Patient Similarity using Heterogenous Graph Neural Networks

Razal Minhas, Julian Gomez, and Sonal Pardeshi

{ razal, julianegomez, pardeshi }@utexas.edu

ABSTRACT

Electronic Health Data (EHR) data volume is growing at an exponential rate, bringing opportunities for advanced analytics to improve patient health care. Despite the advances in technology 97% of the data produced by hospitals goes unused [10]. In this paper we evaluate an approach to patient similarity detection, using a heterogeneous graph representing patients, diagnoses and procedures nodes connected with meaningful relationships. We use MIMIC data and Relational Graph Convolutional Networks (RGCN). Our objective is to learn patient embeddings through contrastive learning and find similar patients across various clinical dimensions. This paper lays the foundations that generate embeddings which can help physicians find patients with similar profiles thereby improving treatment decisions by referencing applicable historical cases.

1. INTRODUCTION

EHRs contain rich information across different domains including diagnoses, procedures, medications and demographics. Physicians and healthcare practitioners need to identify similar patients to inform treatment decisions, but this is a challenging process given the high-dimensionality and heterogeneous medical data. Typical approaches for patient similarity that use feature engineering do not capture the complex nature of relationships between the different medical entities.

We use graphs to represent patients, diagnoses and procedures as a heterogeneous network and apply Relational Graph Convolutional Networks (R-GCN) to learn patient embeddings. The embeddings can be used to identify similar patients using cosine similarity, providing physicians with an easy way to find applicable historical cases.

The graph representation of EHR data preserves the semantics of the medical relationships and a contrastive learning approach using R-GCN is used to learn the patient embeddings. Then a similarity framework is used to identify clinically relevant patient matches.

2. RELATED WORK

Many studies have explored patient similarity metrics for clinical decisions. Early approaches relied on feature-based similarity [2], and more recent works have incorporated deep learning [3]. We find that graph-based methods can show promise due to their ability to model complex relationships [4]. Graph neural networks have been used for various healthcare challenges, including disease prediction [5] and treatment recommendations [6]. R-GCNs were introduced [7] to handle heterogeneous graphs with multiple relationship types, making them suitable for medical data with diverse relationships. Contrastive learning has become a powerful way for learning representations without extensive labeling [8] which we explore in this paper.

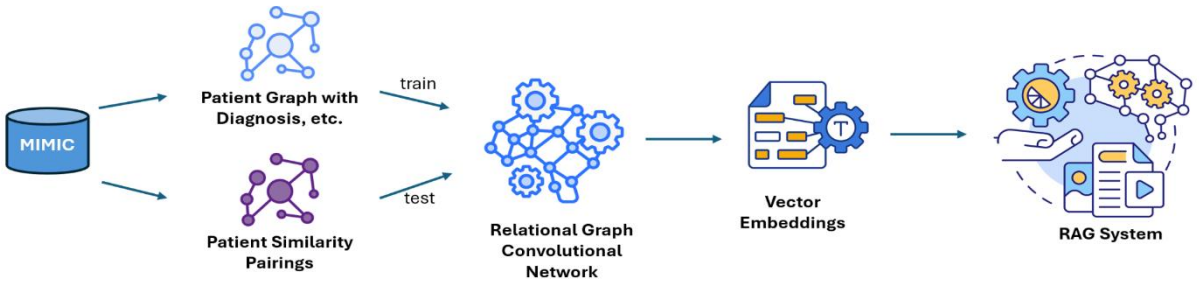

Figure 1: Data flow from training through inference

3. DATA AND FEATURE ENGINEERING

3.1 Dataset

We used the MIMIC-III (Medical Information Mart for Intensive Care) database, which consists of de-identified patient health data for approximately 46.5K patients that were admitted to the intensive care unit at Beth Israel Deaconess Medical Center. It includes 651K diagnoses and 240K procedures. The data was down sampled to 25% of the original size to accommodate the limitations of the GPUs available for this project.

Each patient could have multiple admissions therefore only the latest admission of each patient was considered. Accordingly, the diagnoses and procedures associated with each patient's latest hospital admission were used. The final data size used for the graphs consisted of 11.6k patients, 124k diagnoses and 47k procedures.

3.2 Feature Selection

Patient features include demographic data such as gender and age. The date of birth at the time of admission was used to calculate the age of the patient. Age was categorized further into buckets to provide meaningful groupings and used as an attribute for the patients. Diagnosis features included ICD-9 codes and their descriptions, while procedure features include procedure codes and descriptions. The MIMIC-III tables *d_icd_diagnoses* and *d_icd_procedures* were merged with the *diagnoses_icd* and *procedures_icd* respectively to obtain the descriptions of each object.

3.3 Graph Creation

We constructed a heterogeneous graph using the open source NetworkX library. The *networkx* graph was created with patient nodes representing individual patients, diagnosis nodes representing ICD-9 diagnosis codes and procedure nodes representing medical procedures. The following graph edge types were created:

- Patient→has_diagnosis→Diagnosis
- Patient→has_procedure→Procedure

In total, our graph had 124k "*has_diagnosis*" edges and 49k "*has_procedure*" edges. The resulting graph was visualized using the *PyVis* library to ensure correctness.

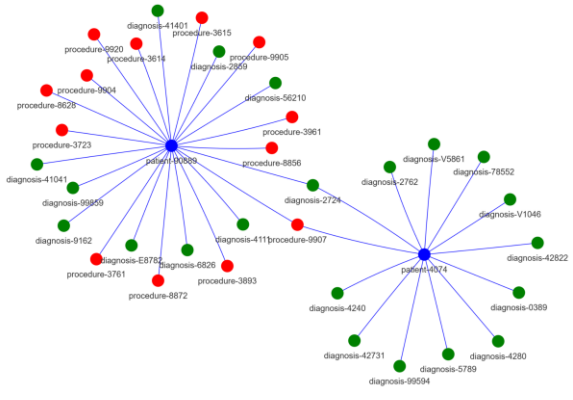

Figure 2: Subsection of graph with 2 patient nodes (blue), diagnoses nodes (green) and procedure nodes (red).

4. METHODOLOGY

4.1 Patient Similarity Data

Contrastive graph neural networks used for similarity detection require similarity data for validation. To train the contrastive learning model, we needed pairs of similar and dissimilar patients. We calculated Jaccard similarity between patient pairs based on the shared diagnoses and procedures:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Where A and B are the sets of diagnoses and procedures for two patients.

Sparse matrices were used to perform calculations for GPU efficiency, as we needed to analyze approximately 67.6M patient pairs. Patients with similarity scores above 0.3 were considered similar, giving us 64K positive pairs. We randomly samples an equal number of dissimilar pairs to create a balanced dataset of 128K pairs.

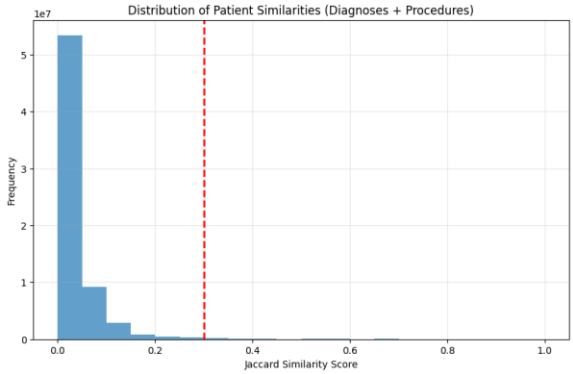

Figure 3: Patient pairwise Jaccard similarity distribution

4.2 Relational Graph Convolutional Network

We implemented an R-GCN using PyTorch Geometric. R-GCNs extend the message-passing framework of GCNs by introducing separate transformation matrices with different weights for each relation type, making them ideal for heterogeneous health graphs such as with MIMIC-III data.

The R-GCN architecture consisted of:

1. Heterogeneous graph convolutional layers that process different node types and edge relationships
2. Projection layer for patient nodes to generate final embeddings
3. Contrastive loss function to learn from similar and dissimilar patient pairs

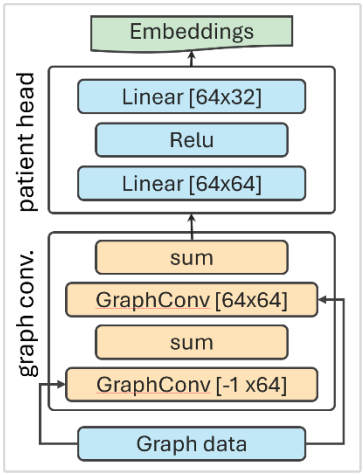

Figure 4: R-GCN architecture used with MIMIC-III data

The model was formulated as:

$$hi(l+1) = \sigma(\sum r \in R \sum j \in Nirci, r1Wr(l)hj(l) + W0(l)hi(l))$$

Where hi(l) is the representation of the node i at layer l, Nir is the set of neighbor indices of node i under relation r, ci,r is a normalization constant, and Wr(l) is the weight matrix for relation r at layer l.

4.3 Contrastive Learning

We trained the model using contrastive learning, to get similar patients closer together in the embedding space while moving dissimilar patients further apart. The contrastive loss function was defined as:

$$L(xi, xj, y) = y \cdot d(xi, xj)2 + (1-y) \cdot max(0, m - d(xi, xj))2$$

Where xi and xj are patient embeddings, y is 0 for similar pairs and 1 for dissimilar pairs, d is the Euclidean distance, and m is a margin hyperparameter.

The graph nodes and edge data were captured according to the specification in `HeteroData` required by `torch_geometric`. Each edge was processed separately according to the R-GCN architecture [1]. The R-GCN outputs patient embeddings which are stored and used for conducting patient similarity. Contrastive loss is used to calculate the pairwise distance between embeddings generated by the model. Validation is carried out using the 128k Jaccard dataset generated by patient pairwise comparison.

4.4 Model Training

We used the Adam optimizer with a learning rate of 0.005 and weight decay of 5e-4. Training was conducted for 100 epochs on GPUs. We observed rapid convergence with the loss decreasing substantially in the first 20 epochs and gradually after.

5. RESULTS

5.1 Model Convergence

The model shows consistent convergence in training, with loss going from 0.846 to 0.015 over 100 epochs. There was successful learning of patient representations that respect similarity relationships.

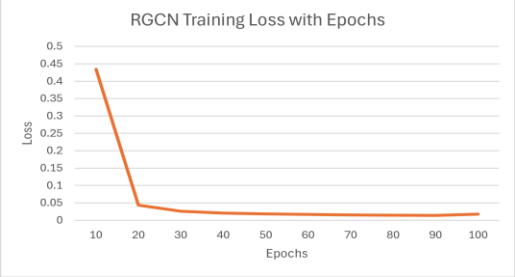

Figure 5: training loss across 100 epochs

5.2 Embedding Analysis

The patient embeddings were 32-dim vectors capturing both the direct attributes of patients and their relationships with diagnoses and procedures. During the test phase we used held out data for evaluating model performance as is the norm. The generated embeddings can now be used to perform similarity comparison with any patient input data to find similar patients.

5.3 Limitations

This study has some limitations. We processed only 25% of the MIMIC-III patient dataset due to computational constraints, potentially limiting the generalizability of the model. Our similarity metric was based solely on diagnoses and procedures, without incorporating other factors like lab values or medications. The clinical validation of the resulting patient similarities by healthcare professionals was beyond the scope of the initial study.

6. CONCLUSION

In this paper, we presented a framework for patient similarity detection using heterogeneous graph neural networks. R-GCN models are well suited to work with MIMIC-III based knowledge graphs and we attempted a 3-node structure with types of relationships. By representing patients, diagnoses and procedures as a connected graph and applying R-GCN with contrastive learning, we successfully generate patient embeddings that capture clinical similarities.

The opportunities for enhancing the graph structure along with leveraging the full data corpus hold promise. R-GCNs mark a significant advancement in patient similarity identification with real world benefits in patient care. With the exponential increase in health care data, the relational graphs can be enhanced with fitness health tracking data in addition to hospital data for a more cohesive patient picture.

Some opportunities for future work are:

- Additional development within the model to log accuracy and further tune the model which could not be addressed within the project timeline
- Leveraging a graph database (such as Neo4j or Cloud Spanner) to persist the graph for continuous updates with new patient data
- Creation of a RAG system that is frequently refreshed with a user interface for physicians
- Addition of edges and nodes to mimic the complete structures present in MIMIC-III
- Analysis and mining of physician notes for diagnosis, procedure and patient nodes.
- Aside from structural improvement opportunities, the model can be further updated and tuned on real world data to improve applicability.

7. REFERENCES

[1] Schlichtkrull, Kipf et al. 2017] Modeling Relational Data with Graph Convolutional Networks. published in the proceedings of the 15th Extended Semantic Web Conference (ESWC) in June 2018. https://arxiv.org/abs/1703.06103

[2] Wang, F., Sun, J., & Ebadollahi, S. (2012). Integrating distance metrics learned from multiple experts and its application in patient similarity assessment. Data Mining and Knowledge Discovery, 25(3), 415-449.

[3] Zhu, Z., Yin, C., Qian, B., Cheng, Y., Wei, J., & Wang, F. (2016). Measuring patient similarities via a deep architecture with medical concept embedding. In IEEE International Conference on Data Mining (ICDM) (pp. 749-758).

[4] Choi, E., Xu, Z., Li, Y., Dusenberry, M. W., Flores, G., Xue, Y., & Dai, A. M. (2020). Graph convolutional transformer: Learning the graphical structure of electronic health records. Journal of Biomedical Informatics, 101, 103383.

[5] Choi, E., Bahadori, M. T., Song, L., Stewart, W. F., & Sun, J. (2017). GRAM: Graph-based attention model for healthcare representation learning. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 787-795).

[6] Shang, J., Xiao, C., Ma, T., Li, H., & Sun, J. (2019). GAMENet: Graph augmented memory networks for recommending medication combination. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, pp. 1126-1133).

[7] Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., & Welling, M. (2018). Modeling relational data with graph convolutional networks. In European Semantic Web Conference (pp. 593-607).

[8] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In International Conference on Machine Learning (pp. 1597-1607).

[9] Li, Y., Rao, S., Solares, J. R. A., Hassaine, A., Ramakrishnan, R., Canoy, D., ... & Rahimi, K. (2020). BEHRT: Transformer for electronic health records. Scientific Reports, 10(1), 1-12.

[10] https://www.weforum.org/stories/2024/01/how-to-harness-health-data-to-improve-patient-outcomes-wef24/