Polypharmacy Side-Effect Prediction with Graph Neural Networks

# Overview/Abstract:

Significant number of diseases directly or indirectly require combination of multiple drugs for treatment. Usage of multiple drugs can significantly exacerbate the chances of adverse drug reaction event (ADE), hindering the treatment of primary underlying condition. Hence, it is utmost import to study the ADE, develop methodologies to predict the ADE and take preventative actions to avoid ADE. In this study we present a graph neural network (GNN) model that processes three types of interactions namely, drug to protein (in humans) interactions, protein to protein interactions, and drug to drug interactions, to detect an ADE corresponding to a pair of drugs. We used publicly available dataset published in Zitnik et.al. [1] to train our model and achieved 99.06% of accuracy in detection of an ADE corresponding to a given pair of drugs.

Table of contents

[Overview/Abstract: 1](#_Toc91170331)

[Introduction: 3](#_Toc91170332)

[Problem: 4](#_Toc91170333)

[Data: 5](#_Toc91170334)

[Protein-to-protein interactions: 5](#_Toc91170335)

[Drug-to-drug interactions: 5](#_Toc91170336)

[Drug-to-protein interactions: 5](#_Toc91170337)

[Methodology: 6](#_Toc91170338)

[Step 1: Initial node representation 6](#_Toc91170339)

[Step 2: Classification model 7](#_Toc91170340)

[Results: 8](#_Toc91170341)

[Step 1: Initial node representation 8](#_Toc91170342)

[Step 2: Classification model 8](#_Toc91170343)

[Discussion/Observations: 9](#_Toc91170344)

[Conclusion: 9](#_Toc91170345)

[References 10](#_Toc91170346)

[Appendix 11](#_Toc91170347)

[Tables: 12](#_Toc91170348)

[Figures 13](#_Toc91170349)

# Introduction:

Many complex diseases need combination of drugs for effective treatment. Additional comorbid conditions can also lead to addition of more drugs in a treatment plan. Such usage of multiple drugs for treatment of a single or multiple conditions is known as polypharmacy. Even though polypharmacy strategies have been effective in treating underlying conditions, they are associated with higher risk of an adverse drug reaction event (ADE). Therefore, for successful polypharmaceutical treatment it is critical predict and prevent an ADE.

For example, people living with HIV (PLWH) are treated with one of antiretroviral therapy regimes. Every regime is a specific combination of multiple antiretroviral drugs and regimes are changed if PLWH develop resistance to a regime. Hence, the entire treatment trajectory of PLWH involves consumption of combination of multiple drugs. For successful antiretroviral therapy, continuity of drug consumption and protection of patient’s hampered immunity is important. Occurrence of an ADE can disturb both, treatment continuity and immunity of PLWH, impede the treatment progress and put patient’s health at risk. Similar to HIV, treatment of many physical conditions include combination of drugs for effective treatment. Such treatments can be made more robust by developing mechanisms to predict ADEs.

Combined effect of drugs in treatment plan on human body and effect of chemical interactions between multiple drugs needs consideration for prediction of ADE. Such fundamental biological and chemical dynamics can shed light on patterns behind an ADE and if considered can improve the performance of prediction model. In this study, we consider protein-to-protein interactions and drug-to-protein interactions datasets to capture the biological and chemical dynamics [1]. We develop a prediction model based on graph neural network (GNN) structure to effectively process the available information for the prediction of occurrence of an ADE for a given pair of drugs. We also extend our model to predict type of an ADE if a given pair of drugs results in ADE.

# Problem:

In this study we specifically study the problem of occurrence of an ADE for a given pair of chemicals. Moreover, we also extend our problem to detect the type of ADE if an ADE occurs. In other words, given following information,

|  |  |
| --- | --- |
|  | A graph of protein-to-protein interactions in humans where,is set of unique proteins[[1]](#footnote-1) in humans and is the set of connections between proteins. This graph is unweighted and undirected graph. |
|  | A graph of chemical-to-chemical interactions (reported events of ADEs) where, is the set of unique chemicals[[2]](#footnote-2) and is the set ADE corresponding to two chemicals. Let . This is an undirected graph with links having multiple features. |
|  | A graph of chemical-to-protein interactions (reported in previously published studies). This is a directed and unweighted graph. |

We develop a model such that,

and

We develop such that it satisfies the following objective,

Where, is the set of model parameters and is the objective function to be minimized. We express as follows,

Equation 1: Objective function

Where,

is a negative sample of a chemical, sampled from set with a distribution

# Data:

We use data published by Zitnik et.al. [1] to develop the above proposed model, . The dataset consists of three components that are, protein-to-protein interactions, drug-to-protein interactions, and drug-to-drug interactions. We will discuss each one in detail in following sections.

## Protein-to-protein interactions:

The in the Problem section is the protein-to-protein interactions data. The data is read as a graph with nodes () representing unique proteins and edges representing the experimentally validated physical reactions between a pair of proteins (). Authors of Zitnik et.al [1] have compiled this dataset from multiple previously published and publicly available datasets [2]–[5]. This graph is undirected and unweighted. There are over 19,000 proteins and over 700,000 physical interactions documented in the graph. The node-degree (number of proteins connected to a protein in the graph i.e., number of proteins that has some validated interaction/s with a protein in the graph) distribution follows the power law with a mean value of 75 and median of 33. Eight components exist in the graph, the largest component subsume 99.9% of all the nodes while the others are small, disconnected sets of nodes.

## Drug-to-drug interactions:

The in the Problem section is the chemical-to-chemical (or drug-to-drug) interactions data. Authors of Zitnik et.al. have combined information from SIDER, OFFSIDES and TWOSIDES datasets [1], [6], [7]. SIDER dataset is created by mining ADEs from drug label text, while OFFSIDES and TWOSIDES datasets are created from reports from patients, doctors, and drug-companies on ADEs. Therefore, all datasets used for creating contains information on ADEs only. Nodes () in the final combined data represents unique chemical/drug and the edges (or ) between a pair of drugs represent the category of the side effect. If two drugs are not connected by an edge, then we assume that there is no side effect corresponding to the pair. There are 645 unique chemicals and over 63,000 side-effects documented in the final dataset. All side-effects present in the dataset belong to 561 unique categories. The number of chemicals (number of nodes) in the graph exponentially reduce with increase in the number of side-effects associated with the chemical i.e., there are very few chemicals with more than 300 documented side-effects associated to it.

## Drug-to-protein interactions:

Authors of Zitnik et.al. used STITCH dataset to create the drug to protein interaction dataset [1], [8]. The STITCH dataset consists of experimentally verified interactions between chemicals (over 500,000 in number) and proteins (little less than 9,000). The final curated dataset; in our study; consists of 645 unique chemicals and over 19,000 proteins. We refer the proteins which are connected to a chemical as target proteins for that specific chemical. The mean and the median of the number of target proteins in are 75 and 9, respectively. We also probed in the dataset to analyze the common side effects pairs of chemicals and that majority of pairs of chemicals do not have any target proteins in common. The 25th, 50th and 75th percentiles of the number of common target proteins (calculated as ratio of intersection set size to the union set size) are 0%, 0%, 0.6%. In other words, most of the pairs of chemicals do not have any target protein in common.

# Methodology:

The problem presented in the problem section is a classification problem i.e., we aim to classify a given pair of drugs into categories, say, ‘side-effects’ and ‘no-side-effects’. In this section we lay out parts of the final model and steps taken to solve the classification problem. A schematic diagram of the method is shown in Figure 1.

## Step 1: Initial node representation

In the available data, there are only two types of nodes available, drug or protein. For drug nodes we have list of side-effects associated with it (in addition to ). Hence, we utilized this information to derive initial node embeddings for drug nodes. Previously published pretrained language model; SciBERT; was used to encode the text information of side-effects into real valued vectors [9]. We specifically calculated average of all side-effect embeddings for every drug.

Therefore,

Equation 2: Initial drug node embeddings

Where,

|  |  |
| --- | --- |
|  | Real valued vector representation of drug node i.e., of type |
|  | Side-effects associated with drug |
|  | SciBERT, a language model [9] |
|  | Text corresponding to side-effect of drug node |
|  | A side effect in |
|  | Drug node |

In addition to the side-effect information, we encode the structural information of the node in the embeddings with unsupervised version of the GraphSAGE algorithm [10]. GraphSAGE is a previously published inductive representation learning method, we brifly discuss the method but the detail discussion can be referred in the source paper [10]. Objective function of this unsupervised task maximizes the similarity between nodes which are present nearby in the graph and minimizes similarity between nodes which are placed far apart in the graph. We used random walk of length two[[3]](#footnote-3), to define ‘nearby’ nodes. The nodes which are not appearing in the random walk are considered as dissimilar nodes or ‘placed far apart’ nodes. In the initialization step of GraphSAGE we set the node embeddings to the matrix computed from equation 2.

Therefore,

|  |  |
| --- | --- |
|  | Real valued vector representation of drug node i.e., node of type |
|  | GraphSAGE model, an inductive node representation learning method [10] |
|  | Drug node |

Unlike drug nodes, protein nodes did not have additional text information than . To compute the initial node embeddings for protein nodes we again used unsupervised GraphSAGE method [10]. Similar to the , we set the random walk length equal to two for creating positive and negative samples.

Therefore,

Equation 3: Initial protein node embeddings

|  |  |
| --- | --- |
|  | Real valued vector representation of protein node i.e., of type |
|  | GraphSAGE model, an inductive node representation learning method [10] |
|  | Protein node |

## Step 2: Classification model

In the previous step we computed initial node embeddings for protein and drug nodes using graphs, respectively. We represent these embeddings as for all protein nodes in and for all drug nodes in . In this step we use both sets of embeddings to, compute drug node embeddings using i.e., , and solve the classification problem formulated in the problem section.

Depending upon the method used for computing we categorize our model as follows,

1. Model 1:

We simply use the average of target protein embeddings for every drug node,

Where, P is the set of target protein for every drug node in .

1. Model 2:

We use a graph convolution block to develop in supervised fashion. In other words, we tune the parameters of block while solving the binary classification problem.

The neural network classifier ultimately used for the binary classification task is kep same in both model types.

# Results:

## Step 1: Initial node representation

We explored only a learning rate and dropout while tuning the parameters of . From the results (Figure 2 and Figure 3) the objective function value started varying significantly after approximately 300 epochs. Continual of training after 500 epochs resulted in clear divergence of the model, resulting in high values of objective function. We also observed that the initial node embeddings used (derived from the side-effect lists) were clustered very close together hence increasing the difficulty for GraphSAGE implementation in finding underlying structural patterns. In the final embeddings from GraphSAGE we still observed a clear division of the initial cluster (marked in red). Such changes imparted to the initial embeddings are critical in downstream parts of the model.

For hyperparameter tuning of the model , we only perturbed depth, random walk length, learning rate at epoch 0 and dropout in a narrow range of values. Table 1 consists of the best combination of hyperparameter values we found during training of the model. Improvement in the objective function value; for both, the training and the validation set; during training is shown in the Figure 2. After training of the model , we conducted an additional validation exercise for the best results. We computed four clusters of protein nodes (in ) with K-means clustering [11]. The clusters were computed for two sets of embeddings, node embeddings randomly drawn from standard normal distribution and node embeddings from . Results of this validation exercise are shown in Figure 3. The sub-figure A is a plot of clusters of randomly sampled node embeddings[[4]](#footnote-4) and sub-figure B is a plot of clusters of embeddings computed from model. A clear distinction is seen between sub-figures, node embedding clusters formed with model are much more coherent than clusters formed with randomly sampled embeddings.

## Step 2: Classification model

Experimentation with the classification model part is very limited. We explored only the learning rate of the classifier part.

# Limitations:

Drug-to-drug interaction dataset is inherently incomplete. Negative sample sets created with such dataset can lead to incorrect and misleading results.

The training and testing datasets created for the classification problem are highly imbalanced. More work is needed to create justifiable datasets for training and testing. Although, this problem is also related to the limitations of the drug-to-drug interaction dataset.

# Conclusion:

We conclude that the current result from the developed model needs to be considered carefully. There are multiple sources e.g., highly imbalanced dataset, the mechanism used for negative sampling, number of negative samples and inherent incompleteness in the dataset, that can lead to a high value of accuracy in detecting presence of side effects for a given pair of nodes. It is critically important to address above mentioned issues and we propose lead the future work in that direction. Besides the misleadingly high classification accuracy, inherent issues in the drug-to-drug interaction dataset, we have achieved excellent results in computing protein node embeddings. The cluster visualization in Figure 5 emphasizes the advantage of graph neural network in encoding available information in real valued vectors. A similar observation, on smaller scale, is made from final drug node embeddings from drug-to-drug interaction dataset. Hence, both instances of unsupervised GraphSAGE use have provided a clear evidence success.

# References

[1] M. Zitnik, M. Agrawal, and J. Leskovec, “Modeling polypharmacy side effects with graph convolutional networks,” *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, Jul. 2018, doi: 10.1093/BIOINFORMATICS/BTY294.

[2] J. Menche *et al.*, “Uncovering disease-disease relationships through the incomplete human interactome,” *Science*, vol. 347, no. 6224, p. 1257601, Feb. 2015, doi: 10.1126/SCIENCE.1257601.

[3] A. Chatr-Aryamontri *et al.*, “The BioGRID interaction database: 2015 update,” *Nucleic Acids Res.*, vol. 43, no. Database issue, pp. D470–D478, Jan. 2015, doi: 10.1093/NAR/GKU1204.

[4] T. Rolland *et al.*, “A proteome-scale map of the human interactome network,” *Cell*, vol. 159, no. 5, pp. 1212–1226, Nov. 2014, doi: 10.1016/J.CELL.2014.10.050/ATTACHMENT/8F86630B-9EB8-429F-B5B6-C2B8D18B237F/MMC7.XLSX.

[5] D. Szklarczyk *et al.*, “The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible,” *Nucleic Acids Res.*, vol. 45, no. D1, pp. D362–D368, 2017, doi: 10.1093/NAR/GKW937.

[6] M. Kuhn, I. Letunic, L. J. Jensen, and P. Bork, “The SIDER database of drugs and side effects,” *Nucleic Acids Res.*, vol. 44, no. D1, pp. D1075–D1079, 2016, doi: 10.1093/NAR/GKV1075.

[7] N. P. Tatonetti, P. P. Ye, R. Daneshjou, and R. B. Altman, “Data-Driven Prediction of Drug Effects and Interactions,” *Sci. Transl. Med.*, vol. 4, no. 125, p. 125ra31, Mar. 2012, doi: 10.1126/SCITRANSLMED.3003377.

[8] D. Szklarczyk, A. Santos, C. Von Mering, L. J. Jensen, P. Bork, and M. Kuhn, “STITCH 5: augmenting protein–chemical interaction networks with tissue and affinity data,” *Nucleic Acids Res.*, vol. 44, no. Database issue, p. D380, 2016, doi: 10.1093/NAR/GKV1277.

[9] I. Beltagy, K. Lo, and A. Cohan, “SCIBERT: A Pretrained Language Model for Scientific Text,” pp. 3615–3620, Accessed: Dec. 22, 2021. [Online]. Available: https://github.com/google-research/.

[10] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive Representation Learning on Large Graphs.”

[11] D. Arthur and S. Vassilvitskii, “k-means++: The Advantages of Careful Seeding.”

[12] M. E. Tipping and C. M. Bishop, “Probabilistic Principal Component Analysis,” *J. R. Stat. Soc. Ser. B (Statistical Methodol.*, vol. 61, no. 3, pp. 611–622, Jan. 1999, doi: 10.1111/1467-9868.00196.

# Tables:

Table 1: Final set of GraphSAGE hyperparameters to get initial drug node embeddings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hyperparameter | Value | Layer specific value | | |
|  |  | Layer 0 | Layer 1 | Layer k |
| Random walk length | 2 | -- | -- | -- |
| Depth | 2 | -- | -- | -- |
| Number of negative samples for every node in training set | 5 | -- | -- | -- |
| Number of positive samples for every node in training set | 1 | -- | -- | -- |
| Learning rate at epoch 0 | 0.001 | -- | -- | -- |
| Learning rate scheduler |  | -- | -- | -- |
| Type | Linear | -- | -- | -- |
| Reduction step | 10 | -- | -- | -- |
| Reduction factor | 0.90 | -- | -- | -- |
| Layer size | -- | 256 | 256 | 256 |
| Dropout |  | 1% | 1% | 1% |
| Neighborhood sample size | -- | 5 | 5 | 5 |
| Aggregator | -- | Mean | Mean | Mean |

Table 2: Final set of GraphSAGE hyperparameters to get initial protein node embeddings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hyperparameter | Value | Layer specific value | | |
|  |  | Layer 0 | Layer 1 | Layer k |
| Random walk length | 2 | -- | -- | -- |
| Depth | 2 | -- | -- | -- |
| Number of negative samples for every node in training set | 10 | -- | -- | -- |
| Number of positive samples for every node in training set | 3 | -- | -- | -- |
| Learning rate at epoch 0 | 0.001 | -- | -- | -- |
| Learning rate scheduler |  | -- | -- | -- |
| Type | Linear | -- | -- | -- |
| Reduction step | 15 | -- | -- | -- |
| Reduction factor | 0.95 | -- | -- | -- |
| Layer size | -- | 768 | 256 | 256 |
| Dropout |  | 0% | 0% | 0% |
| Neighborhood sample size | -- | 5 | 5 | 5 |
| Aggregator | -- | Mean | Mean | Mean |

Table 3: Final results on classification problem

|  |  |
| --- | --- |
| Model | Accuracy on validation set |
| Model 1: using average protein node embeddings for z^cp |  |
| Hidden layers = 0 | 99.06% |
| Model 2: Using GNN to compute z^cp |  |
| Hidden layers = 0 | 99.06% |

# Figures

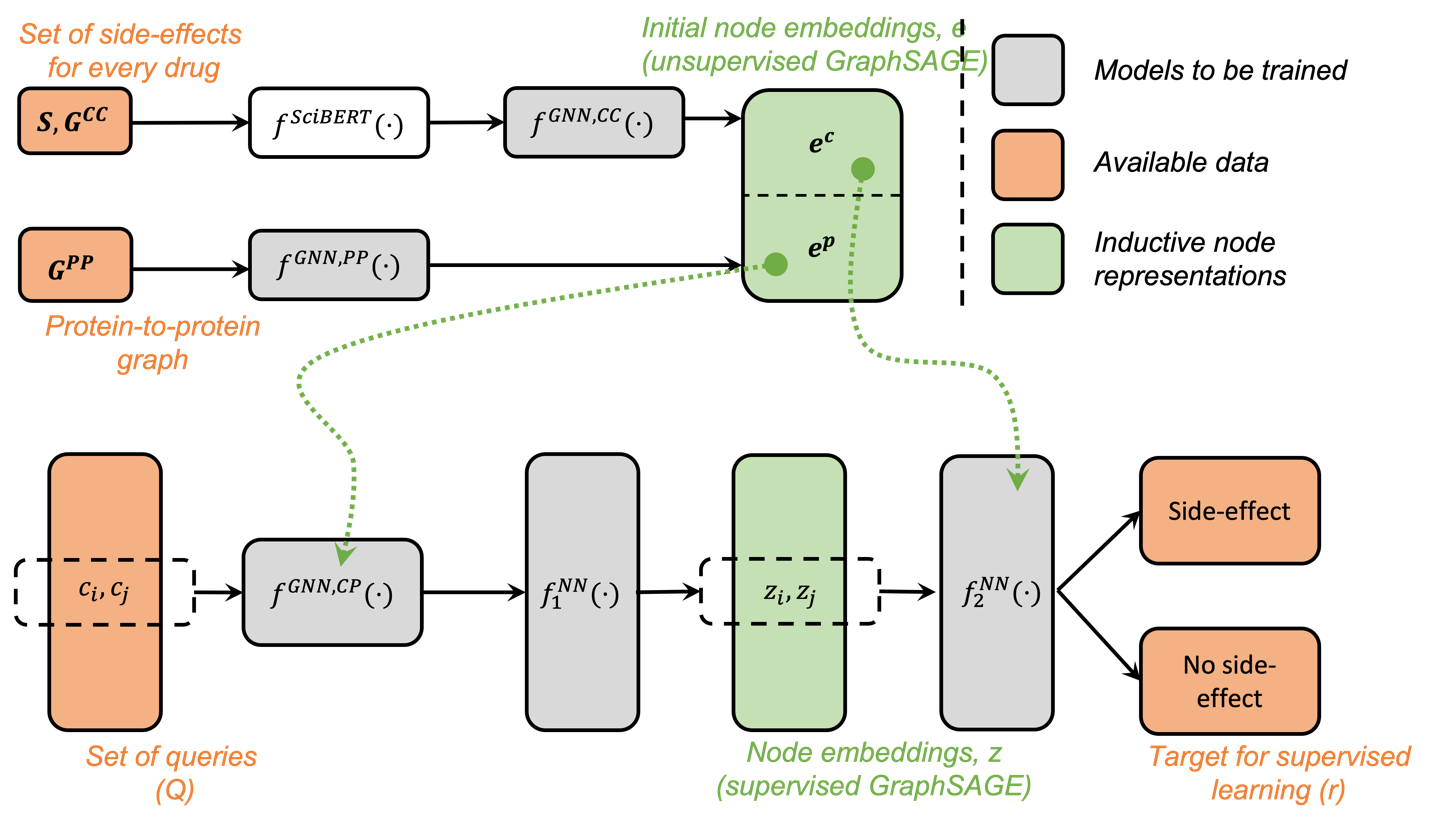


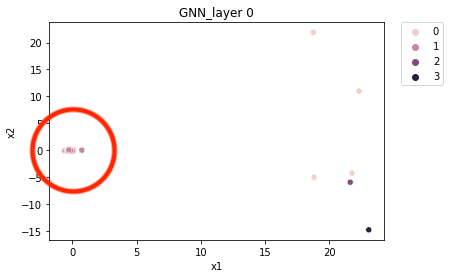
Figure 1: Model structure

Chart

Description automatically generated

Figure 2: Improvement in Cross-Entropy loss during GraphSAGE training for drug node embeddings

A



B

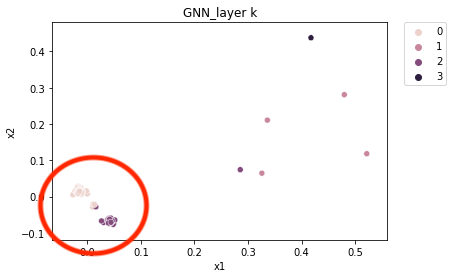


Figure 3: Visualization of drug node embeddings from Step 1 of the method

Chart

Description automatically generated

Figure 4: Improvement in Cross-Entropy loss during GraphSAGE training for protein node embeddings

A

Chart, scatter chart

Description automatically generated

B

Chart, scatter chart

Description automatically generated

Figure 5: Visualization of protein node embeddings from Step 1 of the method

Chart

Description automatically generated

Chart

Description automatically generated

Figure 6: Classification performance with Model 1

1. : Note that the set is not exhaustive [↑](#footnote-ref-1)
2. : Note that the set is not exhaustive [↑](#footnote-ref-2)
3. : Random walk length parameter is set to the value of two mainly because of the computational constraints of the machine we used for implementation. Estimated time for training model for one set of hyperparameters increased from approximately one to two hours to over nine hours after setting random walk length to a value greater than two. [↑](#footnote-ref-3)
4. : Node embedding dimension is reduced to two dimensions for plotting purpose. The horizontal and vertical axes in Figure 3 are abstract axes on which original node embeddings of dimensions 256 are mapped. We use Principal Component Analysis (PCA) for reducing the dimensions of the embeddings [12]. [↑](#footnote-ref-4)