Report Kaggle: Team Alcocoga

Pierrick ALLEGRE

Romain COLLEDANI

Theo COSTES

Antoine GALLIER

Abstract

We propose a solution for “Missing Link Prediction” in a citation network. Using Network Theory and using a directed graph theory approach and classifiers based on ensemble learning and/or deep learning techniques. Compared to the choice of technique, the “Feature Engineering” was the most significant contributor to the successful result.

# Problem and data definition:

From a citation network of research articles, edges have been removed from the nodes. The problematic consists of a directed network and the team is working on the following:

Given An unseen and varied directed citation network, use Feature engineering, Network science/ Graph theory and Machine Learning classification techniques to predict whether a citation link is present between two articles or not.

Hence, it is a binary classification problem requiring use of multiple areas from the field of data science.

To deal with this problem, we get 2 set of data:

* Training\_set.txt: This file contains three columns: the first node is the source (quoting article) and the second node is the target (quoted article). The third column shows if there is a link or not (0,1).
* Node\_information.csv: This file contains information for each of the nodes.We name the columns as the following:
  + id – unique id for each article
  + pub\_year – publishing year
  + Title
  + authors
  + journal\_name
  + abstract – a summary of the article

# Methodology

The below is followed in chronological order and also detailed in the Python Jupyter notebook.

1. Visualizing the network as a graph using DiGraph and NetworkX. We obtained those information : (Number of nodes : 27770) and (Number of edges : 335130)

2. Create a 10% subset of the data for computational purposes

3. Feature Extraction – for underlying structural, graphical and textual similarities between source and target.

4. Running multiple ensemble classifiers under a 3 fold Cross Validation to assess optimal classifer and best hyper parameters.

5. Run the optimal classifier (for exemple : Random Forest) on an unseen dataset.

# Feature Extraction

25 features are created for deriving information:

* Graph Related Features
  + source\_out\_centrality: the percentage of papers the article is quoting.
  + target\_in\_centrality: The percentage of article quoting out target article.
  + source\_centrality: the relative importance in terms of out links
  + target\_centrality: the relative importance in terms of in links
  + source\_evc: Eigenvector Centrality, is computed using adjacency matrix. It represents the source’s influence in the overall network. Hence, it is also referred to as Prestige Score. It further represents whether the source node network structure is structural like the target node.
  + target\_evc: Eigenvector Centrality computed for target node.
  + shortest\_path: number of nodes between source and target
  + target\_pagerank: the ranking of the node based on the structure of incoming links.
  + preferential\_attachment: the likelihood of a connection between two nodes, based on their existing connectivity.
  + source\_hub\_score: a hub will will cite many other nodes (out links from souce)
  + target\_authority\_score: an authority will be cited by other nodes (in links to target)
  + jacard: useful as it counts common neighbors but normalizes by the number of connections made by the two nodes.
  + common\_neighbors: closeness between two articles
  + out\_neighbors: for computing likelihood of 1 citing 2. If 1 cites none, it will not cite 2.
  + in\_neighbors: The likelihood of an article being cited can be gauged from this.
  + popularity: The number of neighbors of the nodes pointing towards the target.
* Meta Data Features
  + pub\_year\_difference: recent articles are likely to cite each other.
  + common\_authors: are likely to collaborate.
  + common\_journals: likely to have common topics and authors at a combined space.
* Textual Features:
  + title\_similarity: Sentence2Vec word embedding similarity used from Spacy.
  + abstract\_similarity: similar method as titles
  + testing\_dist\_absract: Cosine similarity between abstracts. A simple BoW approach would even consider common words but through TF- IDF “inverse” representation, we give weightage to less frequent words and compute cosine similarity.
  + training\_dist\_title: Cosine similarity between titles, same as above. Used as a measure of closeness.
  + common\_successors: the articles to which both source and target point towards.
  + common\_predecesors: the articles which pointed to both source and target.
  + overlap\_title: Common words within titles. This is useful as Stemming has been performed, making this method more insightful.
  + overlap\_abstract: Common words within abstracts.

For feature selection and overfit testing, multiple trial comparisons using validation data and Kaggle test submission were performed.