INF 554 - Introduction to Machine and Deep Learning
Data Challenge 2019

Link Prediction in the French Webgraph

1. Introduction

In this challenge, you will work in *teams of at most three students*. Your task will be to predict links between pages in a subgraph of the French webgraph. The webgraph is a directed graph G(V, E) whose vertices correspond to the pages of the French web, and a directed edge connects page U to page V if there exists a hyperlink on page *U* pointing to page *V*. **From the original subgraph, edges have been deleted at random**. Given a set of candidate edges, your job is to predict which ones appeared in the original subgraph. Each node is associated with a text file extracted from the html of the corresponding webpage. Your solution can be based on supervised or unsupervised techniques or on a combination of both. You should aim for the maximum mean F1 score.

This data challenge is hosted at Kaggle as an in-class competition. In order to access the competition you must have a Kaggle account. If you do not have an account you can create one for free. The URL to register for the competition and have access to all necessary material is the following:

https://www.kaggle.com/c/link-prediction-data-challenge-2019

2. File Description

training.txt - 453.796 labeled node pairs (1 if there is an edge between the two nodes, 0 else). One pair and label per row, as: source node ID, target node ID, and 1 or 0. The IDs match the text files in the node_information.zip file (see below).

testing.txt - 113.449 node pairs. The file contains one node pair per row, as: source node ID, target node ID. The label is not available (your task is to predict it).

node_information.zip - for each of the 33.226 web pages, there is a file containing some of the crawled text.

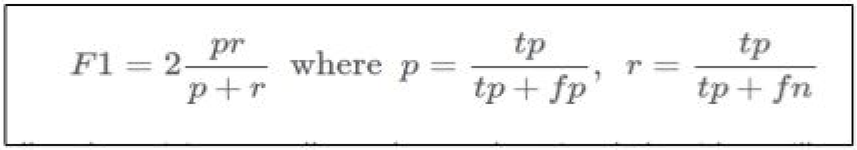
random_predictions.csv - a sample submission file in the correct format (the predictions have been generated by the random guessing baseline).

random_baseline.py - a Python script that given each pair of nodes randomly decides if there is an edge or not. The F1 score of this baseline is approximately 0.5.

graph_baseline.py - a Python script containing a simple baseline method that takes advantage of the structure of the graph. The method calculates the Jaccard Coefficient for each pair of nodes under question and then using, a threshold, decides if an edge should be added or not. The F1 score of this baseline is approximately 0.64 .

3. Evaluation Metric

For each node pair in the testing set, your model should predict whether there is an edge between the two nodes (1) or not (0). The test set contains 50% of true edges (the ones that have been removed from the original network) and 50% of synthetic, wrong edges (pairs of randomly selected nodes between which there was no edge). The evaluation metric for this competition is Mean F1-Score. The F1 score measures accuracy using precision and recall, while weighing both equally. Precision is the ratio of true positives (tp) to all predicted positives (tp + fp). Recall is the ratio of true positives to all actual positives (tp + fn). The F1 score is given by:



$$F1 = 2\frac{pr}{p+r} \quad \text{where } p = \frac{tp}{tp+fp}, \quad r = \frac{tp}{tp+fn}$$

4. Grading Scheme

Grading will be on 100 points total. Each team should deliver:

A submission on the competition Kaggle webpage. (30 points) will be allocated based on raw performance only, provided that the results are reproducible. That is, using only your code and the data provided on the competition page, the jury should be able to train your final model and use it to generate the predictions you submitted for scoring.

A zipped folder including:

1) A report (.pdf file, see details below), max 3 pages, excluding the cover page and references. Also, even though the 3 pages should be self-contained, you can use up to 3 extra pages of appendix (for extra explanations, algorithms, figures, tables, etc.). Please ensure that both your real name(s) and the name of your Kaggle team appear on the cover page.

2) A folder named "code" containing all the scripts needed to reproduce your submission.

The 3-page report should include the following sections (in that order):

- **Section 1: feature engineering (40 points).** Regardless of the performance achieved, the jury will reward the research efforts. Best submissions will capture both graph-theoretical and text information. You are expected to:
  1) explain the motivation and intuition behind each feature. How did you come up with the feature (e.g., research paper)? What is it intended to capture?
  2) rigorously report your experiments about the impact of various combinations of features on predictive performance, and, depending on the classifier, how you tackled the task of feature selection.

- **Section 2: model tuning and comparison (20 points).** Best submissions will:
  1) compare multiple classifiers (e.g., SVM, Random Forest, Boosting, logistic regression...),
  2) for each classifier, explain the procedure that was followed to tackle parameter tuning and prevent overfitting.

Report and code completeness, organization and readability will be worth **10 points**. Best submissions will (1) clearly deliver the solution, providing detailed explanations of each step, (2) provide clear, well organized and commented code, (3) refer to research papers.

Finally, note that the testing set has been randomly partitioned into public and private. Scores on the leaderboard are based on the public set, but final scores (based on which grading will be performed) will be computed on the private set. This removes any incentive for overfitting the testing set.

5. Submission Process

Submission files should be in **.csv format**, and contain two columns respectively named "id" and "predicted". The "id" column should contain row indexes (integers starting from zero). The “predicted" column should contain the predictions (0 or 1 for each node pair). Note that a sample submission file is available for download **(random_predictions.csv)**. You can use it to test that everything works fine.

The competition ends on **December, 30 midnight**. *Until then, you can submit your solution to Kaggle and get a score at most 5 times per day.* After the end of the competition, you need to prepare a compressed file containing your source code and your report explaining your solutions and discussing the scores you have achieved. This file must be uploaded to moodle before . Your final marks will be based on the score you obtained in Kaggle, the quality of your solution and your source code and the quality of your report. **There must be one final submission per team.**
Also, do not forget to include the name of your team and all team members’ real names **on the cover page of the report .**