R-GCN: The R Could Stand for Random

Andrea Bonaiuti 1618825 - Riccardo Ceccarelli 1972735

20th July 2023

1 Method Description

Random Relational Graph Convolutional Networks (RRGCN) is a graph neural network model that combines the principles of Relational Graph Convolutional Networks (RGCN) with randomization techniques. The method implemented has the following steps:

- Data processing: The Entities dataset class, which gets the graph structure and node features from the supplied files, loads the graph data. Edge indices, edge types, node indices, and node properties all serve as representations of the graph.

- Graph Convolution Network (GCN) Encoder: The node embeddings are initialized by the GCN Encoder, who first sets them to zero before applying a normal distribution. The embeddings for nodes of the current type are then calculated by iterating over each node type.

- RGCN Convolutional Layer: Each layer calculates the average of neighbor nodes' representations weighted by an adjacency matrix and corresponds to a particular relation (edge type) in the graph. After that, the neighbor averages are added up to the output.

- RRGCN Model: this class initialize RGCN layers and the GCN encoder and returns as output node embeddings. An additional method here has been implemented, which is ppv, that return the proportion of positive values for each representation dimension. This method is necessary because the final output is obtained by concatenating the refined node representations with the proportion of positive values.

- Classification and Evaluation: Standard classification methods like gradient boosting or CatBoost can be used to train the RRGCN Model. Various classification metrics, such as accuracy, precision, recall, and F1-score, can be used to assess the model's performance.

2 Example Usage

1. Load and process the graph data using the Entities dataset class. Extract attrbiutes from the graph data
2. Build the GCNEncoder class;
3. Build the RGCNConvolution;
4. Define the RRGCN class;
5. Create an RRGCN model instance and specify the required inputs;
6. Pass the inputs through the model to obtain the output.

3 Results Summary

The MUTAG dataset, which aims to determine whether a cell is vulnerable to carcinogenesis, was used to train and assess the RRGCN model.Two different classifier has been used, GradientBoostingClassifier and the CatBoostClassifier:

- On the test set, the GradientBoostingClassifier had an accuracy of 0.75. With similar precision, recall, and F1-score for both classes, it displayed a nicely balanced level of performance. Class 0 scored slightly higher than class 1, though.
- The CatBoostClassifier, like the GradientBoostingClassifier, obtained an accuracy of 0.75 on the test set. In comparison to class 1, it demonstrated greater precision for class 0. Class 1 showed a higher recall, however, indicating that the classifier was more adept at spotting instances of class 1 than class 0 data.

-