothesis is that patient health trend, disease prognosis, treatment, and their latent correlations can all be modeled by recovering the missing links in this bipartite graph (the *link prediction* problem). Starting with sparse patient data or incomplete records, graph completion and record fusion via end-to-end GCN modeling can lead to robust prediction across individual patients and health records. Application in estimating health prognosis shows the efficacy of the proposed method compared to existing approaches.

Keywords: health risk prediction, medical records, health data, comorbidity, bipartite graph, link prediction, recommendation system, Graph Convolution Network, GCN.

1 Introduction

In medicine, *comorbidity*—the presence of one or more mental or physical disorders co-occurring for a patient—constitutes a challenge for healthcare professionals and the healthcare system (Valderas, Starfield, and Sibbald 2009). Many studies (Piane and Smith 2014; Bhattacharya and Shen 2014) suggest that mental disorders correlate with chronic conditions, and mental/medical disorders usually share common risk factors (Antonaci, Nappi, and Galli 2011). From the perspective of public health, understanding the risk factors and latent relations among disorders and comorbidity can lead to early prediction of potential disorders and timely treatment.

Patient health records and person-specific attributes can be organized into a graph structure as a powerful representation for analysis. We focus on two types of attributes for comorbidity modeling and health risk prediction. First, *patient attributes* includes demographics, insurance converge levels, and aid categories can be organized as a type of graph nodes (one node per patient). Secondly, the *healthcare conditions* including physical/mental health, diagnostic results, hospitalization reasons, injury incidences of each patient can be represented as another type of graph nodes. This way, each health record or insurance events can be represented as graph links connecting the two types of nodes, such that the whole health record can be organized as a bipartite graph as shown in Fig. 1(a). Graph data can be stored as a sparse adjacency matrix, where edge weight indicates the condition severity level as shown in Fig. 1(b). Using this representation, comorbidity relations can be modelled and latent conditions can be recovered by predicting the missing links from the sparse bipartite graph.



	Cardio	Diabetic	Depressive	ADHD
P1	4	1	0	0
P2	2	0	1	0
P3	0	0	0	3
P4	0	0	1	2
P5	1	0	0	4

Figure 1: We formulate health risk prediction as a bipartite graph matrix completion problem. The input to our approach is provided as a *rating matrix* consisting of the health conditions of individual patient records. Each graph link spanning a patient and a health condition represents a corresponding health record with a severity condition level.

Link prediction (Liben-Nowell and Kleinberg 2007) is a fundamental problem in network science. Conventional methods rely on parametric statistics, correlation coefficient among nodes, logistic regression, cliques (of three nodes) as network topology to predict latent structures (see § 2).

In this paper, we apply Graph Convolutional Network (GCN) (Kipf and Welling 2016a) to solve the health link prediction problem. GCN has drawn growing attentions due to its end-to-end, model-free capability in automatic learning of complex statistical interactions between features from high-dimensional data. We present patient characteristics along with their health conditions and outcome using undirected bipartite graph. The data is stored as matrix structure, then apply Graph Convolutional Network (GCN) deep learning framework to fill in the sparse matrix of data. Subsequently we use the obtained relation for individual emerging or latent health condition prediction, meanwhile predict the risk of two health outcomes simultaneously, *i.e.* hospitalization and injury incidents. The aim is to predict the severity risk of each chronic condition for each patient, as well as the frequency category of hospitalization and other associated adverse event *i.e.* injury, given their known conditions for Medicaid insurers. Fig. 1 shows an example rating matrix M and a bipartite graph.

Contribution of this paper is two-fold. (1) We formulate the health record modeling as a bipartite graph matrix completion problem, and apply the recent deep graph convolution methods as an effective solution. (2) The approach is applied on health risk prediction with superior accuracy outperforming comparison methods.

The rest of this paper is organized as follows. § 2 surveys fundamentals of graph link prediction, prior arts of matrix completion and Graph Convolutions Network (GCN) in recommendation systems. § 4 describes our formulation of patient health and medical records into a bipartite graph representation, and how we apply GCN for matrix competition as a solution for health risk prediction. § 5 discusses a real-world sensitivity analysis evaluation performed on a large dataset containing samples of patient claim data. § 6 describe performance evaluation of proposed approach and comparison with baseline methods used in collaborative filtering for recommendation systems. § 7 concludes this paper and discuss future directions.

2 Background

Healthcare data analytics. Logistic algorithm is the most common classification method used in healthcare study that models the relationship between binary outcome and independent variables. The importance of each of the explanatory variables is assessed by carrying out statistical tests on the significance of coefficients. These models rely on predefined heuristics, *e.g.* assuming little or no multi-collinearity among the independent variables. Recently, deep learning techniques have been applied to clinical applications for outcome prediction, including Autoencoder(AE), Long Short-Term Memor (LSTM), Restricted Boltzmann Machine (RBM), and Deep Belief Network (DBN) (Shickel et al. 2018). However, the applied convolutional operation is only appropriate for grid structured data. Meanwhile, they are widely known to be difficult to train and computationally heavy.

Recommendation system is an application where link prediction algorithms can be directly applied on a graph structure, for *e.g.* social network analysis or movie preference prediction (Van den Berg, Kipf, and Welling 2017). Traditional approaches involve the calculation of a heuristic similarity score for a pair of nodes, such as the number of common neighbors or the shortest path length connecting the nodes, where pairs of nodes with the highest similarity scores are considered the most likely edges (Lü and Zhou 2011). Collaborative filtering is a family of algorithms working with a rating matrix to find similar users or items and calculate rating based on ratings of similar users. The matrix is typically huge and sparse, with missing values. Data driven machine learning can be used to learn a function that predicts utility of items for each user. k-Nearest Neighbors (kNN) based on cosine or correlation similarity cannot handle sparsity well, as there may not exist enough samples in the neighborhood. Matrix factorization methods work by decomposing the user-item interaction matrix into the product of two matrices (Koren and Bell 2009), with an aim of dimension reduction. The advantage of it over KNN is that even though two users have not rated any same items, it is still possible to find the similarity between them if they share the similar underlying latent features.

Matrix completion (Candès and Recht 2009) is the task of filling in the missing entries of partially observed matrix. The recommendation problem can be posed as a matrix completion problem, starting with a sparse matrix of known user-item preferences. The underlying assumption is that a low-dimensional representation of users and items exists, which can be modeled via *e.g.* a low-rank matrix. Popularly methods in this category includes Alternative Least Square (ALS), spectral regularization with soft threshold, Alternating Direction Method of Multipliers (ADMM), *etc.*

The **Graph Convolutional Matrix Completion** (GC-MC) (Van den Berg, Kipf, and Welling 2017) for recommendation systems perform link prediction on graph using a graph-based auto-encoder framework, building upon recent deep GCNs. The auto-encoder extracts latent features from a user preference dataset through a form of message passing. These latent user-item preference represented on a bipartite interaction graph are used to derive the desired rating through a bi-linear decoder.

3 Healthcare Problem Statement

Our task is to predict the risk of latent comorbidity using administrative healthcare claim data. The population included ($N = 750K$) are a large subset of individuals who were continuously eligible for New York State Medicaid in 2017 and also had a health condition or event (determined by claim data including physical, behavioral, inpatient admission, and injury) in the year. For demonstration purpose, a specific condition or event is identified by invoice type and ICD-10-CM diagnostic codes. The severity level (*i.e.* 1,2,3,4) of a condition or event for each individual is arbitrarily attributed by the occurrence of corresponding diagnostic codes on service claims in the year. e.g. for a patient's heart disease condition, the severity level is categorized as 1, 2, 3 and 4 when ≤ 2, $3 - 10$, $11 - 40$, ≥ 40 times of visits with such diagnostic codes respectively.

The goal is to predict individual's missing condition or event items, *e.g.* under-reported or emerging items, based on their existing conditions and past adverse events, adjusted by demographic characteristics and Medicaid enrollment eligibility category. This task can be cast as a *Link Prediction* problem, considering both person and item node features. A total of 42 conditions are considered:

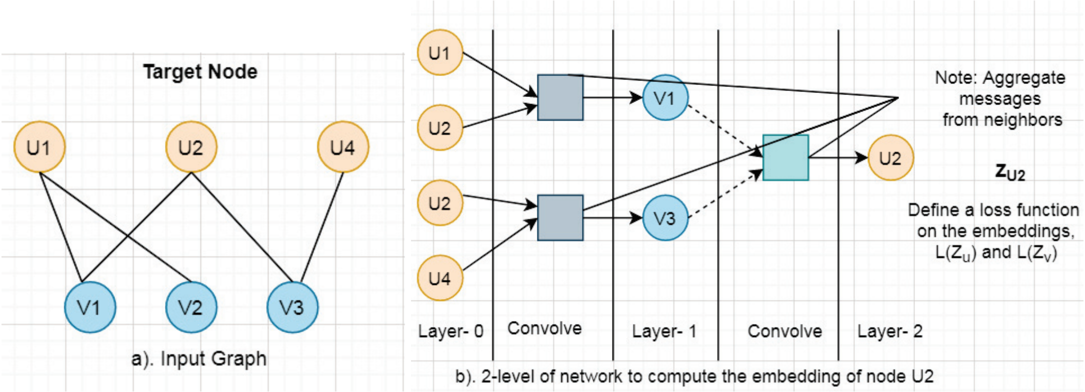

Target Node
U1
U2
U4
V1
V2
V3
a). Input Graph
U1
U2
U2
U4
V1
V3
U2
Note: Aggregate messages from neighbors
$\mathbf{z}_{U2}$
Define a loss function on the embeddings, $L(Z_u)$ and $L(Z_v)$
Layer- 0
Convolve
Layer- 1
Convolve
Layer- 2
b). 2-level of network to compute the embedding of node U2

Figure 2: Node aggregation for link prediction based on neighboring nodes with similar connectivities.

- 10 physical health conditions: Cancer, Chronic obstructive pulmonary disease (COPD), Cerebral infection, Heart disease, Obesity, Arthritis, Diabetes, Dyslipidemia, Epilepsy, Hypertension
- 23 behavior health conditions: Physiological conditions, Psychoactive substance use (Alcohol, Opioid, Cannabis related), Non-mood psychotic, Mood, Bipolar, Affective, Depressive, Anxiety, Reaction to severe stress (PTSD), Physical, Eating, Personality, Obsessive compulsive (OCD), Intellectual disabilities, Pervasive and developmental disorders, Behavioral and Emotional, Attention-deficit hyperactivity (ADHD), Conduct, Emotional, Tic.
- 9 conditions related to adverse events: 3 types of inpatient admissions as well as 6 types of injuries: risk factors, self harm, suicide attempt, symptoms, homicidal suicidal idea, other injury.

In our bipartite graph representation, each condition node stores their corresponding category of conditions or event type as the node features. Attributes for each patient node includes the age as the end of the year; gender (male/female); race/ethnicity (white-non-Hispanic, black-non-Hispanic, Hispanic or unknown); and Medicaid enrollment aid category (Foster Care, Supplemental Security Income - SSI, Temporary Assistance for Needy Families-TANF, and other).

4 Methods

In principle, the objective of matrix completion (MC) is to approximate the matrix with a low-rank matrix. *i.e.* $\min_{\mathbf{x}} \text{rank}(\mathbf{X})$ *s.t.* $x_{ij} = y_{ij}, \forall_{y_{ij}} \in \Omega$, where $\mathbf{X} \in \mathbf{R}^{m \times n}$: the matrix we need to learn; $\mathbf{Y} \in \mathbf{R}^{m \times n}$: the original matrix (including the known entries and the missing data); Ω is the set of known entries in Y; where rank($\mathbf{X}$) is the maximal number of linearly independent columns of $\mathbf{X}$, so that rank($\mathbf{X}$) is minimized when the variables are in a smallest subspace. However, rank minimization is an intractable problem. Among a few methods (Ying et al. 2018; Monti, Bronstein, and Bresson 2017) to approximate the solution, the Graph Convolutional Matrix Completion (GC-MC) (Van den Berg, Kipf, and Welling 2017) framework is adopted in this study, as it focuses on the inclusion of graph-based side information, also based on neural message passing directly on the interaction graph and models the rating graph directly in a single encoder-decoder step.

Specifically, we apply GC-MC (Van den Berg, Kipf, and Welling 2017) on the health record represented as a bipartite graph, and cast the *link prediction* as a matrix completion problem. The health record bipartite graph $G(U, V, E)$ in its initial state consists of links such as $(u_i, v_j) \in E$ (individual medical records), connecting a *patient (user) node* $u_i \in U$ to a *health condition (item) node* $v_j \in V$. A link weight $r \in \{1, ..., R\}$ represents ordinal severity levels. In parallel, a matrix $\mathbf{M}(|U| \times |V|)$ stores the observed health severity data for $|U|$ patients and $|V|$ health conditions as nonzero entries, i.e. $\mathbf{M}_{ij}$ represents an observed severity of condition item v_j on patients u_i . The matrix completion task is to predict the unknown or latent entries of $\mathbf{M}$.

The input graph for each weight r is represented by an adjacency matrix $\mathbf{M}_r$ where all entry values are binary. A value of 1 or 0 at row i and column j indicates whether or not a weighted of r link exists between vertex i and vertex j, respectively. The final input matrix M consists of $\mathbf{M}_1, ..., \mathbf{M}_R$.

As an option, the node Features of patient or health condition can be conceptualized in the form of vectors $\mathbf{x}_k$ for node k where $1 \leq k \leq |U| + |V| = N$, such that the input matrix $\mathbf{X} = [\mathbf{x}_1^T, ..., \mathbf{x}_N^T]^T$ containing the node features for the graph convolution layer is then chosen as an identity matrix, with a unique one-hot vector for every node in the graph. In such way, the graph-based node information can be incorporated seamlessly.

The detailed system architecture is describe in the following subsections. In summary, firstly graph convolutional encoder, a variant of collaborative filtering auto-encoder (Salakhutdinov, Mnih, and Hinton 2007), takes the input graph formulated as matrix $\mathbf{M}$, and an optional node feature matrix $\mathbf{X}$, then produces patient and health condition item node embeddings (or latent representations) $\mathbf{z}_i^u$ ($\mathbf{z}_j^v$) for a single patient i (item j). They are done through a form of message passing on the bipartite interaction graph. In the second phase, the patient and health condition item embedding pairs are used to reconstruct the the links for each edge type (rating) through a bi-linear decoder $\tilde{\mathbf{M}} = g(\mathbf{Z}_u, \mathbf{Z}_v)$. Fig. 2 demonstrates GCN model underlying prediction architecture.

4.1 Graph Auto-Encoder

The conventional autoencoder (VE) is an unsupervised artificial neural network that learns how to efficiently compress and encode data to a lower-dimensional representation (embedding), then learns how to use the embedding to reconstruct the original input. A variational autoencoder (VAE) embeds the input to a regularised distribution, then new data or a random sample is generated from the distribution.

Variational graph autoencoder (VGAE) is a framework for unsupervised learning on graph-structured data based on the variational auto-encoder (VAE). It achieves competitive results on many link prediction tasks. For a non-probabilistic variant of the VGAE model, the embeddings Z and the reconstructed adjacency matrix $\hat{A}$ as follows (Kipf and Welling 2016b):

$$\hat{\mathbf{A}} = \sigma(\mathbf{Z}\mathbf{Z}^T), \text{with } \mathbf{Z} = f(\mathbf{X}, \mathbf{A}). \tag{1}$$

The graph encoder model above takes feature matrix X and a graph adjacency matrix A, and produce an $N \times E$ node embedding matrix $Z = [z_1^T, ..., z_N^T]^T$. In our bipartite graph $G = (\mathcal{W}, \mathcal{E}, \mathcal{R})$ setting, the encoder is analogously formulated as $[U, V] = f(X, M_1, ..., M_R)$, where $M_r \in \{0,1\}^{N_u \times N_v}$ is the adjacency matrix for rating or severity type $r \in R$, such that M_r contains 1's for those elements for which the original rating matrix M contains observed ratings with value r. U and V are now matrices of patient ($N_u \times E$) and item ($N_v \times E$) embeddings, respectively.

4.2 Graph convolutional encoder

For a graph G, convolutional methods represent a node embedding is generated based on local neighborhood and the node features with aggregation algorithms. Nodes have embeddings at each layer, where the number of layers is arbitrary. In a transductive setting, the convolution results in link or rating specific type messages r passing from item j to patient i to be formulated as $\mu_{j \to i,r} = \frac{1}{c_{ij}} \mathbf{W}_r \mathbf{x}_j$. Here, c_{ij} is a normalization constant $\sqrt{|\mathbf{N}(u_i)||\mathbf{N}(v_j)|}$, and $\mathbf{N}(u_i)$ denotes the set of neighbors of node i, $\mathbf{W}_r$ is an link-type specific parameter matrix, and $\mathbf{x}_j$ is the initial feature vector of node j. The accumulated messages at every node i for each rating type r can be expressed as

$$\mathbf{h}_i^u = \sigma[\text{accum}(\sum_{j \in \mathbf{N}_i(u_i)} \mu_{j \to i,1}, ..., \sum_{j \in \mathbf{N}_R(u_i)} \mu_{j \to i,R})] \tag{2}$$

where accum() denotes a stack accumulation operation, $\sigma()$ denotes an element-wise activation function ReLU(.)=max(0,.), such that patient i and health condition item j embedding are expressed as $\mathbf{u}_i = \sigma(\mathbf{W}_u \mathbf{h}_i)$, $\mathbf{v}_j = \sigma(\mathbf{W}_v \mathbf{h}_j)$ respectively.

4.3 Bi-linear decoder

In general GCN settings, decoder model $\tilde{\mathbf{A}} = g(\mathbf{Z})$ takes pairs of node embeddings $(\mathbf{z}_i, \mathbf{z}_j)$ and predicts respective entries $\tilde{\mathbf{A}}$ in the adjacency matrix, corresponding to graph link reconstruction. Here, a bi-linear decoder is applied to treat each rating level a separate class in the bipartite interaction graph. Specifically,

$$p(\tilde{\mathbf{M}}_{ij} = r) = \frac{e^{\mathbf{u}_i^T \mathbf{Q}_r \mathbf{v}_j}}{\sum_{s=1}^R e^{\mathbf{u}_i^T \mathbf{Q}_s \mathbf{v}_j}}, \tag{3}$$

where $\mathbf{Q}_r$ a trainable parameter matrix of shape $E \times E$ and E is the dimensionality of hidden user or item representations $\mathbf{u}_i(\mathbf{v}_j)$. The predicted rating is computed as $\tilde{\mathbf{M}}_{ij} = g(\mathbf{u}_i, \mathbf{v}_j) = \mathbf{E}_{p(\tilde{\mathbf{M}}_{ij}=r)}[r] = \sum_{r \in R} p(\tilde{\mathbf{M}}_{ij} = r)$.

4.4 Model training and loss function

The training objective is to minimize loss function, implemented as the negative log likelihood of the predicted ratings $\tilde{M_{ij}}$:

$$\mathcal{L} = -\sum_{i,j;\Omega_{ij}=1} \sum_{r=1}^{R} I[\mathbf{M}_{ij} = r] \log p(\tilde{\mathbf{M}}_{ij} = r). \tag{4}$$

Ω is used to filter out the unobserved ratings for the optimization. In order to minimize over-fitting, we randomly drop out all outgoing messages of a particular node with a probability $p_{dropout}$.

For the GAE model implementation, we use sparse matrix multiplications with complexity $\mathcal{O}\|\mathcal{E}\|$. The graph convolutional encoder can be vectorized as

$$\begin{bmatrix} U \\ V \end{bmatrix} = f(X, M_1..., M_R) = \sigma(\begin{bmatrix} H_u \\ H_v \end{bmatrix} W^T) \tag{5}$$

with $\begin{bmatrix} H_u \\ H_v \end{bmatrix} = \sigma(\sum_{r=1}^{R} D^{-1} \mathcal{M}_r X W_r^T)$, and $\mathcal{M} = \left(\begin{smallmatrix} 0 & M_r \\ M_r^T & 0 \end{smallmatrix} \right)$

5 Experiment

The proposed GC-MC approach is applied to a large sample of New York State Medicaid enrollees who (1) has full year of coverage in 2017 and (2) has received a behavior health diagnose or medication. The diagnoses of each containing 9 physical or 23 behavioral health conditions during the year for each patient are classified as ratings of $\{1, 2, 3, 4\}$. The rating of 4 indicates most frequently diagnosed. More than one diagnoses can be recorded on one claim or visit. As event items, inpatient admissions for mental health, substance use, and physical health reasons, as well as 6 types of injuries are also classified on the same scale set of 1 through 4 according to their frequencies. The datasets contain $596,475$ patients or users, and 42 items, and their respective user-item interaction graphs when applicable. Among all examined physical health conditions, most of prevalent is COPD (23.1%), then followed by Obesity (16.3%). 14.3% had a ADHD diagnosis, and 15.2% had an anxiety issue.

In order to obtain optimized hyper-parameters, we apply $80/20$ train/validation method and split the original training set, where interactions of randomly selected testing users are cross validated to estimate the performance of recommendation on unseen ratings. We compare the performance of the model with and without node features or characteristics,

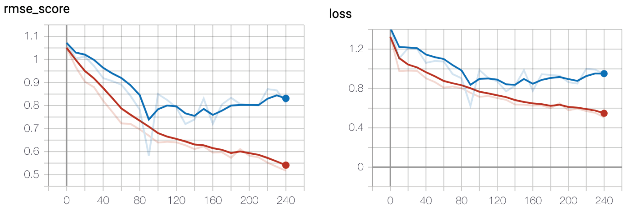


Figure 3: (a) RMSE (b) Loss (dropout= 0.7, accum = stack, includes node features), red: training, blue: validation.

Table 1: Number of patient and rating frequencies used in the experiments. There are 42 conditions. Rating levels are 1,2,3,4.

	Dataset	Patients	Ratings	Density
Full	Training/ Validation (80:20)	596475	1389244	0.055
	Testing	157877	347311	0.052
Sub 1	Training/ Validation (80:20)	228304	277848	0.029
	Testing	65835	69462	0.025
Sub 2	Training/ Validation (80:20)	388798	555697	0.034
	Testing	125380	138924	0.026

such as age and gender for patient node, health category for item node. Both types of models are trained for 250 full-batch epochs. For model evaluation, we compute average of (Root Mean Square Error) RMSE [1] and Cross-entropy loss on the testing data resulting from 5 runs. Table 1 summarizes the dataset statistics.

Parameter settings in our experiments included: 1) Accumulation function, set as stack or sum (as explained in the methods). 2) Dropout rate is tested at multiple values, but set at 0.7. 3) Learning rate is set to 0.01 for the Adam optimizer. 4) Basis weight matrics is set to 2 for decoder's weight sharing. 5) Layer sizes of 500 and 75 for the graph convolution. Fig. 3 and Fig. 4 shows RMSE and loss scores at each epoch; training *vs.* validation (dropout = 0.7, accum = stack). The only difference is that nodes attributed are included or not.

Additionally, we conduct sensitivity analysis to access the model stability. The experiments described above are repeated on randomly selected sub-samples, the final results are comparable in terms of average RMSE and Loss, which are displayed and compared in the Table 2.

6 Evaluation

To evaluate the proposed GC-MC approach, we compare the average of RMSE scores obtained from the model and other baseline methods using the same training set. Table 3 shows comparison results.

Table 2: Comparison of average RMSE and loss scores over 5 runs for $80/20$ training/test dataset splits, with and without node features.

Sample	Feature RMSE	No Feature RMSE	Feature Loss	No Feature Loss
Full	0.8622	0.8590	1.0634	1.0750
Sub 1	0.8594	0.8596	1.0577	1.0649
Sub 2	0.8442	0.8382	1.0282	1.0151

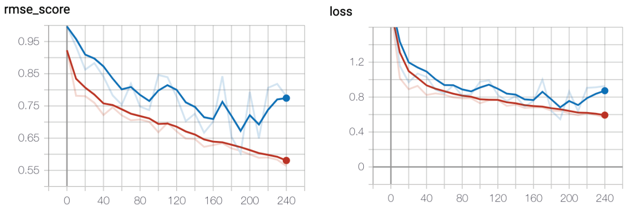


Figure 4: (a) RMSE (b) Loss (dropout= 0.7, accum = stack, without node features), red: training, blue: validation.

6.1 Baseline methods

We perform comparison experiments using the following 3 baseline methods that are commonly used in recommendation systems

Association Rules (AR): AR has the ability to efficiently identify what items appear together in the same session. It has been widely used in recommendation system. Items that are frequently present together are connected with a link in the graph. Rules mined from the interaction matrix should have at least some minimal support and confidence. Support refers frequency of occurrence, and confidence means that rules are not often violated.

K-nearest-neighbor (KNN) graph is a standard method of collaborative filtering (CF), which can be performed based on the users or the conditions on the bipartite graph. The user-based models the ratings with an $n \times m$ matrix, where user u_i, $i = 1, ..., n$ and conditions p_j, $j = 1, ..., m$. The goal is to predict the rating r_{ij} if target user i did not rate a condition j. The process is to calculate the similarities between target user i and all other users, and select the top X similar users. The weighted average of ratings from these X users are calculated as:

$$r_{ij} = \frac{\sum_k \text{similarities}(u_i, u_k) r_{kj}}{\text{total ratings}}, \tag{6}$$

where similarity score can be calculated using *Pearson Correlation* or *Cosine Similarity*. Analogously, for condition-based, two items are similar when they received similar ratings from a same user, so that we can make prediction for a target user on a condition by calculating weighted average of ratings on most X similar items from this user.

[1] $RMSE = \sqrt{\frac{\sum_{i=1}^{N}(r_i - \tilde{r}_i)^2}{N}}$.

Table 3: Comparison of average RMSE of a sample of training set, with and without node features.

Model	RMSE	Feature
AR (Agrawal and Srikant 1994)	1.92	No
MF (Singh and Gordon 2008)	1.86	Yes
Proposed method (GC-MC)	0.86	Yes

Collective Matrix Factorization (MF): a method that decompose the original sparse matrix to low-dimensional matrices with latent factors/features and less sparsity. The goal is to find a set of latent features, with align with a user and an item. The non-negative factorization has a loss function that is non-convex, that can be solved by a few popular algorithms e.g Stochastic Gradient Descent (SGD), is to take derivatives of the loss function with respect to each variable in the model and update feature weights one individual sample at a time, until convergence; Alternative Least Square (ALS) is alternatively to hold user or item factor matrix constant, adjust item or user factor matrix by taking derivatives of loss function and setting it to 0, and then hold item factor matrix constant while adjusting user factor matrix, repeat until convergence.

Table 3 shows that the proposed GC-MC produces better link predictions of healthcare data with smaller RMSE, in comparison with baseline methods. Note that we have also applied KNN in our comparison experiment, but no results are produced due to its limited scalability issue.

Discussion: This end-to-end GCN framework has recently emerged as a powerful deep learning-based approach for link prediction. It learns a target node's representation by propagating neighbor information in an iterative manner until a stable fixed point is reached. It's supported by a large body of recent work to apply the novel approach over simple patient and health condition network.

7 Conclusion

In this paper, we apply a graph link prediction technique on health records for health risk prediction, as an application in public healthcare. Patient records from NYS Medicaid public health data are formulated as a bipartite graph, and a recent deep Graph Convolutional Matrix Completion (GC-MC) network is applied to generate risk predictions. Performance are evaluated and compared with three baseline methods, demonstrating the efficacy of the proposed method.

Future work of this study includes the generation of more dynamic and refined predictions by leveraging the temporal or semantic/causal components of the multi-session health conditions and records. Among the others, subject-specific and explainable predictions are desired properties of health risk and prediction and prevention.

References

Agrawal, R., and Srikant, R. 1994. Fast algorithms for mining association rules in large databases. *VLDB* 487–499.

Antonaci, F.; Nappi, G.; and Galli, F. 2011. Migraine and psychiatric comorbidity: a review of clinical findings. *The Journal of Headache and Pain* 12(2):115–125.

Bhattacharya, R., and Shen, C. 2014. Excess risk of chronic physical conditions associated with depression and anxiety. *BMC Psychiatry* 14(1).

Candès, E., and Recht, B. 2009. Exact matrix completion via convex optimization. *JAMA Psychiatry* 9(6):717–772.

Kipf, T. N., and Welling, M. 2016a. Semi-supervised classification with graph convolutional networks. *arXiv:1609.02907*.

Kipf, T. N., and Welling, M. 2016b. Variational graph auto-encoders. In *arXiv 1611.07308*.

Koren, Y., and Bell, R. 2009. Exact matrix completion via convex optimization. *JAMA Psychiatry* 42(8):30–37.

Liben-Nowell, D., and Kleinberg, J. 2007. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci.* 58(7):1019–1031.

Lü, L., and Zhou, T. 2011. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications* 390(6):1150–1170.

Monti, F.; Bronstein, M. M.; and Bresson, X. 2017. Geometric matrix completion with recurrent multi-graph neural networks. *CoRR* abs/1704.06803.

Piane, G., and Smith, T. 2014. Building an evidence base for the co-occurrence of chronic disease and psychiatric distress and impairment. *Preventing Chronic Disease*.

Salakhutdinov, R.; Mnih, A.; and Hinton, G. 2007. Restricted boltzmann machines for collaborative filtering. In *ICML*, 791–798. New York, NY, USA: ACM.

Shickel, B.; Tighe, P.; Bihorac, A.; and Rashidi, P. 2018. Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE Journal of Biomedical and Health Informatics* 22(5):1589–1604.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *KDD*, 650–658. New York, NY, USA: ACM.

Valderas, J.; Starfield, B.; and Sibbald, C. 2009. Defining comorbidity: Implications for understanding health and health services. *The Annals of Family Medicine* 7(4):357–363.

Van den Berg, R.; Kipf, T.; and Welling, M. 2017. Graph convolutional matrix completion. *arXiv:1706.02263*.

Ying, R.; He, R.; Chen, K.; and Eksombatchai, P. 2018. Graph convolutional neural networks for web-scale recommender systems. *KDD* 50(2):129–135.