Link Prediction in Citation Network

Eleftheria Trigeni
School of Science and Technology
International Hellenic University
Thessaloniki
etrigeni@ihu.edu.gr

Chrysoula Rempatsiou
School of Science and Technology
International Hellenic University
Thessaloniki
crempatsiou@ihu.edu.gr

ABSTRACT

Link Prediction consists a very common problem in network theory. In this project the citation network is a graph with scientific papers as nodes. An edge that points from a node to another indicates that there is a link between them. Some links have been removed from the original graph and the aim of this study is to determine them. In other words, it can be considered as a classification task that predicts if there is a link between two nodes or not. In order to come across with this task, useful metadata about papers are given and based on them some textual features have been constructed. Topological features related to graph have been taken into account too. Several models have been trained and evaluated using F1-score as a metric.

KEYWORDS

Link prediction, Citation Network, Scientific papers, Classification

THE DATASET

To take a better understand of the problem it is useful to explore the dataset. The training set is consisted from 615,512 annotated nodes. The label is 1 if there is a link between them and 0 if there isn't. Every paper has an ID. In the node_information csv file there are some metadata for every ID, that is for every paper. To be more specific for every paper the publication year, the title, the authors and the abstract are provided. Fore some of the papers the name of the journal that they have been published is given too. Our test set is comprised of 32,648 unlabelled node pairs and the goal is to discover these labels. Solution file contains the real labels for every pair of nodes and it is given in order to can evaluate our models and improve our results.

FEATURE ENGINEERING

Making good use of the papers information, the below features constructed.

Number of common authors: We have considered that if two papers have one or more common authors it is more possible to have a similar content and so to there is a link between them. The idea is that usually authors write about similar subjects. However, we found out that 52,41% of the papers that have no common authors, are linked. So we observe that this feature may not contribute in a high degree to our classification problem.

Date Difference: We assume that if two papers have been published closely, it is more possible to refer to the same or similar subject and so a link between them to exist.

Common words in title: According to the training dataset, 41% of the papers that have no common words in title, cite each other. We also discovered that of those papers that have at least one common word in the title, 82% are linked. So if two papers have common words in title, probably their topics are matched.

Common words in abstract: Similar, two papers with common words in the abstract, probably have similar content.

Same Journal: Mostly, journals publish articles and papers based on a particular content. So if two papers published in the same journal we assume that may a link between them is missing from the original graph. However the journal is known for only a sample of our data.

Cosine similarity between abstracts: Cosine similarity is a metric that inform as how close two documents are according to their content. A TF-IDF representation of the abstracts was required in order to calculate it. Cosine similarity takes values in the range [0,1]. The closer to 1, the grater the similarity between two papers. In our training dataset, all the papers that have cosine similarity more than 0.5 are matched. Therefore we assume that cosine similarity will have a determinant role to the link prediction problem.

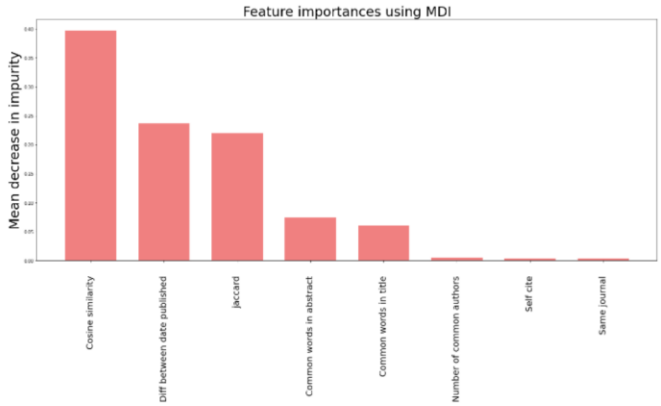
Jaccard similarity between abstracts: Jaccard similarity is an other measure to find how close two abstracts are.

Is self citation: The inspiration of the creation of this feature came up from paper [2]. Shibata, N., Kajikawa, Y., & Sakata, I. used this feature that indicates if two papers have written from at least one common author. In this case, the value is 1 and in the other case it takes the value 0. According to our dataset, 84% of the papers that have at least one common author are linked. However, it is worth to mention that only a small sample

(6%) of are papers in training dataset have at least one common author.

In order to discover which of these 8 features contribute at most to our problem, we trained a Random Forest to find the importance of each feature that is illustrated in Figure 1.

Figure 1: Importances of features constructed from the papers metadata



The first conclusion is that the most important feature related to metadata of the papers is the cosine similarity and our assumption is confirmed. Difference between the published date and jaccard similarity between the abstracts seems to be significant features too. Number of common authors and journal seems not to be determinant about our goal.

Next we took into account the graph and constructed some topology features. Other studies discovered that these features are more significant. In order to implement the below features, several papers have been studied and inspired us.

GRAPH FEATURES:

Firstly we considered our graph as undirected and the below features created:

Common Neighbors: The number of common neighbors in an undirected graph could be an indication of the similarity between two papers. After exploration data analysis we found out that only 5.85% of the papers that have no common neighbors have a link between them. Therefore we assume that nodes without common neighbours have small probability cite each other.

Jaccard coefficient: Jaccard coefficient is given by:

$$J_{(c1,c2)}=\frac{|T_{C1}\cap T_{C2}|}{|T_{C1}\cup T_{C2}|} \quad [3]$$

that is the value of intersection of neighbors between nodes C1,C2 divided by the value of the union of the neighbours of C1,C2. [2] Its values lie in range [0,1].

Adamic/Adar: (Frequency-weighted common neighbors) :A measure introduced by Adamic, L. A., & Adar, E. [5] that counts the number of common neighbors in a citation graph by weighting heavily the neighbours appear more rarely. To be more detailed, if paper A and B cite to paper C but paper C is «popular» there is not a strong evidence that A cite to B. In the other hand if C is not so popular means that not many other nodes cite it, then it is more possible that A cite to B.

Is same cluster: Firstly we seperated the papers into clusters with « community.best_partition(G) ». Then if two papers belong to the same cluster we assigned 1, either 0. This featured introduced by Shibata, N., Kajikawa, Y., & Sakata, I in paper [2].

Preferential attachment: A measure between nodes C1,C2 calculated with the below formula:

$$|T_{C1}|\text{x}|T_{C2}|$$

In other words, it expresses the product between the number of neighbours of C1 and the number of neighbours of C2. The more neighbours each node has, the most possible to exist a link between C1, C2.

Difference in betweenness centrality: Between centrality calculates how many times a node be located on the shortest path between other nodes. So a node with large between centrality is a sign that many paths pass from this node. According to the paper [2] :

Difference of between centrality = BC(paper2) – BC(paper1)

Then we considered the graph as directed and constructed the below features:

Page rank: A measure of the importance of a paper taking into account the quantity and quality of the links that point to it.

Using **HITS algorithm** we defined the **hub** and the **authority score** of each paper:

Hub score (i) = $\sum_{i=1}^{n} auth(i)$

Auth score (i) = $\sum_{i=1}^{n} hub(i)$

If a node has high hub score means that it points to many other nodes.

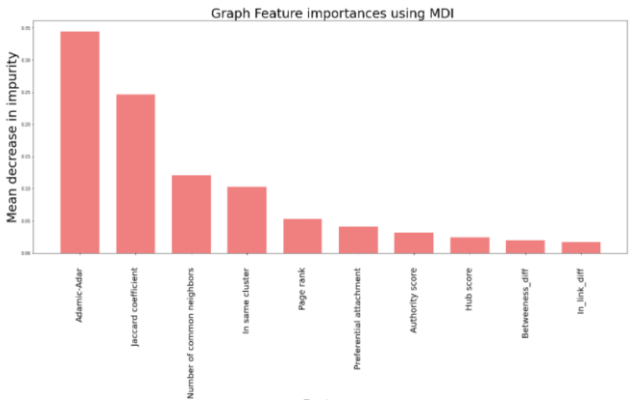
If a node has high authority score means that many points point to it. [10]

In-link difference: As Barabási et al. (2002) and Newman (2001) [11] studied the correlation between the probability of a node to be cited and the number of citations from this node. This led us to calculate the in-link difference similar to [2] :

In-link difference = # inlink(paper2) – # inlink(paper1)

Again, in order to find out which of the graph features are the most important we trained a Random Forest to find the importance of each feature.

Figure 2: Importances of graph-topological features

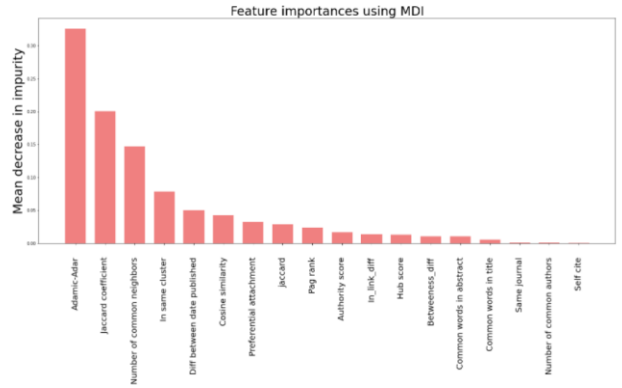


It seems that «Adamic-Adar» will have a pivotal role, followed by «Jaccard coefficient» and the «number of common neighbors» and «same cluster». We should mention that these 4 features emerged when we considered the graph as undirected.

Since we discovered the most important graph and metadata features, it is worthy to examine which features are the most crucial in combination. Therefore we trained a RandomForest Classifier in the whole dataset included all the features. The results of these algorithm would be optimal if we tuned the parameters. We tried Gridsearch provided by sklearn in order to achieve this. Unlikely, the training time was unlimited and that's why we carried on with the default parameters.

The below diagram represents the results. «Adamic-Adar» seems to be the winner. As we observe graph features seems to contribute more in the model. It is worth to mention that Random Forest chose published date in the fourth order and cosine similarity in the fifth order. When we examined only the features constructed from metadata, cosine similarity was more important than date. So the selected features that improve the model are these that give an additional value to the model and that is confirmed by [8] that ended up with the same conclusion.

Figure 3: Importances of total features



After this procedure we have an idea about the importance of each feature. However, the optimal number of features is still unknown. Therefore, we carried on with the feature selection process a little more and we used **Recursive Feature Elimination (RFE)** with cross-validation (4 folds) provided by sklearn. RFE suggested that the optimal subset of features excludes only self cite and same journal. Random Forest also indicated that these two features are not so important for our classification problem. In order to speed the algorithm we run RFE in Google Colab in order to make good use of GPU it provides. We could carry on with other feature selection methods in order to confirm and improve our results. Unfortunately these methods are time expensive.

Several models trained in order to achieve a high F1-score. We didn't take into account «self cite» and «same journal» since as it proved, they are not significant. Furthermore we excluded the « number of common authors». Even RFE included it to the optimal subset of features, we decided to remove it in order to reduce the complexity of models. Besides, the importance of this feature is low according to Random Forest and it is something that is confirmed from our exploratory data analysis. Despite that, it is worth to examine the impact in the performance of the model if we subtract other features, too. Because of Logistic Regression requires the less training time we experimented with this and we removed a feature at a time, starting from «common words in title» that was the next less important feature according to Random Forest. After this action performance of model declined but not significant. Then we excluded the other features one by one. The best F1-score achieved in the initial subset of features. So, we decided to carry on with these features.

Svm with linear and rbf kernel, Random Forest (RF), Logistic Regression (LR) and Light GBM classifier trained with aim to solve the problem.

Support Vector Machine (SVM) is a supervised machine learning technique that is used for classification and linear regression. It is supposed to be a successful method for text classification and linear regression. SVM is looking for the maximum object into a hyperplane that divides it into two classes. [18] A linear SVM model recognizes the smallest weight value and eliminates it. In every iteration one single feature is deleted. [19] In contrast to it, the RBF kernel is a non-linear method. More specifically, it estimates the similarity and the distance of two objects. [20] However, as we observe, the RBF is a highly-consuming technique.

Random Forest (RF) is a supervised method that is used for classification, too. It is a technique that creates decision trees. In particular, RF selects random observations from the sample, builds the tree and the final output is the average of the observations. [21]

Logistic Regression (LR) is a prediction method. It is appropriate for binary dependent variables. Obviously, LR is used to describe data and define the association between the binary variable and the independent one. [22]

LightGBM is a method that is based on decision trees. It helps to maximize the effectiveness of the model and decreases the GPU. LightGBM separates the leaves of the tree. Then, it fixes the leaf with the maximum delta loss, so it succeeds to have lower loss. [23]

MODELS COMPARISON

The below table presents the F1-score that achieved each model. Furthermore the training time is referred since we noticed that it is a crucial factor for model selection.

Table 1: Models performance comparison

	Svm (linear)	Svm (rbf)	RF	Light GBM	LR
F1-score	0.9634	0.9724	0.9737	0.9651	0.9657
Training time	27min 56s	1h 26min 14s	3min 47s	4.44 s	7.85 s

The figure above illustrates the performance of models in our classification task. All the models got very well since F1-score overcame 0.96. Besides, the differentiation in training time is significant. Random Forests seems to be the winner since it achieved the highest F1-score with a small training time. SVM with rbf kernel had a great performance but it is very time expensive. Logistic Regression and Light LGBM reached a high F1-score in only some seconds. SVM with linear kernel seems to be the looser since it has the smaller F1-score and it required also a half hour for training.

GRIDSEARCH

The role of Random Forest is crucial to our problem. It achieved a so high score without to tune its parameters. It is worth to experiment with it and using «GridsearchCV» provided by sklearn to tune its parameters and see if its performance will be improved even more. «GridsearchCV» provided by Sklearn is an algorithm that using cross-validation technique suggests which set of parameters commits to the best score. We applied this technique to find the optimal number of estimators (n_estimators) and the max depth(max_depth), that are the most crucial parameters. Gridsearch pointed out that a Random Forest Classifier with max_depth=8 and n_estimators=500 should be taken into account. Indeed, the best F1-score increased from 0.9737 to 0.9744 that is not a huge difference but still an improvement.

Ideally, we would tune the parameters of all the models. Unlikely this procedure is very slow so we decided to tune only the best model.

ENSEMBLE METHODS

An other try for improvement conducted. Since all our models had a good performance we wondered if the combination of their results translate to an enhancement. That's why we trained a voting classifier that predicts the class labels using majority vote. Actually voting classifier took into account the Svm with rbf kernel, Logistic Regression and the Random Forest Classifier with tuned parameters. We chose these 3 models with F1-score as criterion. A score 0.9732 achieved that is not an improvement. Random Forest is still the winner.

CONCLUSIONS- DISCUSSION

The role of feature engineering is crucial to the link prediction problem and the greatest effort was given to create the features. Graph features proved more important in comparison with the features based on metadata. Besides, the combination of both kind of features contributed to the highest F1-score. All the classifiers performed very well but Random Forest with tuned parameters seems to be the ideal classifier for our classification task.

The limitation in our work was the training time since we had a huge dataset and the models required hours to trained. This led us not to tune all the hyperparameters of the model. For future work this is suggested and expected to have even better results.

REFERENCES

[1]MacRoberts, M. H., & MacRoberts, B. R. (1989). Problems of citation analysis: A critical review. Journal of the American Society for information Science, 40(5), 342-349.

[2] Shibata, N., Kajikawa, Y., & Sakata, I. (2012). Link prediction in citation networks. Journal of the American society for information science and technology, 63(1), 78-85.

[3] Shibata, N., Kajikawa, Y., & Sakata, I. (2011). Measuring relatedness between communities in a citation network. Journal of the American Society for Information Science and Technology, 62(7), 1360-1369.Conference Short Name:WOODSTOCK'18

[4] Chen, H., Li, X., & Huang, Z. (2005, June). Link prediction approach to collaborative filtering. In Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05) (pp. 141-142). IEEE.

[5] Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the web. Social networks, 25(3), 211-230.

[6] 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

[7] 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

[8]https://github.com/raphm/linkprediction/blob/master/link-prediction-report.pdf

[9] Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. Journal of the American society for information science and technology, 58(7), 1019-1031.

[10] https://github.com/GuillaumeDufau/Missing-links-prediction-in-citationnetwork/blob/master/Report_Dufau_Murray_Zheng.pdf

[11] Acedo, F. J., Barroso, C., Casanueva, C., & Galán, J. L. (2006). Co-authorship in management and organizational studies: An empirical and network analysis. Journal of management studies, 43(5), 957-983.

[12] Al Hasan, M., Chaoji, V., Salem, S., & Zaki, M. (2006, April). Link prediction using supervised learning. In SDM06: workshop on link analysis, counter-terrorism and security (Vol. 30, pp. 798-805).

[13] https://cambridge-intelligence.com/keylines-faqs-social-network-analysis/

[14] Hasan, M., Chaoji, V., Salem, S., & Zaki, M. (2006, April). Link prediction using supervised learning. In SDM06: workshop on link analysis, counter-terrorism and security (Vol. 30, pp. 798-805).

[15]https://cambridge-intelligence.com/keylines-faqs-social-network-analysis/

[16]https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[17]https://scikitlearn.org/stable/modules/ensemble.html#voting-classifier

[18] Fernandez Jr, Proceso L., and Jan Miles Co. "Improving the vector auto regression technique for time-series link prediction by using support vector machine." (2016)

[19] Saptarshi Chatterjee, Debangshu Dey, Sugata Munshi, Chapter 4 – Feature selection and classification, Editor(s): Saptarshi Chatterjee, Debangshu Dey, Sugata Munshi, Recent Trends in Computer-Aided Diagnostic Systems for Skin Diseases, Academic Press, 2022, Pages 95-135, ISBN 9780323912112

[20] Liu, Zhiliang, et al. "An Analytical Approach to Fast Parameter Selection of Gaussian RBF Kernel for Support Vector Machine." J. Inf. Sci. Eng. 31.2 (2015): 691-710

[21] Xu, Baoxun, et al. "An Improved Random Forest Classifier for Text Categorization." J. Comput. 7.12 (2012): 2913-2920.

[22] LaValley, Michael P. "Logistic regression." Circulation 117.18 (2008): 2395-2399.

[23] Al Daoud, Essam. "Comparison between XGBoost, LightGBM and CatBoost using a home credit dataset." International Journal of Computer and Information Engineering 13.1 (2019): 6-10