Prediction of Missing Links

Hetvi Bagdai

April 25, 2024

1 Problem Statement

The objective of this problem is to predict missing links in the social network graph, i.e., identify potential connections that are not explicitly represented in the graph but may exist based on the observed interactions..

2 Algorithm Explanation

2.1 Step 1: Construct the Adjacency Matrix

Given a social network graph, construct an adjacency matrix where each cell represents the presence or absence of a directed edge between two individuals. Assign 1 to the matrix cell if there exists a directed edge between the corresponding individuals, and 0 otherwise.

2.2 Step 2: Predict Missing Links

Iterate through the entire adjacency matrix:

1. For each cell with a value of 0, indicating no directed edge between the corresponding individuals:
2. Remove the corresponding row and column from the adjacency matrix.
3. Perform linear regression or the method of least squares to find coefficients that approximately represent the removed row as a linear combination of the remaining rows.
4. Reconstruct the removed row using the coefficients obtained in the previous step. This reconstructed row represents the estimated interaction pattern of the isolated individual with the rest of the network.
5. Apply a threshold to the reconstructed row. If the value of any entry in the reconstructed row exceeds this threshold, consider it as a potential missing link.

6. If the value of any entry in the reconstructed row exceeds the threshold, add the corresponding missing link to the graph.

Repeat this process for each cell with a value of 0 in the adjacency matrix, considering each potential missing link individually.

2.3 Step 3: Output

The output of the algorithm is a list of predicted missing links in the social network graph.

2.4 Explanation of Code

1. **Reading the Data:**
   - It starts by reading a CSV file named 'modified_impression_network.csv' using pandas. This file presumably contains data representing a network where each row corresponds to a node and the columns represent its connections to other nodes.
2. **Creating a Directed Graph:**
   - It initializes an empty directed graph using NetworkX.
   - It iterates over the rows of the DataFrame `data`, assuming the first column represents the source node and the subsequent columns represent the target nodes.
   - For each row, it adds nodes and edges to the graph accordingly.
3. **Prediction of Missing Edges:**
   - The function `predict_missing_edges()` aims to predict missing edges in the graph.
   - It iterates over each cell in the adjacency matrix of the graph.
   - For each cell with a value of 0, indicating a missing edge, it calls `predict_missing_edge()` to predict the edge weight.
   - If the predicted weight is greater than 0.5, it adds the edge to the graph and stores it in the `missing_links` list.
4. **Edge Prediction Function:**
   - `predict_missing_edge()` function predicts the weight of a missing edge using a least squares approximation.
   - It removes the corresponding row and column from the adjacency matrix.
   - It performs a least square to approximate the missing edge weight.
   - It returns the predicted edge weight.

5. **Calculating PageRank Scores:**
    - It calculates PageRank scores for each node in the graph using NetworkX's `pagerank()` function.
6. **Printing Results:**
    - It prints the top leader node based on PageRank score.
    - It prints the number of missing links and the missing links themselves.

3 Method of least Squares

3.1 Least Squares

The method of least squares is a fundamental technique used in statistics and numerical analysis to estimate the parameters of a mathematical model by minimizing the sum of the squares of the differences between observed and predicted values.

1. **Define the Model:** Start by specifying a mathematical model that describes the relationship between the independent variable(s) (predictor variables) and the dependent variable (response variable). This model is typically a linear equation of the form:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon$$

   where y is the response variable, $x_1, x_2, \ldots, x_n$ are the predictor variables, $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients to be estimated, and ϵ represents the error term.

2. **Formulate the Objective Function:** The objective is to minimize the sum of the squared differences between the observed values and the values predicted by the model. This is expressed as the sum of squared residuals:

$$minimize \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

   where N is the number of observations, y_i are the observed values, and $\hat{y}_i$ are the predicted values.

3. **Compute the Residuals:** Calculate the residuals, which are the differences between the observed values and the predicted values for each data point:

$$residual_i = y_i - \hat{y}_i$$

4. **Minimize the Objective Function:** Find the values of the model parameters (coefficients) that minimize the sum of squared residuals. This is typically done using calculus, setting the derivative of the objective

function with respect to each parameter equal to zero and solving for the parameters. However, in numerical methods like the method of least squares, an exact solution might not always be feasible.

5. **Estimate Model Parameters:** Once the optimization process is complete, the estimated values of the model parameters (coefficients) are obtained.

3.2 Least Squares Method in NumPy

NumPy provides a convenient function called np.linalg.lstsq() for solving the least squares problem. This function computes the least squares solution to a linear matrix equation by finding the coefficients that minimize the sum of the squared differences between the observed values and the values predicted by the linear model.

4 Conclusion

In conclusion, it outlines an algorithm for predicting missing links in a social network graph using the method of least squares. The algorithm involves constructing an adjacency matrix from the graph, predicting missing edges by approximating their weights through least squares, and outputting a list of potential missing links.

Furthermore, the method of least squares is explained in detail, emphasizing its application in linear regression and its steps, such as defining the model, formulating the objective function, computing residuals, minimizing the objective function, optimization, and estimating model parameters.

Additionally, the the use of NumPy's ‘np.linalg.lstsq()‘ function is mentioned to solve the least squares problem efficiently.

Overall, the algorithm and the method of least squares provide a systematic approach to predict missing links in a social network graph, enabling insights into potential connections that are not explicitly represented in the graph structure.