2011 23rd IEEE International Conference on Tools with Artificial Intelligence

# Ranking in Co-effecting Multi-Object/Link Types Networks

Bo Zhou, Manna Wu, Xin Xia and Chao Wu
Computer Science College, Zhejiang University
38 Zheda Road, Hangzhou, 310027, China
bzhou@zju.edu.cn,wmn.daisy@gmail.com,{xxkidd,wuchao}@zju.edu.cn

***Abstract*—Research on link based object ranking attracts increasing attention these years, which also brings computer science research and business marketing brand-new concepts, opportunities as well as a great deal of challenges. With prosperity of web pages search engine and widely use of social networks, recent graph-theoretic ranking approaches have achieved remarkable successes although most of them are focus on homogeneous networks studying. Previous study on co-ranking methods tries to divide heterogeneous networks into multiple homogeneous sub-networks and ties between different sub-networks. This paper proposes an efficient topic biased ranking method for bringing order to co-effecting heterogeneous networks among authors, papers and accepted institutions (journals/conferences) within one single random surfer. This new method aims to update ranks for different types of objects (author, paper, journals/conferences) at each random walk.**

***Keywords-Link based object rankink ; homogeneous networks; heterogeneous networks; random walk***

## I. INTRODUCTION

Link mining is a newly emerging research area that is at the intersection of the work in link analysis, hypertext and web mining, relational learning and inductive logic programming, and graph mining [1]. It is widely used in fields such as bioinformatics, information retrieval, social network analysis, security and law enforcement data, bibliographic citations, epidemiological data, trust networks and collaborative filtering problems. Perhaps the most well-known link mining task is that of link-based object ranking (LBR), which is a primary focus of the link analysis community in order to exploit the link structure of a graph to order or prioritize the set of objects within the graph [2]. LBR breaks the biggest limitation of traditional data ranking, which suppose ranked targets should be independent, identically distributed. Instead, what LBR encounter is more complicated but closer to the reality studied data structure, which means ranked objects have dependency that typical traditional ranking algorithms might not work.

The social network analysis (SNA)'s becoming prevalent also demonstrates the profitable business value of this newly area properly. SNA research giants such as Facebook, LinkIn, Myspace and Twitter also set good examples for our open-minded marketing exploiters. However, the remarkable meaning of LBR is not limited to its huge marketing value and profitable business models. To some extent, it breaks the strict restriction of IID (independent, identically distributed) assumption on studied data structure of traditional object ranking approaches and consider more complex data structure which might be presented as object-related (dependence), multiple object types or link types (rich structure including both homogeneous and heterogeneous networks) and dynamic (event based network data), however, these data structure may be broader used. Quantitative evaluation of researchers' contributions has become an increasingly important topic since the late 80's due to its practical importance for making decisions concerning matters of appointment, promotion and funding [3]. Effective models for heterogeneous networks are still under-researching.

This paper takes papers/authors/publishers ranking as a study case, proposes a newly LBR model to address ranking problems for this kind of co-effective heterogeneous networks. Rest of the paper is organized as bellows. Section 2 gives a brief summary of link based object ranking algorithms background. Section 3 introduces the designation of this single random surfer link. Moreover, subject bias of publishers (journals/conferences) is considered to improve the accuracy of ranking results, which will be discussed on section 4. Loss function is specified in section 5 while the experimental results and algorithm evaluation are presented in section 6. Finally, in section 7 the conclusions are drawn.

## II. SINGLE RANDOM SURFER MODEL

### A. Former LBR model for authors/papers network

Consider the authors/papers/publishers network which has multiple entity types and links types. Standard PageRank model or HITS model won't work in this case. It is impossible to sum up a general model available for all kinds of heterogeneous networks due to the unpredictable entity types and links types. The authors/papers/publishers network represents one kind of heterogeneous networks. Basically, a paper being referred many times should be a good one. Also, authority of authors affects ranking scores of their contribution, while authors' ranking subjects to quality of their publish papers. Moreover, good papers are usually collected by authoritative publishers, while ranking of publishers subjects to amount of excellent papers they collect.

Former research on authors/papers network ranking divide this double entity types network into two single entity type sub-networks. Supposed GA is the authors' relationship network; GD is the papers referring network; GAD is bipartite graph represents authorship. Three random walks are necessary in this framework, one on GA, one on GD and one on GAD. To some extent, GA is a social network

1082-3409/11 $26.00 © 2011 IEEE
DOI 10.1109/ICTAI.2011.84

describing authors' behaviors. First, random walks on GA and GD will be proceeding respectively and come out with 2

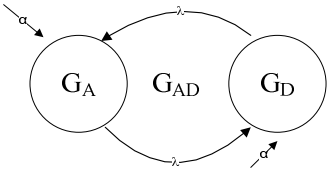


Figure 1 co-ranking model (3 random walks)

ranking results. Third, Inter-class random walk on GAD will suffice to be described by an $n_A \times n_D$ matrix AD and an $n_D \times n_A$ matrix DA, since GAD is bipartite [3]. The final ranking of GA and GD will be generated after the completion of the third random walk proceeding.

### *B. Main idea of Co-ranking algorithm with single random walk*

The main reason why one author has high ranking score is good quality and larger number of his/her papers. Based on this consideration, the random walk on GA is dropped. Only one random walk is needed in this LBR model. This kind of networks is called partial heterogeneous networks. In the random walk proceeding, papers reference will be studied, as it is main contribution of papers' ranking scores. Also, authority of writing authors and accepted publisher(s) are important factors that affect how these papers ranked. At the end of each iteration, the new coming out ranking scores of papers update ranking score of each author they are written by as well as score of each publish they are accepted. New scores of publishers and authors will adjust papers ranking scores in the next iteration too.

### *C. Analysis of Co-ranking algorithm with single random walk*

Use the standard PageRank model to deal with the papers reference sub-network. Denote GD to represent relationship of papers co-reference network. Initialize a primitive matrix $\overline{\overline{D}}$ for GD following standard PageRank algorithm's primitive matrix generation. $\overline{\overline{D}}$ is an $N_D \times N_D$ matrix ($N_D$ is the total amount of papers in the network).

- Primitive Matrix Initialization:
  1. Generate transition matrix $D$ based on the conference links on $G_D$.
  2. $D$ to $\overline{D}$ (dispatching probabilities of dangling links)
  3. Transfer stochastic matrix $\overline{D}$ to primitive matrix $\overline{\overline{D}}$.
  4. Supposed initial vector of probabilities random surfer stays at $Paper_i$. For example $\overrightarrow{D\_Rank_0} = (1/N_D,...,1/N_D)$.

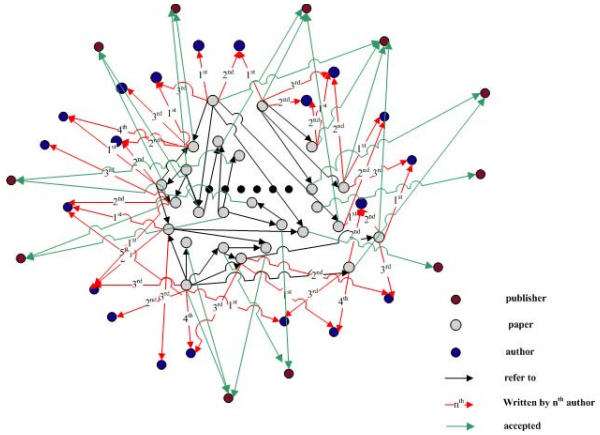


Figure 2 authors/papers/publishers network

- Ranking of Papers at i+1[th] iteration:

$$\overrightarrow{D\_Rank_{i+1}} = \overrightarrow{D\_Rank_i}\,\overline{\overline{D}} \quad (1)$$

- Consider the influence of papers' written authors and accepted publishers

$$\overrightarrow{D\_Rank_{i+1}} = \sigma \times \overrightarrow{D\_Rank_i}\,\overline{\overline{D}} + \varsigma \times \sum_k A\_Rank_i(k) \times \lambda_k + \tau \times P\_Rank_i(j) \quad (2)$$

$A\_Rank_i(k)$ : Paper's k[th] writer's ranking score in i[th] iteration.

$\lambda_k$ : Weight of paper's k[th] writer's effect on adjusting paper's ranking score.

$P\_Rank_i(j)$ : Ranking score of publisher j in ith iteration.

$\sigma$, $\varsigma$ and $\tau$ are used to adjust impact proportion of three parts in formula (4).

- Initialization of ranking score of publishers and authors:

$$\overrightarrow{A\_Rank_0} = (1/N_A,...,1/N_A)$$
$$\overrightarrow{P\_Rank_0} = (1/N_P,...,1/N_P)$$

$N_A$ is the total number of authors and $N_P$ is the number of publishers.

- Relationship Matrices of authorship and publish:

$A_{N_D \times N_A}$ : $N_D \times N_A$ matrix which is initialized based on the authorship graph.

$\overline{A}_{N_D \times N_A}$ : $N_D \times N_A$ matrix which also consider co-authors' impact factor in papers.

$P_{N_D \times N_P}$ : It is a $N_D \times N_P$ matrix which is initialized based on the papers accepting relationship graph.

$\vec{\lambda}_{N_D \times N_A}$ : It is a $N_D \times N_A$ matrix.

$A_{ij}$ is the ranking factor of $Author_j$ contributed by $Paper_i$.

$P_{ij}$ is the ranking factor of $Publisher_j$ contributed by $Paper_i$.

$\lambda_{ij}$ is $Author_j$ 's impact factor on $Paper_i$. Usually, first writer will contribute larger proportion of its ranking than second writer.

Calculate $\overline{A}_{N_D \times N_A}$ :

$$\overline{A}_{ij} = \vec{\lambda}_{ij} A_{ij} \quad (3)$$

- Ranking of Authors at i+1[th] iteration:

$$\overrightarrow{A\_Rank_{i+1}} = \overrightarrow{D\_Rank_i} \times \overline{A}_{N_D \times N_A} \quad (4)$$

Merge formula (5) into (6), so

$$\overrightarrow{A\_Rank_{i+1}} = \overrightarrow{D\_Rank_i} \times \vec{\lambda}_{ij} A_{ij} \quad (5)$$

- Ranking of Publishers at i+1[th] iteration:

$$\overrightarrow{P\_Rank_{i+1}} = \overrightarrow{D\_Rank_i} \times P_{N_D \times N_P} \quad (6)$$

- Normalization:

$$\overrightarrow{D\_Rank} \rightarrow \| \overrightarrow{D\_Rank} \|_1$$
$$\overrightarrow{A\_Rank} \rightarrow \| \overrightarrow{A\_Rank} \|_1$$
$$\overrightarrow{P\_Rank} \rightarrow \| \overrightarrow{P\_Rank} \|_1$$

## III. EXPERIMENTS

### A. Data Preparation

In this paper, we have a web crawler to retrieve data mining related papers information from CiteSeerX. Consider the limitation of real experimental dataset, such as not enclosed relation networks (it is difficult to retrieve the whole dataset). We have some special disposal for the original algorithm, which is expected to deal with N' objects' ranking issue within an N (N>N') objects network.

In this paper, 434 papers will be ranked which are within the query result of data mining from CiteSeerX. However, 16726 papers will involve in this single random surfer algorithm as more papers which cite the ranked papers should be included in this model. The author set has a number of 26538. All these authors are collected from the papers which involve in ranking iterations, including the ranked papers and citing papers.

### B. Experiment Steps

A complete PageRank based model should base on an enclosed data set, meaning all kinds of link calculations won't escape from this data set. It should be a huge data set. It is difficult to get enclosed data set as if links of one citing paper involves, neighbors of it should be considered as well. The data set will explode quickly.

To simplify the experiment, we have some special disposal. Our core PageRank model for papers' ranking is based on a $16726 \times 434$ (total papers number * ranked papers number), instead of a $16726 \times 16726$ matrix. Meanwhile, we have some special initialization for each kind of object as well as their relationship matrices. Description of each experiment step will be introduced as below.

- Step 1: Data Set Preparation

Crawl data from the CiteSeerX and DBLP websites with the query of date mining, merging info from these two websites. Analyze and parse the necessary important information from the raw material, such as paper's DOI number which will be unique identification of each paper, authors including their order of each paper, and publishers as well. A fuzzy matching program is used to identify same object in different systems.

- Step 2: Matrices Building

Identity Matrix I with $16726 \times 434$ elements:

$$\vec{I}(16727,434) : ones(16727,434)$$

Initial D Matrix describing initial probabilities of all these 343 ranked papers, therefore, $\overline{\overline{D}}$ after irreducible transformation is:

$$\overline{\overline{D}} = \partial \times \overline{D} + (1-\partial)/343 \times \vec{I}(0 < \partial < 1)$$

Matrix P expresses the relationship from paper to author. For example, P (i, j) represents paper i's j[th] author. It is created by two parts in this experiment, including TP1 for those 434 ranked papers and the rest citing papers which won't join the final result comparing. The first part is under calculation, while the other part estimated by each paper' citation amount.

$$P = [TP1 \quad TP2]$$

$$TP1 = (\text{sum}(P(1:1,1:434),2)/(\text{sum}(P,2) * 434)) * P(1:1,1:434)$$

$$TP2 = (1/\text{sum}(P,2)) * P(1:1,435:\text{size}(P,2))$$

Matrix A_Factors explains the impact factors of authors in order.

$$\text{A\_Factors}(16726,34) = \left[\mu^1 \quad \ldots\ldots \quad \mu^n\right] \ (\max \text{co-authors number})$$

$$\overrightarrow{D\_Rank_{i+1}} = \sigma \times \overrightarrow{D\_Rank_i}\, \overline{\overline{D}} + \varsigma \times \sum_k A\_Rank_i(k) \times \lambda_k + \tau \times P\_Rank_i(j)$$

### C. Ranking Result Analysis

#### 1) Ranking Result with one parameter set

$$\begin{cases} \partial = 0.5 \\ \sigma = 0.6 \\ \varsigma = 0.25 \\ \tau = 0.15 \end{cases}$$

Authors' impact factors matrix on their paper ($\mu = 0.6$)

$\left[\mu^1 \quad \ldots\ldots \quad \mu^n\right]$

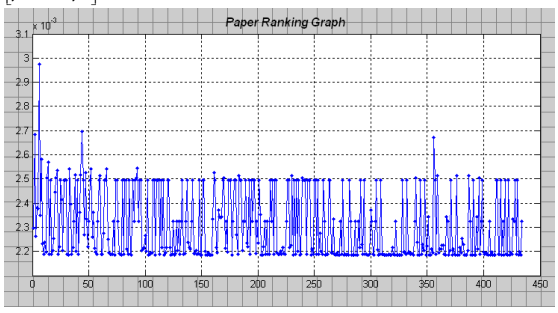


Figure 3 Paper Ranking Graph

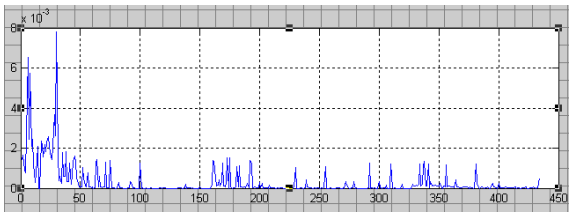


Figure 4 Paper Citation Graph

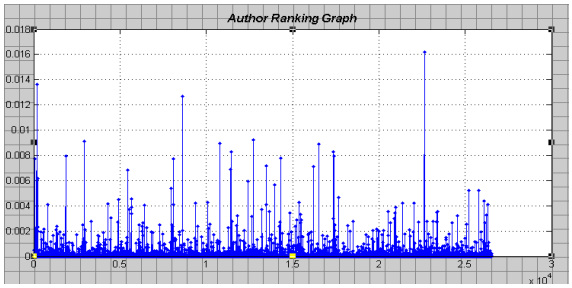


Figure 5 Author Ranking Graph

As the graphic shows above, papers' ranking value not strictly go with the number of citation, however, papers with high citation count and good quality publisher are more possible to earn higher ranking value. For example, paper #6 with 1268 citations which is also accepted by VLDB ranks in the top. It is unfair to decide papers' ranking value only base on their citation. In the real cases, some of those valuable papers might not have a great deal of citations because they are newer ones. Paper with smaller citations counts should not be judged as not so important ones.

Conferences such as KDD, SIGMOD, and ICFP are ranked to be the top. This ranking might not be correct because the dataset is not precise enough. This distinguish of the same publish institution with several display names is not precise enough. There are noisy data in the dataset we used.

### *D. Parameters Setting and Convergence*

The parameter set in this model is $(\mu,\sigma,\varsigma,\tau)$ . $\mu$ specifies the different impact factors of authors, while $(\sigma,\varsigma,\tau)$ represents the weights of impact factors by paper ranking scores, authors ranking scores and publishers ranking score in last iteration.

$\mu$ differentiates importance of paper writers in order. Small $\mu$ determines less impact factor later authors' effect on related papers. If $\mu$ is small enough, the model will be almost like only the first authors will contribute their reputations on those papers. $(\sigma,\varsigma,\tau)$ specifies the impact factors of paper/author/publisher ranking in last iteration doing on related papers. If $\sigma$ is big enough, this model will be simplified as papers related PageRank algorithm.

## IV. Conclusion and Future Work

In this paper, a single random surfer link base object ranking algorithm is explored according the study of relationship between papers, authors and publishers. In this algorithm, papers' scores are co-contributed by their citing papers, writing authors and publishing institutions. This model aims to solve partial heterogeneous networks' objects ranking more efficiently. The main idea of this model is combining several random surfers to be a single one. Authors and publishers' ranking scores will also be updated after each paper PageRank iteration. According the experiment result, ranking result of this model is considerable than the citation scores ranking, also more efficient than those $N^2$ random surfers models. However, this algorithm works only on partial heterogeneous cases such as paper-author-publisher.

Subject bias model should be studied to reach more reasonable result. Meanwhile, publish year is important factor too. New coming papers might have fewer citations, but it does not mean they are not good enough.

## Acknowledgment

The authors wish to thank CiteSeer and DBLP website, which provide free and abundant experiment dataset. Their great efforts on maintaining and updating these data are really appreciated.

## References

[1] Michael l. Jordan. Graphical Models. Statistical Science, 19(1): 140-155, 2004

[2] Lise Getoor and Ghristopher P. Diehl. Link Mining: A Survey. SIGKDD Explorations, 7(2), 2005.

[3] Ding Zhou, Sergey A. Orshanskiy, Hongyuan Zha, and C. Lee Giles. Co-Ranking Authors and Documents in a Heterogeneous Network. In proceeding of IEEE ICDM, 2007.

[4] Lise Getoor. Link mining: A new data mining challenge. SIGKDD Explorations, 5(1):84-89, 2003

[5] Michelangelo Diligenti, Marco Gori, and Marco Maggini. Learning Web Page Scores by Error Back-Propagation. In IJCAI, 2005.

[6] A. Agarwal, S. Chakrabarti, and S. Aggarwal. Learning to rank networked enetities. In proceeding of ACM SIGKDD, 2006.

[7] H. Change and D. Cohn. Learning to create customized authority lists. In ICML, 2000.

[8] W. W. Cohen and E. Minkov. A graph-search framework for associating gene identifiers with documents. BMC Bioinformatics, 7(440), 2006.

[9] M. Collins and T. Koo. Discriminative reranking for natural language parsing. Computational Linguistics, 31(1): 25-69, 2005.

[10] E. Minkov, W. W. Cohen, and A. Y. Ng. Contextual search and name disambiguation in email using graphs. In SIGIR, 2006.

[11] Z. Nie, Y. Zhang, J. R. Wen, and W. Y. Ma. Object-level ranking: Bring order to web objects. In WWW, 2005.

[12] E. Minkov and W. W. Cohen. Learning to Rank Typed Graph Walks: Local and Global Approaches. In WEBKDD/SNA-KDD Workshop, 2007.

[13] K Avrachenkov. The effect of new links on Google PageRank. Stochastic Models, 22:319-331, 2006.