Graph Learning Machine

Final Project

Writer:

Mohammad Rezaei Kalantary

Student Number:

401422087

Professor:

Dr Zahra Taheri

Shahid Beheshti University
Faculty of Mathematics
Spring 1401

Abstract

This report explores the concept of molecular representation learning through the use of quotient graphs. The study focuses on two datasets, Hiv and Lipophilicity, with a specific emphasis on graph classification and graph regression tasks. The code developed for this report implements the proposed methods and provides insights into the effectiveness of quotient graphs for molecular representation learning.

Introduction

This report explores the fascinating field of molecular representation learning, with a focus on the innovative approach of employing quotient graphs. Molecular representation learning has gained significant attention due to its ability to capture the intricate relationships and properties of molecules, enabling accurate predictions in various molecular property tasks.

Graph Neural Networks (GNNs) play a pivotal role in molecular representation learning. GNNs offer a powerful framework to model the structural and relational information inherent in molecules. By treating molecules as graphs, with atoms as nodes and chemical bonds as edges, GNNs can effectively capture the complex molecular structures and their associated features. This enables GNNs to learn informative representations that facilitate the prediction of molecular properties, such as biological activity, drug potency, and more.

Chemistry is a crucial domain in molecular representation learning. By leveraging chemical knowledge, such as bond types, functional groups, and molecular structures, we can enhance the understanding of molecular properties and behaviors. Chemistry provides the foundation for developing sophisticated molecular descriptors and features that can be integrated into GNN architectures, enriching the representation learning process. This synergy between chemistry and GNNs holds immense potential for advancing molecular discovery, drug design, and materials science.

1 Methodology

1. **Dataset Preparation**

   - The HIV dataset, obtained from the Drug Therapeutics Program (DTP) AIDS Antiviral Screen, was used for classification. The dataset contains over 40,000 compounds tested for their ability to inhibit HIV replication.
   - The Lipophilicity dataset, curated from the ChEMBL database, was used for regression. This dataset focuses on the lipophilicity values of drug molecules, an important feature affecting membrane permeability and solubility.

2. **Data Preprocessing**

   - For the HIV dataset, the data was preprocessed to extract the relevant features and labels. This involved selecting appropriate input features, such as molecular descriptors or fingerprints, and converting the target variable into binary labels (active/inactive).
   - For the Lipophilicity dataset, the data was preprocessed to extract the necessary features and target values. This involved handling missing data, normalizing features, and ensuring compatibility with the regression model.

3. **Graph Neural Network (GNN) Architecture**

   (a) Classification

   - The input features for the classification GNN model were molecular descriptors or fingerprints extracted from the HIV dataset.
   - The final output layer of the GNN model was modified to accommodate the classification task, producing binary labels (active/inactive) indicating the compounds' ability to inhibit HIV replication.
   - The cross-entropy loss function was used to optimize the model's parameters during training.
   - Evaluation metrics such as accuracy, precision, recall, and F1 score were used to assess the classification model's performance.

   (b) Regression

   - The input features for the regression GNN model were the features extracted from the Lipophilicity dataset.
   - The final output layer of the GNN model was designed to produce continuous output values representing the lipophilicity values of the drug molecules.
   - The mean squared error (MSE) loss function was employed to optimize the model's parameters during training, aiming to minimize the difference between the predicted and actual lipophilicity values.
   - Evaluation metric root mean squared error (RMSE) were used to assess the regression model's performance.

4. **Model Training and Evaluation**

   - The classification GNN model was trained using the preprocessed HIV dataset. The training process involved optimizing the model's parameters using an appropriate optimizer and minimizing the cross-entropy loss. The model was evaluated using various metrics such as accuracy, precision, recall, and F1 score. Cross-validation and hyperparameter tuning techniques were employed to ensure reliable performance.
   - The regression GNN model was trained using the preprocessed Lipophilicity dataset. The training process involved optimizing the model's parameters using an appropriate optimizer and minimizing the mean squared error (MSE) loss. The model was evaluated using metrics such as mean absolute error (MAE) and root mean squared error (RMSE).

5. **Model Performance Comparison**

6. The classification model's performance was compared against baseline models or existing approaches for HIV activity prediction. Statistical tests or other relevant evaluation techniques were applied to determine the significance of the model's performance.

7. The regression model's performance was compared against baseline models or existing approaches for lipophilicity prediction. Similar evaluation techniques were used to assess the model's effectiveness.

2 Results

2.1 Classification

we can observe the performance of different models, including GCN and GraphSage, with varying layer configurations. The following table summarizes the results obtained during the training and testing phases:

Model	Layers	Average Valid Score	Test Score
GCN	4	0.818	0.760
GCN	2	0.764	0.699
GraphSage	3	0.834	0.780
GraphSage	2	0.809	0.736

Table 1: Performance of Classification Models

In the GCN architecture, the model with 4 layers achieved an average validation score of 0.818 and a test score of 0.760. Meanwhile, the GCN model with 2 layers achieved a slightly lower average validation score of 0.764 and a test score of 0.699. Comparing the two GCN models, we can observe that increasing the number of layers improved the performance of the model.

For the GraphSage architecture, the model with 3 layers achieved the highest average validation score of 0.834 and a test score of 0.780. On the other hand, the GraphSage model with 2 layers achieved an average validation score of 0.809 and a test score of 0.736. This suggests that adding more layers in the GraphSage model also resulted in improved performance.

It is worth noting that the execution times for the models varied. The GCN models took longer to train, with the 4-layer GCN model having an execution time of 1087.717 seconds, while the 2-layer GCN model took 914.096 seconds. On the other hand, the GraphSage models had shorter execution times, with the 3-layer model taking 847.703 seconds and the 2-layer model taking 892.900 seconds.

Overall, the results indicate that both the GCN and GraphSage models were able to learn effectively from the dataset. The models with more layers generally achieved higher validation scores, indicating their ability to capture more complex patterns in the data. However, it is important to consider factors such as execution time and computational resources when selecting the optimal model architecture for a given task.

2.2 Regression

We can observe the performance of different models, including GCN and GraphSage, with varying layer configurations. The following table summarizes the results obtained during the training and testing phases:

Model	Layers	Average Valid Score	Test Score
GCN	3	1.293	1.370
GCN	2	1.332	1.447
GraphSage	2	1.331	1.440
GraphSage	3	1.243	1.294

Table 2: Performance of Regression Models

In the GCN architecture, the model with 3 layers achieved an average validation score of 1.293 and a test score of 1.370. The GCN model with 2 layers performed slightly worse, with an average validation score of 1.332 and a test score of 1.447. Comparing the two GCN models, we can observe that the model with more layers had a slightly better average validation score but a slightly worse test score.

For the GraphSage architecture, the model with 2 layers achieved an average validation score of 1.331 and a test score of 1.440. On the other hand, the GraphSage model with 3 layers achieved an average validation score of 1.243 and a test score of 1.294. In this case, the model with 3 layers had a lower average validation score but a similar test score compared to the model with 2 layers.

The execution times for the regression models were relatively shorter compared to the classification models. The GCN models took 176.062 seconds (3 layers) and 158.593 seconds (2 layers) for training. On the other hand, the GraphSage models took 189.603 seconds (3 layers) and 162.240 seconds (2 layers). These execution times reflect the efficiency of the models in learning from the regression dataset.

In summary, the regression models based on both GCN and GraphSage architectures were able to learn from the dataset and make predictions on the lipophilicity values of the drug molecules. The models with more layers generally achieved better validation scores, indicating their ability to capture more complex relationships between the molecular features and the lipophilicity values. However, it is important to consider factors such as execution time and computational resources when selecting the optimal model architecture for regression tasks.

3 Conclusion

this report, we explored the concept of molecular representation learning using quotient graphs. We focused on two datasets, the HIV dataset for classification and the Lipophilicity dataset for regression, to evaluate the effectiveness of quotient graphs in molecular property prediction tasks. The results obtained from our experiments provide valuable insights into the performance of different graph neural network (GNN) architectures, including GCN and GraphSage, with varying layer configurations.

Both the GCN and GraphSage models showed encouraging results for the classification task. The models with additional layers typically received higher validation scores, demonstrating their capacity to identify intricate data patterns. The 2-layer GCN model received a significantly lower average validation score of 0.764 than the 4-layer GCN model, which had a score of 0.818 overall. The three-layer GraphSage model performed better than the two-layer model, earning an average validation score of 0.834. These findings emphasise the value of building deeper GNN models for better classification performance. When choosing the best model architecture, it is crucial to take into account elements like execution time and processing resources.

The GCN and GraphSage models also showed impressive results in the regression task. The models with more layers generally achieved higher validation scores, indicating their ability to capture more intricate correlations between molecular characteristics and lipophilicity values. The 3-layer GCN model obtained an average validation score of 1.293, slightly outperforming the 2-layer GCN model, which had an average validation score of 1.332. Similarly, the 3-layer GraphSage model achieved an average validation score of 1.243, surpassing the 2-layer GraphSage model, which had an average validation score of 1.331. These findings confirm that improving regression performance can be accomplished by deepening models. However, as indicated in the earlier findings about classification

tasks, it is crucial to take other aspects into account when choosing the best model architecture for regression tasks, such as execution time and processing resources.

Overall, the results obtained from our experiments demonstrate the potential of quotient graphs and GNNs in molecular representation learning. These approaches show promise in accurately predicting molecular properties and can significantly impact various domains such as drug discovery, materials science, and chemical engineering. Further research and exploration of different GNN architectures, as well as the incorporation of additional molecular features, could provide even more powerful models for molecular property prediction tasks.