Link Prediction in Citation Networks

Introduction

In this data challenge you will work in teams containing two or three persons. The problem you have to solve is related to *predicting links* in a *citation network*. In our case, the citation network is a graph $G(V, E)$, where nodes represent scientific papers and a link between nodes u and v denotes that one of the papers cites the other. Each node in the graph contains some properties such as: the paper abstract, the year of publication and the author names. From the original citation network some randomly selected links have been deleted. Your job is given a set of possible links to determine which of them have been deleted and thus they appear in the original network.

This data challenge is being 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://kaggle.com/join/cafoscaridatachallenge

File Description

There are five datasets provided. A short description of these files follows:

training_set.txt - 615,512 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 papers in the node_information.csv file (see below).

Sample

9510123 9502114 1

9707075 9604178 1

9312155 9506142 0

...

testing_set.txt - 32,648 node pairs. The file contains one node pair per row, as: source node ID, target node ID. Evidently, the label is not available (your job is to find the label for every pair).

Sample

9807076 9807139

109162 1182

9702187 9510135

...

node_information.csv - for each paper out of 27,770, contains the following information (1) unique ID, (2) publication year (between 1993 and 2003), (3) title, (4) authors, (5) name of journal (not available for all papers), and (6) abstract. Abstracts are already in lowercase, common English

stopwords have been removed, and punctuation marks have been removed except for intra-word dashes.

Sample

1001,2000,compactification geometry and duality,Paul S. Aspinwall,,these are notes based on lectures given at tasi99 we review the geometry of the moduli space of n 2 theories in four dimensions from the point of view of superstring compactification the cases of a type iia or type iib string compactified on a calabi-yau threefold and the heterotic string compactified on k3xt2 are each considered in detail we pay specific attention to the differences between n 2 theories and n 2 theories the moduli spaces of vector multiplets and the moduli spaces of hypermultiplets are reviewed in the case of hypermultiplets this review is limited by the poor state of our current understanding some peculiarities such as mixed instantons and the non-existence of a universal hypermultiplet are discussed

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

Sample

id,category
0,0
1,0
2,1
...

public_baselines.py - a Python script containing two baseline methods: a) random guessing (F1 score of approximately 0.5) and b) linear SVM with the following three features: (1) number of overlapping words in paper titles, (2) difference in publication years and (3) number of common authors (F1 score of approximately 0.66).

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 testing 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. 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:

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

This metric weights recall and precision equally.

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 as a .pdf file (see details below). Should be 3 pages in length, cover page excluded. 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. Although Python is preferred, you are free to use any other language like R or Matlab.

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

Section 1: feature engineering (40 points). Regardless of the performance achieved, intended to reward the research efforts, creativity and rigor put into feature engineering. Best submissions will capture both textual, graph-theoretical, and meta information available in the node_information.csv file. 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 should **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.

Submission Process

Submission files should be in **.csv format**, and contain two columns respectively named "id" and "category". The "id" column should contain row indexes (integers starting from zero). The "category" 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 end on **June 12, 2017**. Until then, you can submit your solution to Kaggle and get a score. Note that the number of submissions per day is limited. 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 achieved. This file must be uploaded to moodle before **June 15, 2017**. 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 submission per team**.

Bibliography

[1] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.* 58, 7 (May 2007), 1019-1031. DOI=http://dx.doi.org/10.1002/asi.v58:7

[2] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. 2016. A Survey of Link Prediction in Complex Networks. *ACM Comput. Surv.* 49, 4, Article 69 (December 2016), 33 pages. DOI: https://doi.org/10.1145/3012704