Ranking Authors in Digital Libraries

Sujatha Das
Computer Science and Engineering
The Pennsylvania State University
University Park, PA-16802
gsdas@cse.psu.edu

Prasenjit Mitra, C. Lee Giles
Information Sciences and Technology
The Pennsylvania State University
University Park, PA-16802
{pmitra, giles}@ist.psu.edu

ABSTRACT

Searching for people with expertise on a particular `topic` also known as *expert search* is a common task in digital libraries. Most models for this task use only documents as evidence for expertise while ranking people. In digital libraries, other sources of evidence are available such as a document's association with venues and citation links with other documents. We propose graph-based models that accommodate multiple sources of evidence in a PageRank-like algorithm for ranking experts. Our studies on two publicly-available datasets indicate that our model despite being general enough to be directly useful for ranking other types of objects performs on par with probabilistic models commonly used for expert ranking.

1. INTRODUCTION

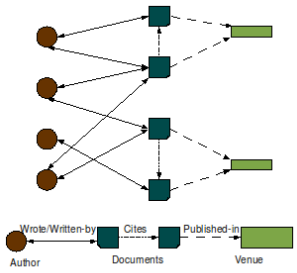
The expertise modeling and expert ranking tasks have been addressed in different flavors in previous research. For example, the enterprise task at TREC requires searching for people in a given enterprise having the required expertise to answer a specified query [1]. On the other hand, the expertise modeling or topical profiling task involves query-independent modeling of a person's expertise [4, 17]. Expert finding and expertise modeling are related in that expert finding techniques can make use of the profiles obtained from expertise modeling. However, several expert finding algorithms avoid explicit modeling of experts (known as candidate-centric approaches) and instead adopt a document-centric approach where ranking is based on a subset of documents obtained using the query. For both expert ranking and expertise modeling, the "evidence" of expertise depends on the context. For instance, in the TREC task, this evidence is obtained from various kinds of documents available in the enterprise such as e-mail communications, people's homepages, technical reports, resumes etc. In digital libraries, evidence of expertise is calculated based on the publications of an author on a given topic, citations accumulated by these publications etc [6, 24].

[1] http://trec.nist.gov/

JCDL'11, June 13–17, 2011, Ottawa, Ontario, Canada.

We focus on the query-dependent expert ranking task in digital library collections such as CiteSeerX. In this setup if the focus is only on obtaining the top-k experts (as opposed to all experts) for a given topic, Balog, et al. showed that document-centric approaches perform on par with the candidate-centric approaches [2]. We study the use of graph-based document-centric models for ranking experts in digital library collections. Collections such as CiteSeerX are centered around objects such as documents, authors, venues and affiliations with various relations among the objects. Such relational data is naturally represented in terms of a heterogeneous graph. For instance, a sample, partial graph generated by the document collection in CiteSeerX is shown in Figure 1, with author, venue and document nodes (which could indicate research papers, homepages etc.) and edges

Figure 1: Sample typed graph



between document-document, document-authors and document-venue nodes. Researchers have suggested applying random walk models on graphs for deriving a measure of importance of a node based on structure of the graph [20, 13]. More recently, Minkov, et al. [18] suggested methods for learning query-dependent ranking functions on graphs using spreading activation models.

Almost all work in expert ranking so far primarily deals with only document and author nodes and the proposed models do not seem easily extendible when additional sources of information are available. On the other hand, folding in other sources such as affiliation or the venue information are likely to yield more accurate rankings. For instance, a paper published in `JCDL` might be treated as more indicative of expertise if the query topic is `digital libraries` than some other conference venues. In this paper, we provide a preliminary investigation into extending PageRank-like models for ranking different types of objects in digital libraries. While our experiments deal with ranking authors in digital libraries in response to topic queries, our ranking model is general enough to be applicable to any type of objects.

PageRank-like models that use relevance propagation have been previously employed in various applications (including expert search). Our formulation particularly focuses on obtaining simple equations (extending PageRank equations for heterogeneous graphs) that enable efficient methods for computing scores and fewer parameters to tune (or learn) for individual ranking applications.

Section 2 summarizes related work on expert finding whereas Section 3 provides more details on our model. Our datasets, experimental setup, results and observations are presented in Section 4 while Section 5 concludes the paper.

2. RELATED WORK

The expert search task was studied in various contexts such as enterprise collections (of TREC), sparse data university environments [3] and for bibliographic data and digital libraries [6, 27]. Previously, experts were ranked using probabilistic models [2, 3], topic models [6, 8, 23, 25], graph-based approaches [7, 27, 22], vector-space [5] and voting models [15]. By and large, probabilistic methods seem to be the commonly applied models in view of their simplicity and run-time efficiency. We compare our model with the probabilistic models in Section 4.

Variants of the HITS [13] and PageRank [20] algorithms were employed for scoring objects in different applications [11, 19, 12, 26]. In particular, relevance propagation models for expert search were studied in e-mail collections by Dom, et al. [7] and enterprise collections by Serdyukov, et al [22]. Zhou, et al. [27] proposed a coupled random walk model between authorship networks and citation networks for ranking authors and documents together. Liu, et al. [14] evaluate the impact of an individual author by computing AuthorRank, a query-independent measure similar to PageRank based on the co-authorship network in digital libraries.

3. RANKING NODES IN TYPED-GRAPHS

Consider an input graph $G = (V, E)$ where V is the set of nodes and the edges in E are directed and have types assigned to them ($t \in T$). Similarly, nodes also have types associated with them. Transitions between nodes possible via edges of type t can be represented by a matrix P_t. Let

$$P = \sum_{t \in T} w_t P_t \tag{1}$$

where $\sum_t w_t = 1, \forall t \in T, w_t \geq 0$. If we ensure that the matrix defined in Equation 1 is irreducible and aperiodic by Ergodic theorem, the Markov chain defined by P has a unique stationary (or limiting) distribution [21]. This distribution over nodes of V, can be obtained by computing the principal eigen vector of the transpose of P (for example, using the power method [9]). Given a random surfer who behaves as described by the transition matrix P, the limiting probability of a particular node in G, denotes the probability with which the surfer visits that node in the limit (i.e. as time tends to infinity). These probability values are deemed as the "importance" scores of nodes using which the nodes can be relatively ranked w.r.t. each other in the graph. PageRank works on the Web graph where the nodes represent 'webpages' and edges are either of the type 'hyperlinks' or dummy edges (added to ensure irreducibility). PageRank and the related HITS algorithms were adapted for ranking nodes in several systems [11, 12, 26]. In Equation 1, the value $P_t(i, j)$ denotes the probability of jumping to node j from i via an edge of type t. The w_t values can be viewed as the importance of a particular edge-type during the authority flow in the graph. For instance, in PageRank the dummy edges are typically assigned "edge-type importance" between 0.1 to 0.2 (damping coefficient).

We adopt the original PageRank formulation in assigning transition probabilities in the query-dependent graphs for expert ranking. During the random walk, a surfer at node u at time step k, chooses a type of transition t from available types of transitions with probabilities given by the edge-type weight parameters (w_ts). Once a transition type t is chosen, the probability of being at v where $\{v|u \rightarrow_t v\}$ at time step $k+1$ is set to $\frac{1}{deg_t(u)}$. That is, once an edge-type is chosen, an outgoing edge among the edges of that type is chosen with uniform probability. The existence of steady-state distribution in this graph is ensured by making the underlying aggregate transition matrix aperiodic and irreducible as suggested by Haveliwala, et al [9]. That is, to handle nodes with no outgoing edges, uniform jumps from these dangling nodes are added to all other nodes, whereas irreducibility is ensured by making one of the transition matrices in Equation 1 to be $E = [1]_{n \times 1} \times \boldsymbol{v}^T$ where n is the number of nodes in G and $\boldsymbol{v} = [\frac{1}{n}]_{n \times 1}$. The jumps enabled by E are as referred to as teleportation whereas $\boldsymbol{v}$ is called the teleport or personalization vector or reset distribution. Topic-sensitive and personalized variants of PageRank manipulate the entries in $\boldsymbol{v}$ to bias jumps to distributions of interest [10]. Minkov [18] experimented with random walk models for several entity-similarity tasks and suggested setting the teleport vector based on the nodes initially activated by the input query for capturing the query-dependent aspects.

For expert search, for a query q, we obtain the graph, G_q on which the PageRank equations are computed by first using q to retrieve an initial set of document nodes. Various available relations are then used to expand the original set of document nodes. For instance author nodes of the documents can be added via the `written-by` edges whereas other documents can be added via the `cited-by` edges. We experimented with two settings for the teleport vector. The retrieval scores of the initial set of document nodes are normalized and used as the reset distribution. For the second setting, we use a uniform reset distribution for $\mathbf{v}$.

Nie, et al. proposed the PopRank model for ranking objects on the Web [19]. Serdyukov, et al. used PopRank-like equations for ranking experts in the enterprise setting [22]. Both these models suggest PageRank-like equations where the damping coefficient (that handles irreducibility) is treated differently from other edge-types. For example, PopRank uses $R_X = \epsilon R_{EX} + (1 - \epsilon) \sum_{\forall Y} \gamma_{YX} {M_{YX}}^T R_Y$ where X and Y refer to types of objects, R_X and R_Y, the popularity scores for objects of type X and Y, M_{YX}, the adjacency matrices, γ_{YX}, the popularity propagation factor from objects of type Y to objects of type X, R_{EX}, the web popularity scores for objects of type X and ϵ the damping factor. Compared to these approaches, our model treats all edge-types (including those added for irreducibility) and node-types uniformly leading to simpler equations. Minkov [18] uses edge-type weights to describe transition matrix entries as $Pr(x \rightarrow y) = \frac{\sum_{l \in L_{xy}} \theta_l}{\sum_{y' \in ch(x)} \sum_{l' \in L_{xy'}} \theta'_l}$ where $ch(x)$ refers to children of x, θ_l is the weight assigned to edge-type l and L_{xy} is the set of edge types of the outgoing edges from x to y.

Instead of the transition matrices obtained in this way, our formulation results in an aggregate matrix that is a linear combination of individual transition matrices. The equations obtained from our modeling make it straightforward to extend the efficient techniques proposed by Haveliwala, et al. [9] (for computing PageRank) for our case. We next describe the details of these extensions.

Let P_t refer to an individual transition matrix (describing edges of the edge-type t) and let

$$P'_t = P_t + \boldsymbol{d}_t.\boldsymbol{v}^T \quad (2)$$

$$E = [1]_{n\times 1} \times \boldsymbol{v}^T, v = [\frac{1}{n}]_{n\times 1} \quad (3)$$

$$P'' = \sum_t w_t P'_t + (1 - \sum_t w_t)E \quad (4)$$

Here $\boldsymbol{d}_t$ is an n-dimensional vector for edge-type t identifying nodes having no edges of type t. That is $d_t(i) = 1$ if node i has no outgoing edges of type t and is 0 otherwise. The final scores are obtained by computing the eigen vector of the transpose of the transition matrix, $A = P''^T$. Starting with a initial vector x, this involves repeated computation of Ax until convergence. For a single transition matrix, Haveliwala, et al. propose an efficient algorithm for this computation that makes use of the sparsity of the underlying transition matrix and avoids having to explicitly maintain rows for dangling nodes in the transition matrix and instantiating E. We extend these techniques for our model by defining P and *temp* as follows:

$$P = \sum_t w_t P_t$$

$$temp = \sum_t w_t d_t + (1 - \sum_t w_t)\vec{1}$$

Algorithm 1 enumerates the steps that need to be repeated until convergence for obtaining the scores for all nodes in the graph for a given query. In particular, nodes of the desired type (author nodes for expert search) can be relatively ranked w.r.t. each other based on their scores. Using a sparse matrix library, the product $P^T x$ can be computed efficiently and the remaining computations only require operations on vectors.

Algorithm 1 Computing Ax where $A = P''^T$

$\alpha = temp^T x$
$y = P^T x + \alpha \boldsymbol{v}$
$x = y$

4. EXPERIMENTS

We evaluate the performance of Algorithm 1 on the following freely available datasets for expert search.

1. **The ArnetMiner Dataset** [2] This dataset contains a small set of 7 queries along with expert lists collected by pooling results from Libra, Rexa and ArnetMiner digital library search engines that were judged for relevance by three volunteers. This dataset was previously used for evaluating expert finding in bibliographic data [6]. We use the CiteSeerX document collection as our corpus for evaluating expert search with this dataset.
2. **The UvT Expert Collection** The UvT expert collection was harvested from the Webwijs system developed at Tilburg University in the Netherlands in October 2006 [3]. This collection has information for about 1168 experts who are represented using their homepages, course pages, research descriptions and publications lists. This collection was used by Balog, et al. for evaluating expertise retrieval in sparse data environments [3].

[2] http://arnetminer.org/lab-datasets/expertfinding/

[3] http://ilk.uvt.nl/uvt-expert-collection/

A summary of these datasets is provided in Table 2.

Table 2: Summary of Datasets

Dataset	#Queries	#Experts	Corpus Size
ArnetMiner	7	245	796673
UvT	1491	1168	19127

4.1 Baseline Method

We compare the performance of our model with that of the probabilistic model commonly used for expert search. Balog, et al. [2, 3] use the estimates of $p(ca|q)$ where q is the query and ca is a candidate for ranking experts. $p(ca|q) = \frac{p(ca,q)}{p(q)}$ and $p(ca, q)$ is defined as

$$p(ca, q) = \sum_{d\in D} p(d)p(ca, q|d) = \sum_{d\in D} p(d)p(q|d)p(ca|d, q) \quad (5)$$

D is the set of documents related to the query that a candidate ca is associated with. Assuming ca is conditionally independent of q given a document one can write $p(ca|d, q) = p(ca|d)$ and treating $p(d)$ and $p(q)$ as uniform $p(ca|q) \propto \sum_{d\in D} p(q|d)p(ca|d)$. $p(ca|d)$ is defined as $\frac{a(d,ca)}{\sum_{c'\in C} a(d,c')}$ where C is the set of all candidates and $a(d, ca)$ is the association between document d and candidate ca. The $p(q|d)$ scores for a document are estimated using language modeling. Deng, et al. extended the probabilistic model proposed by Balog, et al. for the bibliographic data [6]. The $p(ca|d)$ values were defined as $\frac{1}{n_d}$ or 0 depending on whether ca is the author of d and n_d is the number of authors for d. The prior probability, $p(d)$, was defined in terms of the number of citations that the document has. For example, $p(d) \propto ln(e + c_d)$ where c_d is the citation number for d.

4.2 Results and Observations

For the ArnetMiner dataset, we use Deng, et al.'s variant of the probabilistic model as the baseline since citation information is available for the CiteSeerX corpus. A set of 100 documents is first retrieved with the query and these nodes are expanded to form the graph by adding author and document nodes (via the `written-by` and `cited-by` edges). For the UvT collection, the type information of the document is available. The document could be a homepage(h), a research description (r), a publication(p) or a course page(c). We use the query to retrieve 100 pages of each type and then use the author associations provided in the dataset to obtain the authors(a) to build the graph. The mean average precision (MAP) and mean reciprocal rank (MRR) measures typically used for ranked retrieval [16] are reported in Table 1. In this table, we report the baseline performance and the performance with our model for a specific edge-type weights configuration. For instance, on the ArnetMiner dataset, the author to document and document to author edges were given the weights, 0.1 each whereas document to document edge weights were set to 0.7. Table 3 illustrates the dependence of our model's performance on these weights for the UvT collection. Given sufficient training data with correct ranked lists, graph-walk parameters can be automatically learnt. Machine learning techniques for learning walk parameters for PageRank on the Web and random walks that capture similarity between nodes in an entity graph were previously studied [19, 1, 18].

Table 1: Performance of our model vs. the baseline probabilistic model

Dataset	Setting	MAP@10	MRR@10	MAP@20	MRR@20	MAP@30	MRR@30
ArnetMiner	Baseline	0.1778	0.3381	0.1709	0.3381	0.1615	0.3381
	Our Method	0.2122	0.5	0.1825	0.5119	0.1599	0.5119
UvT	Baseline	0.1430	0.3319	0.1165	0.3379	0.1029	0.3396
	Our Method	0.1438	0.3319	0.1169	0.3377	0.1034	0.3397

From the tables it might seem that there is no apparent benefit to using our model when compared to the probabilistic model. However, note that our model is not specific to experts (author nodes) per se but indeed other nodes (such as venues) could be similarly ranked without any changes to the model. Venue and affiliation information was not available in our experimental datasets (we are assembling a dataset using data from CiteSeerX). Instead all nodes are document nodes, which can be modeled via the probabilistic method which also might explain the lack of large performance benefits. However, when other nodes and connections are available, only the graph building process undergoes a change rendering our model more flexible. In theory, we could also extend probabilistic models, but estimating, joint probabilities in case of several variables is usually more difficult and independence between variables is assumed to handle this problem. On the other hand, since all nodes are treated uniformly in our model, adding another source simply corresponds to adding another set of nodes with the corresponding edges.

Table 3: Performance on the UvT collection for various edge-type weight settings

Setting	MAP@10	MRR@10
ac=0.45, ca=0.45	0.0869	0.2154
ah=0.45, ha=0.45	0.0427	0.1009
ap=0.45, pa=0.45	0.1115	0.2668
ar=0.45, ra=0.45	0.1218	0.2887
all weights=0.1	0.1429	0.3317

5. SUMMARY AND FUTURE WORK

We proposed graph-based algorithms for ranking experts in response to topic queries in the context of a digital library. Our model has the natural ability to accommodate available evidence from different objects and relations and provides handles to tune the effect of evidence from each source based on domain knowledge or relevance feedback. Our experiments with two publicly available datasets highlight the effectiveness of our model in ranking experts in digital libraries. We are currently focusing on improving the expert ranking performance using evidence from researcher homepages and venue information. Moreover our model needs to be experimentally validated for ranking other types of nodes such as venues and affiliations. An orthogonal line of interest is related to applying machine learning techniques for learning the egde-type weights based on known rankings in the training set for tuning the random walk instead of setting them with the help of domain experts.

6. REFERENCES

[1] A. Agarwal, S. Chakrabarti, and S. Aggarwal. Learning to rank networked entities. In *KDD*, 2006.

[2] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR*, 2006.

[3] K. Balog, T. Bogers, L. Azzopardi, M. de Rijke, and A. van den Bosch. Broad expertise retrieval in sparse data environments. In *SIGIR*, 2007.

[4] K. Balog and M. de Rijke. Determining expert profiles (with an application to expert finding). In *IJCAI*, 2007.

[5] G. Demartini, J. Gaugaz, and W. Nejdl. A vector space model for ranking entities and its application to expert search. In *ECIR*, 2009.

[6] H. Deng, I. King, and M. R. Lyu. Formal models for expert finding on dblp bibliography data. In *ICDM*, 2008.

[7] B. Dom, I. Eiron, A. Cozzi, and Y. Zhang. Graph-based ranking algorithms for e-mail expertise analysis. In *SIGMOD workshop, DMKD*, 2003.

[8] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 2004.

[9] T. Haveliwala, S. K. Kamvar, D. Klein, C. Manning, and G. G. Golub. Computing pagerank using power extrapolation. In *Stanford University Technical Report*, 2003.

[10] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW*, 2002.

[11] A. Hulgeri and C. Nakhe. Keyword searching and browsing in databases using banks. In *ICDE*, 2002.

[12] H. Hwang, V. Hristidis, and Y. Papakonstantinou. Objectrank: a system for authority-based search on databases. In *SIGMOD*, 2006.

[13] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46, 1999.

[14] X. Liu, J. Bollen, M. L. Nelson, and H. Van de Sompel. Co-authorship networks in the digital library research community. *Inf. Process. Manage.*, 2005.

[15] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *CIKM*, 2006.

[16] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2007.

[17] D. Mimno and A. McCallum. Mining a digital library for influential authors. In *JCDL*, 2007.

[18] E. Minkov. Adaptive graph walk based similarity measures in entity-relation graphs. *PhD Thesis*, 2009.

[19] Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma. Object-level ranking: bringing order to web objects. In *World Wide Web*, 2005.

[20] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, 1999.

[21] S. M. Ross. *Introduction to Probability Models, Ninth Edition*. Academic Press, Inc., 2006.

[22] P. Serdyukov, H. Rode, and D. Hiemstra. Modeling multi-step relevance propagation for expert finding. In *CIKM*, 2008.

[23] J. Tang, R. Jin, and J. Zhang. A topic modeling approach and its integration into the random walk framework for academic search. In *ICDM*, 2008.

[24] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *KDD*, 2008.

[25] Y. Tu, N. Johri, D. Roth, and J. Hockenmaier. Citation author topic model in expert search. In *COLING*, 2010.

[26] D. Walker, H. Xie, K.-K. Yan, and S. Maslov. Ranking scientific publications using a simple model of network traffic. *J.STAT.MECH P*, 2007.

[27] D. Zhou, S. A. Orshanskiy, H. Zha, and C. L. Giles. Co-ranking authors and documents in a heterogeneous network. In *ICDM*, 2007.