Link Prediction in Citation Networks

Vasileios Barzokas
Department of Informatics
Aristotle University
Thessaloniki, Greece
vmparzok@csd.auth.gr

ABSTRACT

In this study we develop methods and try to evaluate models for predicting links in an academic citation network, by taking two different aspects into consideration: a) having an insight about the existing network and some of its links and trying to restore a portion of it that has been deliberately removed and b) having no information about the existing network and rely only on the information of the scientific papers in order to predict the structure of the whole network.

For the first aspect we used supervised binary classification and more specifically the method of **Logistic Regression** which had a very good result, with F1 score close to 86% against the testing set. For the second aspect we relied mainly on **Jaccard Similarity** of the MinHash LSH of each paper's abstract which had being vectorized using TF-IDF.

CCS CONCEPTS

• Information retrieval • Similarity measures • Clustering and classification

KEYWORDS

citation network, network analysis, binary classification, Jaccard coefficient, term frequency, inverse document frequency

Introduction

Our dataset contained 27,770 academic papers that were associated with the following information:

1. unique ID
2. publication year (between 1993 and 2003)
3. title
4. authors
5. name of journal
6. abstract

Some of the above information was not always available, like abstract or name of journal. We also had available a ground truth file of the whole network, used for evaluation purposes of our models. The implementation was under Apache Spark v2.4 using Scala language v2.11.8.

Methodology

1.1 Binary classification

For the first aspect of this study, where we followed the binary classification approach using Logistic Regression we chose to extract six features in total, which were:

1. number of common words in abstract
2. number of common words in title
3. number of common authors
4. difference in publication year
5. is same journal
6. TF-IDF vector of the abstract

Prior to applying the classification algorithm, we first preprocessed some of the text, mainly by tokenizing the title and abstract texts and then removing the common English stop words. This helped in removing words that would add some noise to the results of our training set.

1.1.1 Features

Below we explain why and how we chose each feature in more detail:

(1) number of common words in abstract. The most significant parts of a research are presented in the abstract of its academic publication and moreover it is common that methodologies and metrics keywords are referenced there. After tokenizing and cleaning the text of each abstract, we created an intersection of common words between each academic paper.

(2) number of common words in title. Same as the common words in abstract, and probably with higher significance since the amount of words in an academic paper's title are very limited so if there are common words between two papers it would be another good indication of a citation probability. After tokenizing and cleaning the text of each title, we created an intersection of common words between each academic paper.

*(3) number of common authors.*The authors who have written together more than one academic paper tend to cite each other's publications, since their work is related and quite a few times happens under the same academic institution. Here we just tokenized the authors of each paper and looked for common entries between each academic paper.

(4) difference in publication year. We assumed that the more the difference between two papers were published in terms of years, the less was the probability of them citing each other.

$$dif_{year} = published(to) - published(from)$$

(5) is same journal. Academic papers published on a journal tend to cite others that are published on the same, since the researchers working on them have common scientific interests. We considered as 1 if two papers are published on same journal or 0 otherwise.

(6) TF-IDF vectorizing. It is one of the most used methods for reflecting how important a word is to a document in a corpus. We set the number of Term-Frequency vector to 10000.

1.1.2 Evaluating results

The results of the Logistic Regression algorithm after running for 100 iterations, were quite good and more over we were able to achieve an F1 score of around 0.86, when using all the 6 features described previously.

1.1 Jaccard similarity

For the second aspect of this study, we followed the same approach as before for preprocessing the data and after calculating againg the TF-IDF vector for each academic paper's abstract and extracting the MinHash LSH value for it, we cross joined it with all the other papers in the network and calculated the Jaccard similarity value. After that we filtered the edges that had a similarity above 0.97.

TF is defined as:

$$1 + \log f_{t,d}$$

And IDF is defined as

$$\log\left(1 + \frac{N}{n_t}\right)$$

The above problem is highly intense and resource consuming in terms of computational power and our testing machine was failing to complete it, since Spark was running in a single machine and we could not benefit from its distributed processing powers.
The furthest path that we could reach was randomly selecting the 20% of the information that we had available for the 27,770 nodes which after comparing with the ground truth dataset prove to have a very low F1, which was expected. However we believe that if the same Scala code was used in a distributed environment, the computation would have been eventually completed and probably in a relatively low amount of time and the results would have been better in terms of F1 score.
Jaccard similarity is defined as:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Conclusion

When using pre existing knowledge about the network and trying to predict some parts of it using binary classification methods, we were able to reach a quite good result with F1 score close to 0.86. While when we didn't have such information available, the problem was becoming very resource demanding and we were not able to reach to a full result using a single computer.

REFERENCES

[1] Naoki S. and Yuya K. (2011). Link Prediction in Citation Networks, JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY, 63(1), 78–85.
[2] Wikipedia, MinHash, https://en.wikipedia.org/wiki/MinHash
[3] Wikipedia, TF-IDF, https://en.wikipedia.org/wiki/Tf%E2%80%93idf