Citation Recovery with Weighted Citation Network Links and Contextual Semantic Information

Jonathan C. Barker **Kartik Goyal** **Zhengzhong Liu**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{jcbarker, kartikgo, liu}@cs.cmu.edu

Abstract

In this work, we address the problem of missing citation prediction. Current approaches to this problem includes making prediction through the citation network structure and identifying them through semantic similarity. In this work, we attempt to incorporate both these information to help make better prediction. We spot the phenomenon that not all cited papers are equally important to the citing paper. Thus we adopt a supervised random walk method to learn the appropriate weights on these links using textual semantic information and cross-document topic models. We evaluate our weighted network by predicting missing citations for a paper, given the paper's current citations.

1 Introduction

Identifying relevant literature from electronic collection is becoming difficult with the rapid growth of the collection. This makes it very challenging for researchers to conduct efficient literature review. Automation of citation prediction task become necessary. Citation networks with citation links can provide us binary information about whether a link exists between the citing and the cited papers, but fail to encode information about the relative importance and relevance. As a result, it is difficult to draw profound conclusions from these networks about the relationships between different network nodes. We address this problem by assigning weights to network links according to their importance and relevance to the cited paper in order to mine deep information. Tasks like automatic paper recommendation and missing citation prediction can be made possible. Authors can use it to detect any papers they might not have cited in their initial draft, while journal reviewers can be aided in detecting any relevant works which are not cited by the submitted papers.

In practice, random walks techniques [1, 15, 17] have been employed to find node relevance in networks. These techniques generally work with a uniformly weighted network and utilize the graph structure. Our belief is that these techniques can be used more effectively if the network graph were to contain weighted links. The weights on the links could encode rich semantic information such as semantic correlation, citing patterns, shared authors and temporal proximity.

The major challenge in this approach is figuring out how various sources of information can be combined with the network structure. In doing so we must determine the relative importance of these sources/features with respect to the relationship score of two papers. When have finally synthesized this information to create our weighted network we aim to show that we can use it to produce more meaningful results for tasks such as missing citation prediction and recovery.

2 Related Work

There are also many unsupervised methods for link prediction. These methods mainly focus on digging the network structure. Liben-Nowell and Kleinnberg [9] experiments on several measures

on node proximity on a co-authorship network and conclude that Adamic/Adar method gives the best performance. Kashima et.al [7] propose a semi-supervised method using link propagation for prediction.
There are a few works using relational learning methods to approach this problem [16, 13]. Backstorm and Leskovec [1] perform supervised random walks on networks to learn the weights on the links between community members. The weighted network is used to predict future links between the members of the social network. The weights are learnt by minimizing an error function on the probability of predicting incorrect and correct future links. Lao and Cohen [8] use a path-constrained random walk to. Yu et.al [19] use a topic model to restrict the search space and learn the similarity using meta-path features. These method focus on sophisticate path formation but does not consider local deep semantic information.
Nallapati et al. [11] use topic models created with Latent Dirichlet Allocation(LDA) [2] to predict the existence of a citation link. Apart from generating words, they also generate the reference links from the topic model, thus creating a topical dependency between cited and citing papers. LDA can be also used for determining semantic correlations between two documents. Celikyilmaz et.al [4] studied the similarity measures based upon LDA on the Question Answering systems. These similarity measures used both, topic proportions and importance of similar words for estimating similarity.

3 Method

Problem formulation: A major challenge of this work is how to learn the relevant weights of features of according, we aim to learn a function that can properly assign weights based on the dataset. Here we adopt a method similar to [1] which optimize a function that can assign higher random walk probability to real missing nodes versus other nodes. Consider we are given a uni-directed graph $G(V, E)$(citation links only come from one paper to another paper.), a node s representing the paper we are interested in. Node s links to several nodes in the network. The task is to identify the other potentially connectable links $D = d_1, ..., d_m$ and avoid linking to the $L = l_1, ..., l_n$ where there should not be a link. For each edge $e_{uv} \in E$ we denote the set of pairwise features that associated with node (u, v) as ψ_{uv}. By creating multiple pairs of d and l, our objective is to maximize the possibility p_d of recovering nodes in D and minimize the possibility p_l of linking to nodes in L.
Here we use a random walk method to calculate these possibilities. In order to direct the random walk to our objective using the features, we assign each edge $e \in E$ a weight using a parameterized function $f_w(\psi)$ which combine the feature ψ with the parameters w. So the w is exactly the parameter vector that we are going to learn. Similar to [1], our target is to minimize the function:

$$F(w) = L(w) + h(p_l - p_d) \tag{1}$$

with respect to w with certain regularization function L on w and the loss function h. By taking the partial derivative of $F(w)$ we have:

$$\frac{\partial F(w)}{\partial w} = \frac{\partial L(w)}{\partial w} + \sum_{l,d} \frac{\partial (p_l - p_d)}{\partial w} \tag{2}$$

The relation between the possibility of linking to node p_u can be related with w using the transition matrix Q. The basic transition matrix $Q^{(0)}$ is given by:

$$Q_{uv} = \begin{cases} \frac{f_w(\psi_{uv})}{\sum_t f_w(\psi_{ut})} & (u, v) \in E \\ 0 & \text{otherwise} \end{cases}$$

In our problem we are interested a more general Q which consider the preference vector, which is given by:

$$Q = (1 - \alpha)Q^{(0)} + \alpha \mathbf{1}(u \in P) \tag{3}$$

where P is the preference vector.

The random walk score is given by the stationary vector p where:

$$p^T = p^T Q \tag{4}$$

Eq. 4 and the transition matrix Q give us the connection between w and p. Then we can use the gradient descent method to optimize $F(w)$. Detailed derivation can be found in [1].

Feature selection: Pairwise features between a cited paper and a citing paper can be gathered from citation patterns, semantic correlation and the network metadata:

1. Number of citances for the cited paper. ("citance" refers to a citing sentence in a paper)
2. Co-citance score. This is defined as the number of times another paper is cited in the same citance as the target cited paper.
3. The number of common papers which both citing and cited papers cite.
4. Topical similarity between the citing and cited paper, calculated using topic proportions from Latent Dirichlet Allocation(LDA).
5. Number of common authors.
6. The difference between time of publication, measured in years.

Features 3, 5 and 6 are meta-data related to the citation network. To determine 1 and 2 we used regular expressions to extract citances and match them with the cited papers. Mallet toolkit [10] was used for running LDA, which uses gibbs sampling for parameter inference, on the raw text of all the papers. The topic proportion vectors were created and were used to calculate cosine similarity between documents.

4 Experiment

4.1 Experiment setup

We are currently conducting the experiments on the ACL Anthology Network (AAN) [6, 14] which contains the citation network of ACL paper collections and the relevant metadata from year 1965 to 2011 with over 15,000 papers. We choose to use papers from year 2010 data as the training set with papers before 2010 as the candidate set. Similarly, papers from year 2011 data (about 760 papers) are used as the testing set and all other papers are treated as the candidate set (about 14,600 papers). To generate the D and L set, we randomly remove links from each testing paper with 25% possibility. The removed links will be treated as the D set and all other links will non-existing links will be treated as L set.
As pointed out by [11], normal metrics such as precision and recall are not suitable for this task because we do not expect a paper to cite all relevant papers. We adopt their metric RKL (short for RanK Last), which represent the rank of the last citation we recovered. On the other hand, we are also interested when we can get the first correct prediction, so we introduce another metric RKF (short for Rank First).

4.2 General considerations on model:

In the experiments, there are several decisions that we are going to evaluate. (A) Selection of preference vector; (B) Selection of restart possibility α; (C) Selection of loss function h; (D) Selection of regularization function L ($L1$ or $L2$ norm). (E) Experiments on feature selection. By midterm we conduct some simple experiments and the rest will be needed to further evaluated.
(A) Basic model and preference vector. The purpose of our method is to mine possible citation papers. Beside from using weights to direct the random walk. The selection of preference vector also plays an important role. Random Walk with Restart (RWR) have proved to be useful in looking for similar nodes in neighbors [1, 17]. However, in a paper the author would not only cite papers that have links with each other. We make an hypothesis that author of a paper should be more likely to cite papers in multiple topic groups corresponding to the topics discussed by the paper. Thus it make sense to set the preference vector based on the topical similarity with the target document. To verify the hypothesis, we conduct some simple experiments. We consider different the combination of random walk and LDA model using Cosine similarity over topic distribution similarity. In addition, we consider different handling over the dangling nodes. In "Strongly preference" mode, we patches all dangling nodes adding transitions following our preference vector but in "Weakly preference" mode, we patches them with a uniform transition towards all other nodes. We use the WebGraph[3]

Table 1: Experiment on Basic Model and Preference Vector

Method	Average RKF	Average RKL
RWR (weakly preference)	1084.33	4523.92
RWR (strongly preference)	1692.63	6123.96
LDA similarity prediction	392.57	2460.04
RWR with LDA preference (weakly preference)	**151.04**	**809.23**
RWR with LDA preference (strongly preference)	162.83	850.92

package to do the experiments. Different combinations give us 5 different experiments.
The preference vector P is stochastic, we assign to the target node s a preference $\beta(0 < \beta < 1)$. Then we assign the rest $1 - \beta$ weights to all the other nodes in proportion to their similarity with s. In this experiment, β is set as 0.8 and α is 0.2 .
The different experiments results are shown in table 1. The general trend shows that "weakly preference" performs better than "strongly preference". On the other hand, we can see that simple LDA similarity outperforms the basic RWR, this is consistent with our hypothesis that author would prefer to cite papers in multiple topical groups. However, the combination of LDA and RWR gives the best performance, which is consistent with our hypothesis. It is also a positive sign that our method of combining semantic and network information is effective.
(B) Selection of restart possibility. By far we have only tested the α value on the basic random walk method. In Fig. 1 we can see that the performance on different $alpha$ are similar except when close to the two ends (0 or 1). Further experiments are needed to see whether similar phenomenon happens.
(C) Selection of loss function. We need to consider different loss functions on other

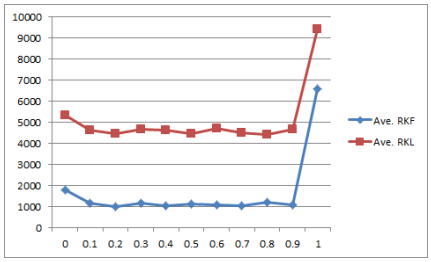

Figure 1: Behavior of different alpha value in basic random walk

(D) Selection of regularization function. We need to test how different regularization methods affect the performance.
(E) Experiments on feature selection We need to test the effectiveness of some features.

4.3 Observations and other considerations

We observe that self-citation also have impact on the performance because the one would likely to cite his own paper if they are on the similar topic. We would like to test our algorithm on a citation network without self-citations.

4.4 Evaluation result on data set

We need to present our final results on data set and compare our method with other methods.

5 Conclusion

References

[1] Lars Backstrom and Jure Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 635–644, New York, NY, USA, 2011. ACM.

[2] David M. Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[3] Paolo Boldi and Sebastiano Vigna. The WebGraph Framework I : Compression Techniques. *Proceedings of the 13th International World Wide Web Conference*, pages 595–602, 2004.

[4] Asli Celikyilmaz, Dilek Hakkani-Tur, and Gokhan Tur. Lda based similarity modeling for question answering. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, SS '10, pages 1–9, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[5] Aaron Clauset, Cristopher Moore, and M E J Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, May 2008.

[6] Bryan Gibson Pradeep Muthukrishnan Dragomir R. Radev, Mark Thomas Joseph. A Bibliometric and Network Analysis of the field of Computational Linguistics. *Journal of the American Society for Information Science and Technology*, 2009.

[7] Hisashi Kashima, T Kato, and Y Yamanishi. Link propagation: A fast semi-supervised learning algorithm for link prediction. In *Proceedings of the 2009 SIAM International conference on data mining*, 2009.

[8] Ni Lao and William W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67, July 2010.

[9] David Liben-Nowell and J Kleinberg. The link prediction problem for social networks. In *Proceeding CIKM '03 Proceedings of the twelfth international conference on Information and knowledge management*, number November 2003, pages 556–559, 2007.

[10] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit, 2002.

[11] Ramesh Nallapati, Amr Ahmed, Eric P. Xing, and William Cohen. Joint latent topic models for text and citations. In *KDD*, pages 542–550, 2008.

[12] Ramesh M. Nallapati, Amr Ahmed, Eric P. Xing, and William W. Cohen. Joint latent topic models for text and citations. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '08*, page 542, New York, New York, USA, 2008. ACM Press.

[13] Alexandrin Popescul and LH Ungar. Statistical relational learning for link prediction. In *In Workshop on Learning Statistical Models from Relational Data at the International Joint Conference on Articial Intelligence (2003)*, 2003.

[14] Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. The ACL anthology network corpus. In *Proceedings, ACL Workshop on Natural Language Processing and Information Retrieval for Digital Libraries*, Singapore, 2009.

[15] Purnamrita Sarkar and Andrew W. Moore. Fast dynamic reranking in large graphs. In *Proceedings of the 18th international conference on World Wide Web - WWW '09*, page 31, New York, New York, USA, 2009. ACM Press.

[16] Ben Taskar, MF Wong, P Abbeel, and D Koller. Link prediction in relational data. In *Seventeenth Annual Conference on Neural Information Processing Systems - NIPS'03*, 2003.

[17] Hanghang Tong, Christos Faloutsos, and JY Pan. Fast random walk with restart and its applications. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 613–622. Ieee, December 2006.

[18] Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *KDD*, 2009.

[19] Xiao Yu, Quanquan Gu, Mianwei Zhou, and Jiawei Han. Citation prediction in heterogeneous bibliographic networks. In *Proccedings of the SIAM International Conference on Data Mining (SDM12)*, 2012.

6 Appendix

6.1 Plan of activities in proposal

Software to write:
We must write feature extractors, modify existing machine learning algorithm implementations for feature integration and parameter estimation, and create a testing framework.

Papers to read:
Ramesh Nallapati, Amr Ahmed, Eric P. Xing, and William W. Cohen, Joint Latent Topic Models for Text and Citations. Proceedings of The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (KDD 2008)

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research, 3:9931022, 2003.

Midway report milestone:
By this time we will have a system which can rank the citations within a paper, produce a weighted network, and predict citations using this weighted network. Using this system we can then evaluate performance by comparing predictions with given citations.

Team responsibilities:
Since our task involves a lot of feature engineering we will divide the investigation and extraction of features equally among our team members. Our remaining tasks include integration of features into a weight prediction algorithm (Jon), parameter estimation for the weight prediction algorithm (Kartik), and developing a testing framework for citation predictions (Zhengzhong).

6.2 Current plan of activities

Software to write:
We have finish the basic evaluation framework and feature extractors. We still need to merge our code of data preparation with our training code. We also need to do more data clean up to get higher quality features from data.

Papers to read:
Although we have finished literature review, we still need to read some selected papers in our related works again to get a deeper understanding to the problem, important papers including Lao & Cohen(2010),.Yu et,al (2012) and Backstrom & Leskovec(2011)

Future activities:
As the general framework of our project has been established, our future activities will be experiment centric. We will do experiment on different methods and variations to get the best out from our model. The most important experiments to do would be feature selection and loss function experiments.

Team responsibilities:
Jon will continue work on feature extraction and data clean up, Kartik will work on LDA feature and similarity measures, and Zhengzhong will merge project framework and conduct experiments on the model decisions listed in the Experiment section. All members will help with experiments once the framework is finalized.