Technologies for Big Data Analytics 2018-2019

Project: *Link Prediction in Citation Networks*

1. Introduction

In this data challenge **you will work in teams of at most 2 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**. Given a set of possible links, your job is to determine the ones that appeared in the original network. **You will work in Spark using the Scala programming language**.

2. 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

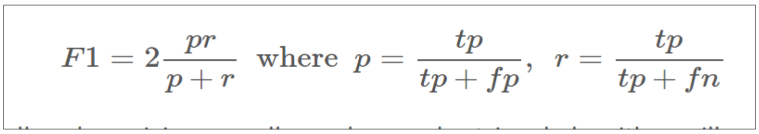
```
id,category
0,0
1,0
2,1
...
```

Cit-HepTh.txt – this is the complete ground truth network. You should use this file only to evaluate your solutions with respect to the accuracy. **<u>You should not take it into account to create your solutions</u>**.

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 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.

4. Problems to Solve

You have to provide solutions for the following problems:

P1. Given the network and a list of possible links, provide predictions if the links exist or not. You ma attack the problem by different perspectives. For example, you may solve the problem as a classification problem using a supervised approach or an unsupervised one.

P2. You are given only the papers and you are asked to discover all possible links that may appear in the network. In this version, you do not have an initial insight about which papers are linked. This problem is in general more difficult compared to P1, because the number of possible links is quadratic in the number of papers.

Note that both problems must be solved using Spark, which means that you need to provide distributed algorithms to solve them.

5. Submission Process

In addition to the source code you need to prepare a short report explaining your solutions and providing results based on accuracy and efficiency. You will upload the deliverables in moodle in the corresponding link that will be provided later. Also, note that you will present your work in class.

Good Luck!