EXPLAINABILITY FRAMEWORK FOR MOVIELENS GRAPH

Amal Nimmy Lal[1 [6987112]] and Aryman Deshwal[2 [4011205]]

Universität Paderborn, Warburger Str. 100, 33098 Paderborn
amalnlal@mail.uni-paderborn.de | arymand@mail.uni-paderborn.de
https://www.uni-paderborn.de/

Abstract. This project focuses on enhancing the explainability of Graph Neural Networks (GNNs) in the context of link prediction using a heterogeneous graph. GNNs have proven effective in various applications, including link prediction tasks. The objective of this study is to develop an interpretable GNN model that provides explanations for individual edges in the graph. The methodology involves creating a heterogeneous graph from the MovieLens graph, training a GNN model on the graph, and conducting validation to evaluate its performance. To achieve explainability, we employ the Captum explainer technique to provide insights into the prediction process of a single edge in the graph. Additionally, a surrogate model is employed to verify the predictions made by the Captum explainer. The results demonstrate the effectiveness of the proposed approach, with a validation AUC score of 0.9392 for the GNN model. By utilizing the Captum explainer and the surrogate model, we successfully provide explanations for the link predictions made by the GNN, contributing to the interpretability of the model. This research contributes to the field of explainable GNNs by applying the Captum explainer and surrogate model to enhance the interpretability of link prediction tasks. The findings highlight the potential for developing more transparent and explainable GNN models, facilitating better understanding and trust in the predictions made for link prediction applications.

Keywords: Graph Neural Networks· Explainability· Link prediction· Heterogeneous graph· MovieLens graph· Captum explainer· Surrogate model· Model validation

1 Introduction

This project aims to enhance the interpretability and explainability of Graph Neural Networks (GNNs) for link prediction tasks using the heterogeneous MovieLens dataset. GNNs have proven to be powerful models for capturing complex relationships in graph-structured data, making them suitable for link prediction. However, the lack of interpretability in GNNs raises concerns about their reliability, especially in domains where decisions based on predictions have significant consequences. Therefore, developing interpretable GNN models and providing explanations for link predictions is crucial to improve transparency and trust

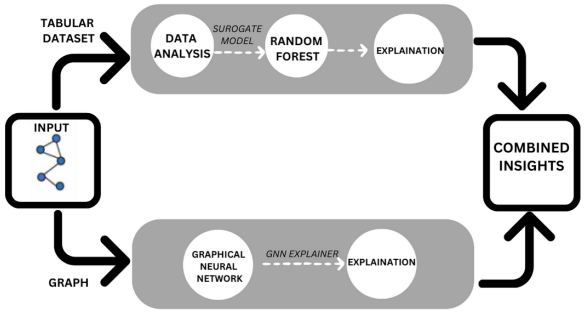

Fig. 1. Project Workflow - A flowchart depicting the project's steps. It demonstrates the extraction of a graph from the MovieLens dataset, training with GNN, using Captum explainer for explainability, also converting the graph to tabular form, training with Random Forest, and combining insights from both explanations..

in GNN-based models. The MovieLens dataset provides diverse information, including movie ratings, user profiles, genre details, and movie metadata. By leveraging the heterogeneous nature, we aim to capture the rich attributes and relationships within the movie domain, enabling more accurate link predictions.

To achieve our goals, we construct a heterogeneous graph representation from the MovieLens dataset and train a GNN model on this graph, as depicted in Figure 1. GNNs utilize node features and graph topology to capture intricate patterns and dependencies, enabling accurate link predictions To address the interpretability challenge, we employ the Captum explainer technique. Captum explainer provides explanations for individual link predictions, highlighting the influential nodes, edges, and subgraphs. Analyzing these explanations enhances our understanding of the factors contributing to successful link predictions in the MovieLens dataset.

Additionally, we convert the heterogeneous graph into a tabular dataframe representation and construct a surrogate model using Random Forest. The surrogate model offers an alternative perspective for explaining link predictions, utilizing feature importance techniques and examining decision pathways. Comparing the explanations from the Captum explainer and the surrogate model helps validate the reliability of the GNN's explanations and provides a comprehensive understanding of the link prediction process.

This project contributes to advancing the interpretability of link prediction tasks and provides valuable insights for decision-making in the movie domain. The obtained insights can be applied to improve recommendation systems, understand user preferences, and enhance movie content selection.

In the following sections, we present a detailed analysis of the MovieLens heterogeneous graph, including its characteristics and data analysis (Section 2). We then focus on training and evaluating the GNN model, discussing the data preprocessing steps, model architecture, and evaluation metrics (Section 3). Section 4 explores the interpretability of the GNN model using Captum explainer tech-

niques, shedding light on its decision-making process. Experimental results and performance analysis are presented in Section 5, including accuracy, precision, and other relevant metrics. Finally, Section 6 concludes the project, summarizing the findings, contributions, and potential future research directions.

2 Data Analysis

In our analysis of the MovieLens heterogeneous graph dataset, we investigated important aspects to gain insights into its composition and predictive factors. We began by examining the distribution of nodes, revealing 610 unique users and a matrix representation of movie features with 20 columns, encompassing genres like **Action, Adventure, Animation, Children, Comedy, Crime, Documentary, Drama, Sci-Fi, Thriller** (Fig 2).

Table 1. Key Features of the MovieLens Graph.

S.no	Metrics of MovieLens Graph	Values
1.	Number of nodes	10,334
2.	Number of edges	100,836
3.	Average Node Degree	19.52

Next, we performed exploratory data analysis (EDA) on the movie features to uncover meaningful relationships and correlations. This involved visualizing the distribution of features, exploring feature interactions, and conducting statistical analyses. The results are summarized in the correlation matrix (Fig 4) and the count of nodes and edges is presented in Table 1.

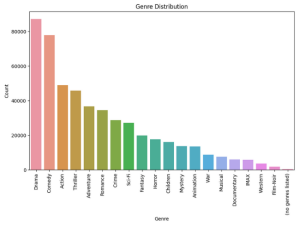

Fig.2 Genres Distribution

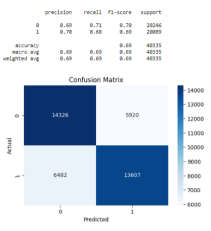

Fig.3 Confusion Matrix

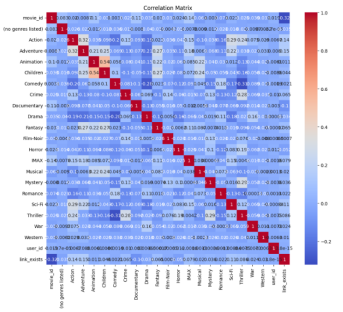

Fig.4 Correlation Matrix

To gain further insights, we converted the graph into a tabular format and analyzed the feature importance of the movie dataset using a Random Forest

model.This analysis allowed us to identify the most influential features in predicting movie ratings.When evaluating the performance of a binary classification Random Forest model on the dataset, we obtained an overall accuracy of 75 percent and an AUC score of 0.75 (see Figure 3 for the confusion matrix). These results indicate the model's ability to classify movie ratings accurately

The analysis provides a deeper understanding of the dataset and forms a basis for further exploration and modeling. These insights enable informed decisions for personalized recommendations and user-centric movie experiences. Overall, our analysis of the MovieLens heterogeneous graph dataset contributes valuable insights into the composition of nodes, connectivity between users and movies, and the features associated with movies.

3 GNN Training and Evaluation

In the GNN training and evaluation section, we implemented and assessed the performance of a Graph Neural Network (GNN) model for link prediction in a heterogeneous graph.

To begin with, we utilized the PyTorch and PyTorch Geometric libraries for building and training our GNN model. We imported the necessary modules, including pandas for data manipulation, torch for tensor computations, and torch_geometric for graph-related operations. We also checked the availability of GPU for accelerated computation and set the device accordingly.

Our dataset consisted of movie ratings, which we represented as a heterogeneous graph. We loaded the movie and ratings data into pandas data frames and performed data preprocessing steps such as one-hot encoding of movie genres and mapping of user and movie IDs to a continuous range of indices.

Next, we constructed the graph structure by creating edge indices that connected users to movies based on the ratings. We used the torch_geometric HeteroData object to store the graph data, including node features and edge indices. We also applied the necessary transforms to convert the graph into an undirected form.

For training the GNN model, we defined the architecture consisting of a GNN encoder and an edge decoder. The GNN encoder utilized SAGEConv layers to aggregate information from neighboring nodes and learn node representations. We employed the to_hetero function from torch_geometric to convert the encoder into a heterogeneous GNN model.

To handle the large-scale graph and enable efficient training, we used the LinkNeighborLoader from torch_geometric.loader module to create mini-batches of link neighborhoods. We specified the number of neighbors to sample for each node type and set the negative sampling ratio for generating negative training examples. We also defined the loss function as binary cross-entropy with logits and used the Adam optimizer for parameter updates.

During the training process, we iterated over the mini-batches and performed forward and backward passes through the model. We computed the loss based on the predicted edge labels and the ground truth labels. The optimizer was

then used to update the model parameters. We tracked the training loss for each epoch and evaluated the model's performance.

For evaluating the model, we created a separate validation loader using the LinkNeighborLoader. We calculated the area under the receiver operating characteristic curve (AUC) as a performance metric. We iterated over the validation loader, made predictions on the mini-batches, and compared them with the ground truth labels to compute the AUC score.

Overall, the GNN training and evaluation section of our mini project focused on constructing and training a GNN model for link prediction in a heterogeneous graph. We implemented the necessary components, evaluated the model's performance using AUC, and obtained valuable insights into the graph structure and relationships.

4 Explanation of GNN

4.1 Model Explainability with Captum explainer

In order to enhance the interpretability of our graph neural network (GNN) model and gain deeper insights into its decision-making process, we employed the Captum explainer technique. Captum explainer provides a valuable approach to understanding the behavior of GNNs by attributing importance scores to nodes and edges within the graph.

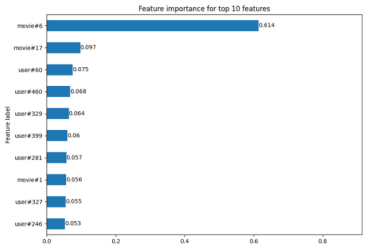

Fig.5 Feature Importance

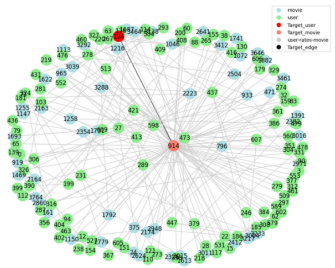

Fig.6 Induced Subgraph with Masked Zero Attributions

Furthermore, we examined the induced subgraph with masked zero attributions, which highlights the important nodes and edges contributing to link predictions. This visualization, presented in Figure 6, provides a clearer understanding of the key components driving the prediction process.

To implement Captum explainer, we leveraged the CaptumExplainer from the Captum library, which is specifically designed to work with Captum explainer

algorithms. By utilizing this powerful tool, we were able to generate explanations for our GNN model. The explainer operates directly on the trained GNN model, providing explanations at the level of individual edges in the graph.

To obtain these explanations, we specified the target edge indices for the explainer. By doing so, we were able to extract importance scores for the corresponding nodes and edges in the graph. These importance scores served as a measure of each feature's contribution to the model's predictions, enabling us to identify the most influential features and their relationships.

Visualizing the feature importance scores through a top-k feature importance plot which is depicted in Fig.5, we were able to highlight the key factors driving the decision-making process of our GNN model. This visualization provided valuable insights into the relationships and interactions between features within the graph, shedding light on the underlying mechanisms of our model's predictions

4.2 Comparison with Surrogate Model - Random Forest

To further validate the insights gained from the Captum explainer, we also converted our graphical dataset into a tabular format and employed a surrogate model, namely Random Forest. This approach allowed us to leverage the interpretability of traditional machine learning models and gain additional insights that could be compared with the explanations provided by the Captum explainer.

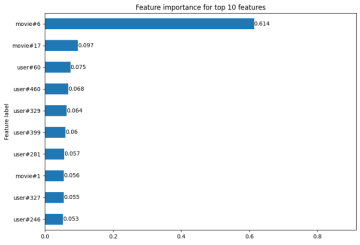

Fig.7 Feature Importance (Captum explainer)

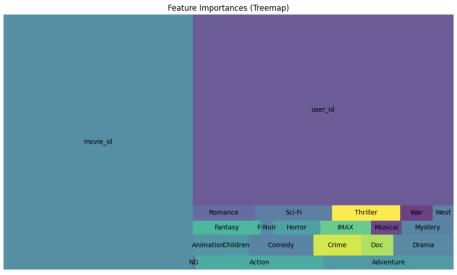

Fig.8 Feature Importance (Random Forest)

Using the same set of features employed in the GNN model, we trained a Random Forest model on the tabular dataset. The Random Forest model, known for its interpretability, provided us with feature importances, indicating the relative significance of each feature in predicting the target variable.

By comparing the feature importances obtained from the Random Forest model with the Captum explainer's importance scores as you can see in Fig.7 and Fig.8, we aimed to identify common patterns and discrepancies. This com-

parative analysis provided us with a more comprehensive understanding of the influential features and their relationships within the dataset.

The combination of the Captum explainer and the surrogate Random Forest model offered complementary perspectives on the dataset. It allowed us to gain robust insights into the factors driving the predictions of our GNN model and facilitated a more comprehensive understanding of its decision-making process.

This analysis significantly contributed to the interpretability and explainability of our GNN model, enhancing its transparency and trustworthiness in real-world applications. By incorporating both the Captum explainer and the surrogate model analysis, we achieved a deeper understanding of the complex relationships and decision-making process of our GNN model, making it more interpretable and facilitating valuable insights for domain experts and stakeholders.

5 Conclusion

In conclusion, our project developed an explainable Graph Neural Network (GNN) model for link prediction. By utilizing the Captum explainer and surrogate modeling with Random Forest, we gained insights into the model's decision-making process and validated its findings. This approach enhanced interpretability and provided meaningful explanations for the predictions. Our work demonstrates the importance of both predictive performance and understanding the underlying mechanisms of complex models. By combining GNNs, data transformation, and surrogate modeling, we contribute to the field of explainable AI and lay the foundation for future graph-based analytics research and applications.

6 Contributions of team members

Both team members, Amal and Aryman, played vital roles in the project's success. Amal excelled in meticulous data preprocessing and data analysis, ensuring the dataset's quality and engineering informative features. Her expertise greatly contributed to the model's performance. Meanwhile, Aryman focused on model development and training, demonstrating a deep understanding of GNNs and effectively utilizing the PyTorch and PyTorch Geometric libraries. His efforts led to a well-trained and high-performing model. Through the collaborative work of Amal and Aryman, combining data analysis, preprocessing, and model development, the team achieved remarkable results in link prediction and explainability. The synergy between Amal's data preprocessing skills and Aryman's model development expertise was instrumental in the project's overall success.

References

1. Zong,N. et al. (2017) Deep mining heterogeneous networks of biomedical linked data to predict novel drug–target associations. Bioinformatics, 33, 2337–2344.

2. GraphSAGE: Inductive Representation Learning on Large Graphs by William L. Hamilton et al..
3. Heterogeneous Graph Attention Network by Chuxu Zhang et al.
4. Link Prediction Based on Graph Neural Networks by Muhao Chen et al.
5. Muhan Zhang, Zhicheng Cui, Shali Jiang, and Yixin Chen. Beyond link prediction: Predicting hyperlinks in adjacency space. In AAAI, pages 4430–4437, 2018.
6. Graph Neural Networks with Generated Parameters for Link Prediction by Yanbang Wang et al.
7. Explainable Reasoning over Knowledge Graphs via Vector Symbolic Architectures by Tim Dettmers et al.
8. Explainable AI for Graphs: A Review by Alex Fabbri et al.
9. Explainability in Graph Neural Networks: A Taxonomic Survey by Chuanwei Ruan et al.
10. Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. Physica A: Statistical Mechanics and its Applications, 390(6):1150–1170, 2011.
11. David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. Journal of the American society for information science and technology, 58(7):1019–1031, 2007.
12. Schlichtkrull, M., Kipf, T. N., Bloem, P., Berg, R. V. D., Titov, I., Welling, M. (2018). Modeling Relational Data with Graph Convolutional Networks. In European Semantic Web Conference (ESWC) (pp. 593-607).
13. Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K. I., Jegelka, S. (2019). Representation learning on graphs with jumping knowledge networks. In International Conference on Machine Learning (ICML) (pp. 6465-6474).
14. Zhang, M., Chen, Y., Chen, X. (2018). Link prediction based on graph neural networks. In Advances in Neural Information Processing Systems (NeurIPS) (pp. 5165-5175).
15. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining (KDD) (pp. 974-983).
16. Shchur, O., Mumme, M., Bojchevski, A., Günnemann, S. (2018). Pitfalls of graph neural network evaluation. In International Conference on Learning Representations (ICLR).
17. Monti, F., Bronstein, M. M., Bresson, X. (2017). Geometric matrix completion with recurrent multi-graph neural networks. In Advances in Neural Information Processing Systems (NeurIPS) (pp. 3697-3707).
18. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W. L., Leskovec, J. (2019). GNNExplainer: Generating Explanations for Graph Neural Networks. In Advances in Neural Information Processing Systems (NeurIPS) (pp. 9248-9259).
19. Zhang, C., Chen, Z., Wang, J., Tang, J. (2020). Heterogeneous graph attention network. In Proceedings of the 29th ACM International Conference on Information Knowledge Management (CIKM) (pp. 2853-2860).
20. Wang, D., Cui, P., Zhu, W., Wang, H. (2019). Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery Data Mining (KDD) (pp. 1225-1234).