International Journal of Innovative Computing, Information and Control
Volume **11**, Number **4**, August **2015**

ICIC International ©2015 ISSN 1349-4198
pp. 1295–1316

RANKING AUTHORS WITH LEARNING-TO-RANK TOPIC MODELING

Zaihan Yang, Liangjie Hong, Dawei Yin and Brian D. Davison

Department of Computer Science and Engineering
Lehigh University
Bethlehem, PA 18015, USA
{ zay206; lih307; day207; davison }@lehigh.edu

Received January 2014; revised March 2015

Abstract. *Topic modeling has emerged as a popular learning technique not only in mining text representations, but also in modeling authors' interests and influence, as well as predicting linkage among documents or authors. However, few existing topic models distinguish and make use of the prior knowledge in regard to the different importance of documents (authors) over topics. In this paper, we focus on the ability of topic models in modeling author interests and influence. We introduce a pair-wise based learning-to-rank algorithm into the topic modeling process with the hypothesis that investigating and exploring the prior-knowledge on authors' different importance over topics can help to achieve more accurate and cohesive topic modeling results. Moreover, the framework integrating learning-to-rank mechanism with topic modeling can help to facilitate ranking in new authors. In this paper, we particularly apply this integrated model into two applications: the task of predicting future award winners of research communities, and predicting future PC members of scientific conferences. Experiments based on two real world data sets demonstrate that our proposed model can achieve competitive ranking performance with several state-of-the-art learning-to-rank or topic modeling algorithms.*

Keywords: Topic modeling, Supervised learning, Expertise ranking, Prediction

1. **Introduction.** Generative topic modeling has become a popular machine learning technique and has shown remarkable success not only in text mining, but also in modeling authors' interests and influence, and predicting linkage among documents (authors). Ever since the success of the original two representative topic models, the pLSA and LDA, which focus on pure content analysis by discovering the latent topics from large document collections, a large body of literature on topic models has been established, mostly by incorporating additional contextual information, such as time, geographical locations, or integrating linkage or social network information. Authorship is one important contextual feature, which when incorporated into topic modeling, can be used to derive the topic distribution over authors rather than documents, and therefore can be used to model authors' interests and influence.

Most of the existing topic models, however, are unsupervised. Documents or authors are treated equally, while no prior-knowledge of their different importance over topics has been explored or investigated. However, this is not the real situation. Sometimes, we can know in advance that some document is more about a certain topic than other documents, and that one author (researcher) is more prestigious in one research domain than other authors. By exploring this prior-knowledge and applying a supervised learning scheme into the topic modeling process, we hypothesize that we can achieve more accurate and cohesive topic modeling results, which can in turn help in better distinguishing the different importance (ranking) of new documents (authors) in terms of their relevance or authority over topics.

In this paper, we concentrate on the ability of topic models in modeling authors' authority (interests or influence) in a research domain[1], a typical task known as expert ranking (expertise ranking or expert finding). In spite of many recent developments fulfilling this task, several challenges still remain. First of all, the sparseness problem in document content would prevent the 'bag-of-words'-based algorithms (term frequency, TF-IDF, language model) from being accurate. It is well-acknowledged that documents related to an author provide strong evidence in evaluating authors' expertise; however, such document content (especially considering the paper abstract) is normally very sparse, and therefore, a 'bag-of-words' based algorithm cannot effectively capture the underlying semantics. The topic modeling approach, however, is believed to provide a better solution in this aspect. Secondly, few existing topic modeling based approaches, however, incorporate additional features such as network based features and temporal features into the topic modeling process to represent an author's authority. Thirdly, most of the existing work on expert ranking rely on carefully designed ranking models based on heuristics or traditional probabilistic principles, rather than applying machine learning techniques to learn ranking functions automatically.

To fulfill the challenges mentioned above, we propose in this paper a supervised learning scheme by incorporating the prior knowledge of the different importance over topics between pairs of authors into the topic modeling process, which results in a framework integrating the pair-wise learning-to-rank algorithm into topic modeling. We name this novel model as LtoR topic modeling (abbreviated as **LtoRTM**). In the training process, we can not only infer the authors' distribution over topics and topics' distribution over words, but also the coefficient representing the different weights of topics. In the testing process, we can infer the topic proportion of new authors. Furthermore, based on the new authors' topic distributions, and the learned coefficient in the training process, we can generate a ranked list of authors in terms of their different importance (authority) across topics.

We go beyond pure contextual information by incorporating additional features into the **LtoRTM** model such as the number of publications or citations of authors, resulting in the **LtoRTMF** (learning-to-rank topic modeling with additional features) model. To evaluate the effectiveness of our proposed models, we apply the model to two expert ranking related applications: the task of predicting community-based future award winners and predicting future PC members of several significant conferences in computer science disciplines.

The models we proposed are essentially a combination of the topic models and learning-to-rank schemes. To properly and effectively integrate the advantages of these two mechansims is the biggest motivation of our work and represents our most significant contribution. We choose to use topic modeling rather than bag-of-words approaches to effectively discover the latent meaning of clusters of words and achieve more coherent contextual anlaysis results. Applying learning-to-rank mechanism into topic modeling can help us more conveniently integrate additional supportive features and train the ranking functions automatically. Moreover, we identify three groups of author-authority related features which measure the expertise of a researcher from multiple aspects. The features we extracted evaluate authors' expertise from a more complete view compared to previous research, and can also be used in other expert-ranking related tasks.

We highlight and sum up the main properties and contributions of our paper as follows.

[1] In the paper, we use research domain, community and its associated query as interchangeable concepts.

- We propose a novel probabilistic topic modeling framework to model authors' research influence and interests. The framework is fundamentally a supervised learning scheme in which we incorporate the prior knowledge on the different importance over topics between pairs of authors into topic modeling process. To our best knowledge, this results in the first framework integrating pair-wise learning-to-rank into topic modeling. We name this model as the **LtoRTM** model.
- We further extend the **LtoRTM** model by further identifying and incorporating supporting features associated with authors' expertise in addition to the pure contextual information, resulting in the **LtoRTMF** model.
- We evaluate the effectiveness of our model by applying it into two applications measuring author authorities: the tasks of predicting future award winners and future PC members. Experiments have been conducted on real-world data sets to test the performance of the proposed model and compare it with several other state-of-the-art topic modeling or learning-to-rank algorithms.

2. **Related Work.** In this section, we review three lines of research work that are related to our work, and discuss the novelty of our work from them.

2.1. **Topic modeling.** Generative topic modeling has become a popular machine learning technique for topic-related content representations. Ever since the success of the original two representative topic models, pLSA [17] and LDA [6], which focus on pure content analysis by discovering the latent topics from large document collections, a large body of literature on topic models has been established, mostly by incorporating additional contextual information, such as time [4], authorship [31, 34, 36], geographical locations [42], or integrating linkage or social network information [7, 11, 27]. The linkage information being modeled often represents the similarity between two linked documents, rather than the difference between documents, which is the focus of our work in this paper.

Blei and McAuliffe proposed a supervised LDA model [5] in 2007, which is a promising improvement over the original LDA, as it converts the topic modeling approach, which is traditionally believed to be an unsupervised learning technique into a supervised one. Several other works [30, 44] have been proposed, following this direction. However, in these works, the labels are often attached to individual documents rather than every pair of documents to distinguish their different preference over topics. Our work, however, borrows the idea of pair-wise learning-to-rank into the topic modeling process.

Duan et al. proposed a ranking-based topic modeling [10], which utilizes the importance of documents and incorporates the TopicalPageRank [28] into topic modeling. Compared with our work, their documents' importance is not defined upon pairs of documents. Moreover, their model is built upon pLSA instead of LDA, and the model is designed for document clustering and classification applications, which are all different from our model.

2.2. **Learning-to-rank.** Learning-to-rank (LtoR for short) [21] is a recent trend of applying machine learning techniques to learn ranking functions automatically. In the standard LtoR setting, a typical training set is composed of queries, documents (represented by a feature set) and their associated labels. A machine learning algorithm would be employed to learn the ranking model, with the goal to predict the ground truth label in the training set as accurately as possible in terms of a loss function. In the test phase, when a new query comes in, the learned model is applied to rank the documents according to their relevance to the query. Depending on different hypotheses, input spaces, output spaces and loss functions, approaches to LtoR can be loosely grouped into three categories: point-wise, pairwise, and list-wise.

2.3. **Expertise ranking.** Expert ranking has been a promising research focus with the rapid development of on-line academic search engines, such as ArnetMiner and Microsoft Academic Search. Given a user query, the task of expert ranking basically involves identifying and ranking a list of researchers based on their expertise in that query-specific domain. Two categories of approaches have been focus of research in the past years: the pure content analysis based approach [1, 13, 22], which emphasizes evaluating authors' expertise by measuring the relevance between their associated documents and the query, and the social network based approach [9, 35, 39], which evaluates authors' expertise by exploiting the social interaction of authors and other scientific facets, such as their co-authorships, their citations to other papers/authors and more. Balog et al. [2] made a survey on the current main approaches for expertise retrieval, in which they more emphasized on summarizing the content-based approaches and divided them into probabilistic generative and discriminative model based approaches.

The topic modeling approach is one important group of probabilistic generative models for expert ranking. Typical works in this category include the models of CAT [36], ACT [34], ACTC [37], ALT [20] and ACVT [40]. However, none of them combine topic modeling with learning-to-rank approaches.

Fang et al. [13] proposed a probabilistic discriminative model for expert ranking, which is essentially a learning-to-rank method. Two other representative approaches using learning-to-rank for expert ranking include the work conducted by Moreira [26] and the work done by Macdonald and Ounis [23], both of which applied several existing learning-to-rank algorithms for ranking experts (bloggers). None of these models integrate the advantage of topic modeling though, and the latter two are applications of existing algorithms.

3. **Model Design.** This novel topic model we develop is a hierarchical probabilistic model, where each document is associated with attribute information. In this section, we first introduce the model where only pure contextual attributes, i.e., the words of the documents, are considered, and then further extend the model by incorporating additional features.

3.1. **Model description and generative process.** The model builds upon the previous works, including [7, 11], which extend the original LDA model by incorporating linkage between pairs of documents into topic modeling process. However, two characteristics distinguish our model from previous work. Firstly, we focus on modeling author interests and influence. Therefore, instead of modeling individual documents, we construct a virtual profile to represent each author (researcher) by concatenating all his/her publications. As a result, the topic proportion we derive for each virtual profile represents authors' distribution (authority) over topics. In the following part of the paper, we use document and virtual profile interchangeably. Secondly, we model the difference between pairs of author virtual profiles in terms of their topic distribution rather than the linkage information which measures the similarity between two connected documents.

We depict the graphical model of LtoRTM in Figure 1, which is a segment of the complete model consisting of only two connected virtual profiles. As indicated, it is a concatenation of two original LDA graphical plates, each of which represents one author virtual profile, connected by a binary variable indicator $y^c_{ai,aj}$, which represents the authority preference between authors a_i and a_j in community c. Note that we use d_i to represent the author virtual profile for author a_i.

Similar to the original LDA, each author virtual profile is represented by a plate, in which the shaded circle $\boldsymbol{w_d}$ is the observed data, representing each position-based word appearing in the profile, and the un-shaded circle $\boldsymbol{z}$ is the random variable representing the

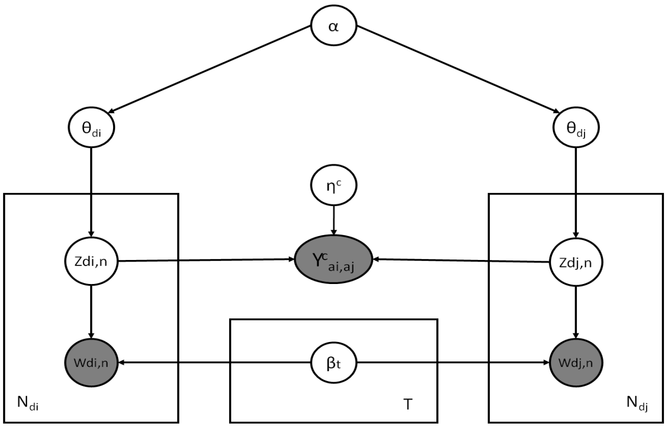


Figure 1. Graphical model for LtoRTM (d_i represents the author virtual profile of author a_i)

Table 1. Notation

Symbol	Size	Description
W	scalar	size of word vocabulary
D	scalar	number of author profiles
T	scalar	number of latent topics
N_{di}	scalar	the number of words in author profile d_i
N	scalar	the number of words in corpus
F	scalar	the number of features of authors
		Observed Data
$\boldsymbol{w_{di}}$	$\lvert\boldsymbol{w_{di}}\rvert$	the words lists of author profile d_i for author a_i
$\boldsymbol{w}$	N	the set of word tokens in corpus
$\boldsymbol{f_{ai}}$	F	feature set of author a_i
$y^c_{ai,aj}$		binary indicator
		Hyper-Parameters
α	$1 \times T$	Dirichlet prior for θ
η^c	$1 \times T$	coefficient
$\eta^c_1\ \eta^c_2$	$1 \times (T + \lvert F\rvert)$	coefficient
		Random Variables
θ	$A \times T$	distribution of authors over topics
β	$T \times V$	distribution of topics over words
$\boldsymbol{z_{di}}$	$1 \times T$	topic assignments for ith word in author profile d_i

topic assignment for one particular word. θ_d is a multinomial random variable, indicating the distribution of author virtual profile d over topics. β is global multinomial random variable, indicating the topic distribution over words in the whole corpus. Suppose that W, D, T are the number of distinct word (word vocabulary), the number of author virtual profiles and the number of topics respectively. We can represent θ as a $D \times T$ matrix, where each row represents one θ_d. Similarity, β can be represented as a $T \times W$ matrix. There also exists a T dimensional Dirichlet prior hyper-parameter α, which determines θ. Since our model is built upon the non-smoothed LDA, we do not introduce the Dirichlet prior for β. Additional details of the model parameters are illustrated in Table 1.

Given a collection of author virtual profiles, one essential target of our topic modeling is to discover the semantically coherent clusters of words (known as topics) to represent the profiles. Until now, we have introduced the model that can fulfill the task. Moreover, in order to model the authority preference over topics between author profiles, we further introduce a binary variable indicator $y^c_{ai,aj}$, named as the **binary preference indicator**, to indicate the authority preference between authors a_i and a_j. We have $y^c_{ai,aj} = 1$ if author a_i is believed to be more prestigious than author a_j in domain (community) c. This binary indicator is distributed according to a distribution that depends on the topic assignments for the two participating author profiles: d_i and d_j, and a domain (community)-specific regression parameter η^c.

The generative process of this model is divided into two periods, and can be described as follows:

- Stage 1: For each author virtual profile d:
  - Draw the topic proportion $\theta_d|\alpha \sim Dir(\alpha)$
  - For each word $w_{d,n}$ in profile d:
    - Draw the topic assignment $z_{d,n}|\theta_d \sim Multi(\theta_d)$
    - Draw word $w_{d,n}|z_{d,n}, \beta \sim Multi(\beta_{z_{d,n}})$
- Stage 2: For each pair of author profiles d_i and d_j with known preference:
  - Draw the binary preference indicator, satisfying:

$$y^c_{ai,aj}|\boldsymbol{z_{di}}, \boldsymbol{z_{dj}} \sim \psi(\cdot|\boldsymbol{z_{di}}, \boldsymbol{z_{dj}}, \eta^c) \tag{1}$$

where, $\boldsymbol{z_{di}} = z_{di,1}, z_{di,2}, \ldots, z_{di,n}$.

To note that $\boldsymbol{z_{di}}$ can be represented as a matrix, where each $z_{di,n}$ is a vector with only one element set to be 1 and the other elements set to be 0. It indicates the specific topic assignment for the nth word $w_{di,n}$ in author profile d_i.

ψ represents the distribution function that $y^c_{ai,aj}$ depends on. In order to model the difference in terms of authors' authority over topics, we assume that $y^c_{ai,aj}$ depends on the difference between $\boldsymbol{z_{di}}$ and $\boldsymbol{z_{dj}}$. In addition, since it is a binary indicator, we suppose that it follows the Bernoulli distribution, in which:

$$y^c_{ai,aj}|\boldsymbol{z_{di}}, \boldsymbol{z_{dj}}, \eta^c, \upsilon^c \sim Bernoulli\left(\sigma\left(\eta^T_c(\overline{\boldsymbol{z}}_{di} - \overline{\boldsymbol{z}}_{dj}) + \upsilon^c\right)\right)$$

in which, $\sigma(\cdot)$ is the sigmoid function. This function models each per-pair binary variable $y^c_{ai,aj}$ as a logistic regression with hidden co-variates, parametrized by coefficient η^c and the intercept υ^c. We further represent the original matrix $\boldsymbol{z_{di}}$ as a T dimensional vector $\overline{\boldsymbol{z}}_{di}$, where $\overline{\boldsymbol{z}}_{di} = \frac{1}{N_{di}} \sum_{n=1}^{n=N_{di}} z_{di,n}$.

3.2. **Incorporating features.** In the model we introduced in the previous section, authors' different preferences over topics are only determined by their associated contextual information, i.e., the papers they have published. As we can see from the generative process of the model, the binary preference indicator only depends on authors' topic assignments which are derived from author profiles. However, to measure an author's authority is a complicated process, as authors' expertise is not only determined by the papers they have written, but also by several other factors, such as their collaboration with other researchers, the influence of their published works, and some temporal characteristics of the authors, such as, how many years they have devoted to research, and how frequently they publish. To better model how authors' authority is differentiated, we extend the model we proposed in the previous section by introducing an additional factor representing features.

3.2.1. *LtoRTM with features.* We depict the extended graphical model of LtoRTM in Figure 2. We name it as the **LtoRTMF** model. As indicated, we represent each author a by an oval, in which, the author's virtual profile generated by the concatenation of his/her publications is still represented by a plate. In addition to that, we introduce a shaded circle $\boldsymbol{f_{ai}}$ to represent the features associated with author a_i. Features are assumed to

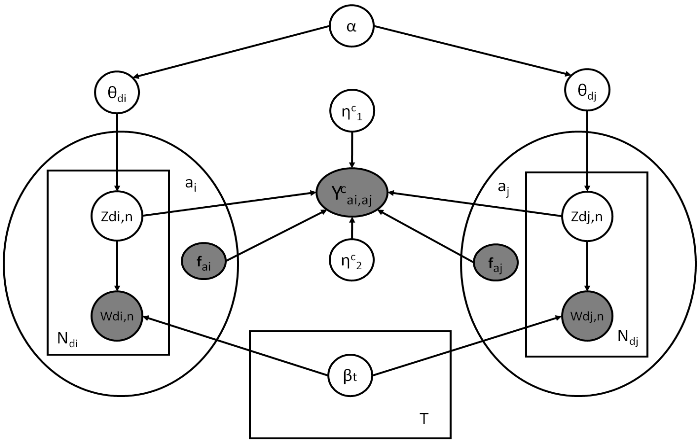


FIGURE 2. Graphical model for LtoRTMF (d_i represents the author virtual profile of author a_i; $\boldsymbol{f_{ai}}$ represents the author feature set of author a_i)

be observed data. Under this scheme, the authority preference between authors a_i and a_j is not only determined by the topic assignments of their virtual contextual profiles, but jointly determined by both the content information and additional features. Correspondingly, we introduce two coefficients: η_1^c, a T dimensional vector, is the regression parameter for topic assignment $\boldsymbol{z}$, and η_2^c is the regression parameter for feature set. The size of η_2^c would be determined by the number of features we identify. Now, the binary preference indicator $y_{ai,aj}^c$ would be determined by following the distribution as:

$$\begin{aligned} & y_{ai,aj}^c | \boldsymbol{z_{di}}, \boldsymbol{z_{dj}}, \boldsymbol{f_{ai}}, \boldsymbol{f_{aj}}, \eta_1^c, \eta_2^c \\ & \quad \sim \ \ Bernoulli \left(\sigma \left(\eta_{c1}^T \left(\overline{\boldsymbol{z}}_{di} - \overline{\boldsymbol{z}}_{dj}\right) + \eta_{c2}^T \left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right) + \upsilon^c\right)\right) \end{aligned}$$

3.2.2. *Features.* To represent authors' (researchers') authority, we identify several groups of features, each of which measures the expertise of an author from one aspect. Generally speaking, the features we consider reflect the overall expertise of an author, for example, the total number of publications of an author, as well as his/her expertise in a specific domain or community, for example, the author's number of publications in one domain. The whole feature set can be divided into four groups: 1) content profile based features; 2) simple bibliographic based features; 3) network based features; 4) temporal features.

Content profile based features: Even though we directly model the contextual virtual profile of an author by discovering its coherent clusters of words and representing it by a distribution over topics, we are also interested in measuring the content profiles by other widely-used IR metrics. Here we compute the traditional BM25 score of each author virtual profile, as well as the relevance score using standard language models. Both these two features are domain-based.

Simple bibliographic based features: We adopt a set of simple bibliographic features. These include:

total publication number (totalPubNo): which indicates the total number of publications of one author, across different research domains.

total citation number (totalCitNo): which indicates the total number of citations an author received from other papers published in different domains.

H-index [15]: H-index is the most well-known measurement in evaluating a researcher's expertise. A researcher is said to have an H-index with size h if h of his or her total papers has at least h citations each. This index is affected by the number of citation that a researcher has and the citation distribution among a researcher's various papers.

G-index [12]: G-index is another primarily used measurement. The G-index value is the highest integer (g) such that all the papers ranked in Position 1 to g in terms of their citation number have a combined number of citations of at least g^2.

Rational H-index distance (HD-index) [32]: this variant of H-index calculates the number of citations that are needed to increase the H-index by 1 point.

Rational H-index X (HX-index) [32]: the original H-index indicates the largest number of papers an author has with at least h citations. However, a researcher may have more than h papers, for example, n papers, that have at least h citations. If we define $x = n - h$, then the HX-index is calculated by $HX = h + x(s - h)$, where s is the total number of publications an author has.

E-index [43]: the original H-index only concentrates on the set of papers an author published, each of which has at least h citations. This set of papers is often referred to as the *h-core* papers of an author. By using this measurement, the only citation information that can be retrieved is h^2, i.e., at least h^2 citations of an author can be received. However, the additional citation for papers is the *h-core* which would be completely ignored. To complement the H-index for the ignored excess citations, E-index is proposed, which can be computed by $e^2 = \sum_{j=1}^{h}(cit_j - h) = \sum_{j=1}^{h} cit_j - h^2$, where cit_j are the citations received by the jth paper in the *h-core* set. We can further have E-index $= sqrt(e^2)$.

Individual H-index IH-index [3]: this measurement is proposed to reduce the effects of co-authorship. It can be computed by dividing the standard H-index by the average number of authors in the *h-core* set: IH-index $= h^2/N_a^T$, where N_a^T is the total number of authors in *h-core* set.

Normalized Individual H-index NIH-index [14]: this measurement is also proposed to reduce the coauthor's effect, but is much finer-grained than the previous one. To compute it, we can firstly normalize the number of citations for each paper in the *h-core* by dividing the number of its citation by its number of authors. Then we compute the H-index score based on these normalized citation counts.

It is noticeable to mention that we calculate all the features mentioned above from all its publications, as well as only those publications from a specific research domain. For example, we can compute the overall H-index of an author, by doing that, all the papers written by that author would be considered. However, when computing the H-index of an author in a specific domain c, we would only consider those papers published in that domain, and compute its citations only based on other papers that are also from that domain.

Network based features: This group of features measure how well an author collaborates with other authors, and how their publications influence other authors. We construct two types of network, and apply the PageRank algorithm to compute the authors' authority scores. The networks we considered are:

Coauthor Network: this network is generated by connecting authors by their coauthor-relationships. For the sake of PageRank algorithm, we convert one non-directional edge into two directional edges. As a result, one non-weighted edge would exist from author a_i to author a_j and from author a_j to author a_i if they have written at least one paper together.

Citation Network: this directed network is generated by connecting authors by their citations. One non-weighted edge would point from author a_i to a_j if at least one publication of author a_i cites one paper of author a_j.

We also generate such two kinds of networks for each research community we considered.

Temporal features: This group of features measures authors' authority by some temporal characteristics associated with them. These include:

CareerTime [41]: this measures how long a researcher has devoted to academic research. We assume that the longer career time a researcher has, the higher authority he may have.

LastRestTime [41]: this indicates how many years have passed since the last publication of a researcher. We assume that a long time rest without academic output will negatively affect a researcher's academic reputation.

PubInterval [41]: this measures how many years on average would a researcher take between every two consecutive publications. We assume that more frequent publication indicates more active academic participation.

Citation Influence ratio [41]: we define and consider one other temporal factor which tests the long time influence of a researcher's publication, and thus indirectly represents the influence of the researcher. We assume that if a paper continues to be cited a long time after its publication, it brings higher prestige to its author (e.g., the paper PageRank [29] is frequently and persistently cited by the following papers). To model this temporal factor, we first introduce a decay function to differentiate the weight between a pair of paper citations. If paper p_j published in year y_j cites another paper p_i published in year y_i, $(y_j - y_i) \geq 0$, we define a probability as the *citation influence ratio* of paper p_j on p_i as: $CIR(p_{ji}) = \beta_1 \left(1 - \beta_2^{y_j - y_i}\right)$, where β_2 $(0 < \beta_2 < 1)$ is the decay base. We now define the *citation influence* between a pair of authors as: $CI(a_{ji}) = \sum CIR(p_{ji})$, where p_j is any paper of author a_j, p_i is any paper of a_i, and p_j cites p_i.

Contemporary H-index CH-index [33]: this index adds an age-related weighting to each paper. The basic assumption is that the older the paper is, the less the weight is. The new citation count for each paper of an author can be computed as $S^c(i) = \gamma \times (Y(now) - Y(i) + 1)^{-\delta} \times |C(i)|$, where $Y(i)$ is the year when paper i is published, and $|C(i)|$ is the set of paper citing paper i. In computation, δ is often set to be 1, and γ is set to be 4. After computing this new citation count for each paper, we can compute the H-index as the standard one based on the new citation count of each paper.

AR-index [18]: it is also an age-weighted index. The citation count of each paper would be divided by the age of that paper, and then the AR-index is the square root of the sum of all the papers in the *h-core* of an author.

AWCR-index [14]: this is basically the same as the AR-index, but it sums over the weighted citation count of all the papers of an author rather than only the papers in the *h-core* set.

AvgPubNo: this is computed by dividing the total publication number of an author by the $CareerTime$ of this author.

AvgCiteNo: this is computed by dividing the total number of citations of an author by his/her $CareerTime$.

These features are also computed either based on all publications across domains or on those domain-specific publications. Overall, we have identified 42 distinct features.

4. **Model Inference, Estimation and Ranking Scheme.** In this section, we introduce how to solve the generative topic models we proposed in the previous sections, which includes the model inference for 1) topic assignment ($\boldsymbol{z}$), 2) θ (virtual-profile-topic distribution), and 3) β (the topic-word distribution), as well as the parameter estimations for 1) α (the Dirichlet prior) and 2) η^c (the regression coefficient). Based on the variables and parameters learned from the training set, we also introduce how to achieve the topic assignment and topic proportions for new testing author profiles, and how to rank them.

4.1. **Inference and estimation.** Given a collection of author virtual profiles $\boldsymbol{D}$, in order to solve the topic model as we proposed, we would like to find parameters α, β, η^c, that can maximize the (marginal) log likelihood of the data:

$$\begin{aligned} l\left(\alpha,\beta,\eta^c\right) &= \log\left(p\left(\boldsymbol{W},\boldsymbol{Y}|\alpha,\beta,\eta^c\right)\right) \\ &= \log\left(\left[\prod_{d:1\to D} p(\boldsymbol{w}|\alpha,\beta)\right]\left[\prod_{(di,dj)\in\boldsymbol{E}} p(y^c_{ai,aj}|\eta^c)\right]\right) \\ &= \log\left(\prod_{d=1}^{D}\int p(\theta|\alpha)\left(\prod_{n=1}^{Nd}\sum_{z_{d,n}} p(z_{d,n}|\theta_d)p(w_{d,n}|z_{d,n},\beta)\right)d\theta \right. \\ &\quad \left. \times \prod_{(di,dj)\in\boldsymbol{E}}\sum_{\overline{\boldsymbol{z}}_{di}}\sum_{\overline{\boldsymbol{z}}_{dj}} p(y^c_{ai,aj}|\overline{\boldsymbol{z}}_{di},\overline{\boldsymbol{z}}_{dj},\eta^c)\right) \end{aligned}$$

where, we denote $\boldsymbol{E}$ as the set of pairs of author profiles with known preferences. In our model, we would only model those pairs of author profiles with explicitly known preferences.

However, to maximize such log likelihood is intractable due to the problematic coupling between θ and β, which is caused by the existing edges between θ, z and β. Even though exact inference is intractable, there exist a wide variety of approximate inference algorithms, including variational inference [6], expectation propagation [25], and Markov chain Monte Carlo (MCMC) schemes [16]. In our work, we make use of the variational inference for approximating the posterior inference, and apply this procedure in a variational EM algorithm for parameter estimation.

The basic idea of variational inference is to make use of the Jensen's inequality to obtain an adjustable lower bound on the log likelihood. A simple way to obtain a tractable family of lower bounds is to consider simple modifications of the original graphical model in which some of the edges and nodes are removed, and the resulting graphical model is endowed with free variational parameters as follows in Equation (2):

$$q(\theta,\boldsymbol{z}|\gamma,\phi) = q(\theta|\gamma)\prod_{n=1}^{N} q(z_n|\phi_n) \tag{2}$$

where, γ and ϕ are two free variational parameters. γ is a Dirichlet parameter, which similar to θ, can be represented by a $D\times T$ matrix; and ϕ is a multinomial parameter, which similar to $\boldsymbol{z}$, can also be represented as of $D\times N\times T$ tensor, where D is the number of author profiles in corpus, N is the number of position-based word tokens, and T is the number of pre-defined topics. Note that, $E_q[z_{d,n}] = \phi_{d,n}$.

With γ and ϕ, and integrating over the two random variables θ and z, the log of the marginal probability can be represented as:

$$\begin{aligned} \log(p(w,y|\alpha,\beta,\eta^c)) &= \log\left(\int\sum_z p(w,y,\theta,z|\alpha,\beta,\eta^c)d\theta\right) \\ &= \log\left(\int\sum_z \frac{p(w,y,\theta,z|\alpha,\beta,\eta^c)q(\theta,z)}{q(\theta,z)}d\theta\right) \end{aligned}$$

According to Jensen's inequality $\log(E(a)) \geq E(\log(a))$, we can further have:

$$\begin{aligned} \log\left(E_q\left[\frac{p(w,y,\theta,z|\alpha,\beta,\eta^c)}{q(\theta,z)}\right]\right) &\geq E_q\left[\log\left(\frac{p(w,y,\theta,z|\alpha,\beta,\eta^c)}{q(\theta,z)}\right)\right] \\ &= E_q[\log(p(w,y,\theta,z|\alpha,\beta,\eta^c))] - E_q[\log(q(\theta,z))] \end{aligned}$$

This is the lower bound of the original log likelihood, and is the goal probability we need to maximize.

To denote $E_q[\log(p(w, y, \theta, z|\alpha, \beta, \eta^c))] - E_q[\log(q(\theta, z))]$ as $L(\gamma, \phi; \alpha, \beta, \eta^c)$, we can expand it as:

$$\begin{aligned} L(\gamma, \phi; \alpha, \beta, \eta^c) = & \sum_{(di,dj)\in \boldsymbol{E}} E_q[\log(p(y^c_{ai,aj}|\overline{\boldsymbol{z}}_{di}, \overline{\boldsymbol{z}}_{dj}, \eta^c))] \\ & + \sum_d E_q[\log(p(\theta_d|\alpha))] + \sum_d \sum_z E_q[\log(p(z_{d,n}|\theta_d))] \\ & + \sum_d \sum_z E_q[\log(p(w_{d,n}|z_{d,n}, \beta))] \\ & - E_q[\log(q(\theta|\gamma))] - E_q[\log(q(z|\phi))] \end{aligned}$$

Each element on the right-hand side of the above equation can be further expanded. However, due to space limit, we only concentrate on the expansion of the first element, which represents the primary contribution of our model, and leave the expansion of the rest of the elements to readers for reviewing references in [6].

In our *LtoRTM* model, $y^c_{ai,aj}$ follows the Bernoulli distribution, taking η^c, $\boldsymbol{z_{di}}$, $\boldsymbol{z_{dj}}$ as parameters. In the extended *LtoRTMF* model, it further depends on the feature set of authors: $\boldsymbol{f_{ai}}$, $\boldsymbol{f_{aj}}$.

By representing Bernoulli distribution as a generalized linear model, we can have in the *LtoRTM* model, the probability:

$$p\left(y^c_{ai,aj}|\overline{\boldsymbol{z}}_{di}, \overline{\boldsymbol{z}}_{dj}, \eta^c\right) = \exp\left\{y^c_{ai,aj}\eta^T_c(\overline{\boldsymbol{z}}_{di} - \overline{\boldsymbol{z}}_{dj}) - \log\left(1 + \exp\left(\eta^T_c(\overline{\boldsymbol{z}}_{di} - \overline{\boldsymbol{z}}_{dj})\right)\right)\right\} \quad (3)$$

and in the *LtoRTMF* model:

$$\begin{aligned} p\left(y^c_{ai,aj}|\overline{\boldsymbol{z}}_{di}, \overline{\boldsymbol{z}}_{dj}, \overline{\boldsymbol{f}}_{ai}, \overline{\boldsymbol{f}}_{aj}, \eta^c_1, \eta^c_2\right) = & \exp\left\{y^c_{ai,aj}\left(\eta^T_{c1}(\overline{\boldsymbol{z}}_{di} - \overline{\boldsymbol{z}}_{dj}) + \eta^T_{c2}\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right)\right)\right. \\ & \left. - \log\left(1 + \exp\left(\eta^T_{c1}(\overline{\boldsymbol{z}}_{di} - \overline{\boldsymbol{z}}_{dj}) + \eta^T_{c2}\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right)\right)\right)\right\} \end{aligned}$$

By taking log of the probability, and using first-order approximation to compute their expectations, we can finally have: in the *LtoRTM* model:

$$E[\log(p(y^c_{ai,aj}|\overline{\boldsymbol{z}}_{di}, \overline{\boldsymbol{z}}_{dj}, \eta^c))] = y^c_{ai,aj}\eta^T_c\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) - \log\left(1 + \exp\left(\eta^T_c\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right)\right)\right) \quad (4)$$

and in the *LtoRTMF* model:

$$\begin{aligned} & E\left[\log\left(p\left(y^c_{ai,aj}|\overline{\boldsymbol{z}}_{di}, \overline{\boldsymbol{z}}_{dj}, \overline{\boldsymbol{f}}_{ai}, \overline{\boldsymbol{f}}_{aj}, \eta^c_1, \eta^c_2\right)\right)\right] \\ = & \, y^c_{ai,aj}\left(\eta^T_{c1}\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) + \eta^T_{c2}\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right)\right) \\ & - \log\left(1 + \exp\left(\eta^T_{c1}\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) + \eta^T_{c2}\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right)\right)\right) \end{aligned}$$

We have until now expanded $L(\gamma, \phi; \alpha, \beta, \eta^c)$. We then show how to maximize L with respect to ϕ, γ, α, β and η^c.

Inferring ϕ

To maximize L with respect to ϕ, we can collect the terms associated with ϕ. Since $y^c_{ai,aj}$ depends on the difference between $\boldsymbol{z_{di}}$ and $\boldsymbol{z_{dj}}$, which have been represented by ϕ_{di} and ϕ_{dj}, we need to take derivatives with respect to ϕ_{di} and ϕ_{dj} respectively.

In the *LtoRTM* model, we have

$$\begin{aligned} \phi_{di,n} \propto & \log \beta_{\cdot}, w_{di,n} + \Gamma(\gamma_{di}) - \mathbf{1}\Gamma\left(\mathbf{1}^T\gamma_{di}\right) \\ & + \sum_{(di,dj)\in \boldsymbol{E}} \left(\frac{y^c_{ai,aj}}{N_{di}}\eta^T_c - \frac{\eta^T_c}{N_{di}} \frac{\exp\left\{\eta^T_c\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right)\right\}}{1 + \exp\left\{\eta^T_c\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right)\right\}}\right) \end{aligned}$$

$$\begin{aligned}\phi_{dj,n} \propto{} & \log \beta\cdot, w_{dj,n} + \Gamma(\gamma_{dj}) - \mathbf{1}\Gamma\left(\mathbf{1}^T\gamma_{dj}\right) \\ & - \sum_{(di,dj)\in \boldsymbol{E}} \left(\frac{y^c_{ai,aj}}{N_{dj}}\eta_c^T + \frac{\eta_c^T}{N_{dj}} \frac{\exp\left\{\eta_c^T\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right)\right\}}{1+\exp\left\{\eta_c^T\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right)\right\}} \right)\end{aligned}$$

where,

$$\overline{\boldsymbol{\phi}}_{di} = \frac{1}{N_{di}}\sum_n \phi_{di,n} \tag{5}$$

and in the *LtoRTMF* model with additional features, we have:

$$\begin{aligned}\phi_{di,n} \propto{} & \log \beta\cdot, w_{di,n} + \Gamma(\gamma_{di}) - \mathbf{1}\Gamma\left(\mathbf{1}^T\gamma_{di}\right) \\ & + \sum_{(di,dj)\in \boldsymbol{E}} \left(\frac{y^c_{ai,aj}}{N_{di}}\eta_{c1}^T - \frac{\eta_{c1}^T}{N_{di}} \frac{\exp\left\{\eta_{c1}^T\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) + \eta_{c2}^T\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right)\right\}}{1+\exp\left\{\eta_{c1}^T\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) + \eta_{c2}^T\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right)\right\}} \right)\end{aligned}$$

$$\begin{aligned}\phi_{dj,n} \propto{} & \log \beta\cdot, w_{dj,n} + \Gamma(\gamma_{dj}) - \mathbf{1}\Gamma\left(\mathbf{1}^T\gamma_{dj}\right) \\ & - \sum_{(di,dj)\in \boldsymbol{E}} \left(\frac{y^c_{ai,aj}}{N_{dj}}\eta_{c1}^T + \frac{\eta_{c1}^T}{N_{dj}} \frac{\exp\left\{\eta_{c1}^T\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) + \eta_{c2}^T\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right)\right\}}{1+\exp\left\{\eta_{c1}^T\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) + \eta_{c2}^T\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right)\right\}} \right)\end{aligned}$$

Inferring η

In the *LtoRTM* model,

$$\frac{\partial \mathcal{L}}{\partial \eta^c} = \sum_{(di,dj)\in \boldsymbol{E}} \left(y^c_{ai,aj}\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) - \left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) \frac{\exp\left\{\eta_c^T\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right)\right\}}{1+\exp\left\{\eta_c^T\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right)\right\}} \right)$$

and in the *LtoRTMF* model, where we consider two coefficients η^c_1 and η^c_2, we have:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \eta^c_1} = {} & \sum_{(di,dj)\in \boldsymbol{E}} \Bigg(y^c_{ai,aj}\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) \\ & - \left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) \frac{\exp\left\{\eta_{c1}^T\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) + \eta_{c2}^T\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right)\right\}}{1+\exp\left\{\eta_{c1}^T\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) + \eta_{c2}^T\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right)\right\}} \Bigg)\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \eta^c_2} = {} & \sum_{(di,dj)\in \boldsymbol{E}} \Bigg(y^c_{ai,aj}\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right) \\ & - \left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right) \frac{\exp\left\{\eta_{c1}^T\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) + \eta_{c2}^T\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right)\right\}}{1+\exp\left\{\eta_{c1}^T\left(\overline{\boldsymbol{\phi}}_{di} - \overline{\boldsymbol{\phi}}_{dj}\right) + \eta_{c2}^T\left(\overline{\boldsymbol{f}}_{ai} - \overline{\boldsymbol{f}}_{aj}\right)\right\}} \Bigg)\end{aligned}$$

We leave the updating rule for α, β and γ for readers' reference, since they are the same as the original LDA model [6].

4.2. **Ranking scheme.** In the training process, we have approximated the posterior distribution of γ (representing θ), ϕ (representing the topic assignments $\boldsymbol{z_d}$), β, as well as α and η^c. In the testing phase, a set of new author virtual profiles would be given. The words in those profiles are the observed data, but we would not know the preference between every pair of the profiles. In the testing phase, the α, η^c and β variables would be regarded as the known parameters, as their values have been estimated during the training process. As a result, what we need to approximate for the new author profiles are 1) the topic assignments for their word tokens (θ or in variational inference, the γ), and 2) the author-profile-topic distributions $\boldsymbol{z}$ or in variational inference, the ϕ:

$$p\left(\gamma, \phi | D^{test}, \alpha, \beta, \eta^{c}\right) \tag{6}$$

Without incorporating the pair-wise preference information between author profiles, our model would retreat to the original LDA model. We would, therefore, skip the detailed description on parameter inference for testing in this paper. Readers can find the inference process as introduced in paper [6].

After approximating the γ and the ϕ variables for author profiles in testing set, we can compute the authority score of each author a_i (represented by his/her author profile d_i) and rank them by:

$$P(a_i|c) = P(d_i|c) = \eta_c^T \overline{\boldsymbol{\phi}}_{di} \tag{7}$$

or, with additional features:

$$P(a_i|c) = \eta_{c1}^T \overline{\boldsymbol{\phi}}_{di} + \eta_{c2}^T \boldsymbol{f_{ai}} \tag{8}$$

5. Experimental Evaluation.

To demonstrate the effectiveness of our LtoRTM and LtoRTMF model, we conducted experimental studies comparing them with several state-of-the-art topic models and learning-to-rank algorithms. Particularly, we apply our model into two applications, which evaluate the expertise of researchers from two aspects: the prediction of SIG-community award winners and the prediction of PC members of the main conference of several research communities.

5.1. Experiments setup.

5.1.1. **Data set.** The experiments were carried out over two real world data sets. The first data set is a subset of the **ACM Digital Library**, from which we crawled one descriptive web page for each 172,890 distinct papers having both title and abstract information. For each published paper, we extracted the information about its authors and references. Due to possible author names' ambiguity, we represent each candidate author name by a concatenation of the first name and last name, while removing all the middle names. We then use exact match to merge candidate author names. Finally, we obtain 170,897 distinct authors, and 2097 venues.

The second data set we utilized is the data set 'DBLP-Citation-network V5' provided by Tsinghua University for their ArnetMiner academic search engine [35]. This data set is the crawling result from the ArnetMiner search engine on Feb. 21st, 2011 and further combined with the citation information from ACM. We name this data set as the **ArnetMiner dataset**. The original data set is reported to have 1,572,277 papers and to include 2,084,019 citation-relationships. After carrying out the same data processing method as we did for the ACM data set, we find 1,572,277 papers, 795,385 authors and 6010 venues.

For papers in each data set, we filter out the stop words in paper content, and collect the words that appear more than 10 times in the entire corpus. We finally retrieve 43,748 and 107,576 distinct words for ACM and ArnetMiner data sets respectively.

5.1.2. **Research domain identification.** To identify a research community, we first manually cluster papers into different domains, and further group their associated authors. We choose six research communities as our targeting communities (see Table 2). For each such research community, we collected and merged the Top 20 venues identified by the MSRA academic search engine[2] and ArnetMiner search engine[3] for that research community respectively. Papers that are published in those venues are considered to be

[2] http://research.microsoft.com/en-us/projects/academic/

[3] http://arnetminer.org/

TABLE 2. Community, query and award winners ground truth. Numbers outside of the parentheses or in the parentheses indicate the number of winners available in ACM and ArnetMiner data set respectively.

Community	Corresponding query	SIG award winners
		(1990-2009)
sigarch	hardware architecture	27(27)
sigsoft	software engineering	15(15)
sigkdd	data mining	7(7)
sigir	information retrieval	9(9)
sigcomm	network communication	18(18)
sigmod	database	18(18)

domain-specific papers of that community, and the authors of these papers are considered to be the domain-specific authors of that community. We collect the domain-specific features based on the domains we identified.

6. **Experiments Methodology and Result.**

6.1. **Application.**

6.1.1. *Task description and ground truth generation.* Both LotRTM and LtoRTMF are especially designed for modeling author's authority (interests or influence). In this paper, we focus on two applications that are closely related to expert ranking: predicting future award winners of a specific research community (the ACM SIG community), and predicting PC members of a main conference in research domain. We choose these two applications for two reasons: 1) they evaluate the expertise of a researcher from two different points of view; 2) we can retrieve excellent objective ground truths for both of them, and therefore can avoid human labeling which is assumed to be biased and subjective.

Award Winner Prediction: Each year, in many ACM SIG communities, some outstanding researchers will be granted an award in honor of his or her profound impact and numerous research contributions. For example, in 2012, Prof. Norbert Fuhr has been granted the 'Salton Award' in 'SIGIR' community for his 'pioneering, sustained, and continuing contributions to the theoretical foundations of information retrieval and database systems'.

It would be an interesting research task to predict the future award winners given historical information. To be more specific, the task of predicting award winners can be described as: *Given a specific research community c, and all its historical award winners before year* $Y1$*, can we successfully predict its award winner on year* $Y1$*?* Normally, only one researcher would be granted the award each year.

From the ACM SIG official web site, we selected six SIG communities, and collected their historical award winners from 1990 to 2009, out of which, 2000-2009 is the period of time that we intend to predict. We generate the corresponding query for each community based on the main research area of that community, for example, the query for SIGIR community is 'information retrieval'. We also check the generated queries with the 23 categories provided by Microsoft Academic search engine, and make sure that each query corresponds to one category. We assume that these queries cover the main disciplines of computer science research, and that they represent reasonable topics that users might use for information. These queries are intended to be broad queries. More detailed information on the chosen communities, their queries, and number of historical award winners is reported in Table 2. We set the number of topics to be 20 for this task.

Table 3. Community, conference, and PC member ground truth

Community	Conference	Years				
KDD	kdd	2000	2001	2002	2003	2004
		55(57)	74(78)	73(78)	113(116)	124(127)
		2005	2006	2007	2008	2009
		129(130)	178(184)	210(219)	235(241)	230(247)
IR	sigir	2000	2001	2002	2003	2004
		78(81)	41(43)	189(197)	38(38)	33(33)
		2005	2006	2007	2008	2009
		24(24)	114(114)	352(367)	365(381)	569(590)
MOD	sigmod	2000	2001	2002	2003	2004
		14(14)	52(52)	65(65)	102(103)	136(136)
		2005	2006	2007	2008	2009
		135(140)	42(44)	4(4)	126(128)	126(129)

Conference PC Member Prediction: Working as a PC member of the main conference in a research community is an important indicator of a researcher's expertise. This task of PC member prediction can be described as *Given a conference (representing a research community c), and all its PC members before year $Y1$, can we successfully predict its PC members on year $Y1$?*

For three SIG communities (SIGKDD, SIGIR, SIGMOD), we choose one main conference for each of them as our targeting conference, and collect its PC members from its official website between 2000 and 2009. 2005-2009 is the period of time that we intend to predict. Table 3 shows the community, the chosen conferences, as well as the number of PC members (also in our data corpus) for that conference between 2000-2009. For this task, we set the number of topics to be 10.

6.1.2. *Training and testing set generation.* Both the training and testing sets are generated on per-community and per-year basis. Since we have few positive samples, as compared to a much larger set of negative samples, we pre-set a pos-neg ratio λ to randomly select negative samples. The process of generating the training set is as follows: suppose we intend to predict the award winner (or PC member) for community SIGKDD on year Y_i, we retrieve and regard all award winners (or PC members) of SIGKDD on year Y_j ($1990 \leq Y_j \leq Y_i - 1$) as positive samples, and for each positive sample, we randomly choose λ times other authors which are not SIGKDD award winners (or PC members) on year Y_j. Such a process would be repeated 100 times, and all positive and negative samples would then form the training set of community SIGKDD on year Y_i. λ can be a tuned parameter, and in our current experiments, we set it to be 2.

For generating the testing set, for each community c on year Y_i, we would retrieve the Top 1000 authors in terms of their in-domain(c) publication number as the testing set. We have also tried to generate the testing set by retrieving the Top 1000 authors in terms of their BM25 scores or a pool list of the merged Top 200 authors across all features; however, working on testing samples retrieved by their in-domain publication number gives the best performance.

6.1.3. *Baseline algorithms.* We compare the performance of our proposed models with four baseline algorithms: RankSVM, AdaRank, Supervised LDA and Coordinate Ascent, all of which are state-of-the-art algorithms for either learning-to-rank or topic modeling.

RankSVM (rSVM) [19] is a pair-wise learning-to-rank algorithm, which borrows the idea of SVM and therefore is designed to maximize the margin between positively and negatively labeled documents in the training set by minimizing the number of discordant pairs. Its learning task can be defined as the following quadratic programming problem.

$$\min_{\omega,\xi_{q,i,j}} \frac{1}{2}\|\omega\|^2 + c\sum_{q,i,j}\xi_{q,i,j} \quad \text{subject to}$$

$$\omega^T X_i^q \geq \omega^T X_j^q + 1 - \xi_{q,i,j},$$

$$\forall X_i^q \succ X_j^q,\ \xi_{q,i,j} \geq 0$$

where X_i^q represents the query-document feature vectors for document i. $X_i^q \succ X_j^q$ implies that document i is ranked higher than document X_j^q with respect to query q in the training set. $\xi_{q,i,j}$ denotes the non-negative slack variable. c is the parameter determining the trade-off between the training error and margin size. $\|\omega\|^2$ represents the structural loss.

AdaRank [38] is a list-wise learning-to-rank algorithm. Instead of training ranking models by minimizing the loss function loosely related to the performance measures (e.g., minimizing classification error on instance pairs), AdaRank is proposed to minimize the loss function directly defined on the performance measures (i.e., MAP, MRR, NDCG) by repeatedly constructing 'weak rankers' on the basis of re-weighted training data, and finally linearly combines the learned weak rankers to make predictions over testing data.

Supervised LDA (sLDA) [5] extends the original LDA model by adding a response variable connected to each document. Its ultimate goal, correspondingly, is to infer the latent topic structure of an unlabeled document, and then generate a prediction of its response. Supervised LDA is especially designed for applications like predicting the ratings of movie reviews and the category of a document. Even though it is also a supervised learning algorithm, it does not explore the difference between every pair of documents. The response is only determined by the topic assignment of individual document.

Coordinate Ascent [24] (CA for short) is another list-wise learning-to-rank algorithm directly targeting at optimizing the performance measure. Its basic idea is to iteratively optimize a multivariate objective function by solving a series of one dimensional optimization. In each iteration, one single feature will be randomly chosen to be optimized and all other features will be kept the same as in the last round.

For all four baselines, we feed them the same training data and testing data as we generated for running our LtoRTM and LtoRTMF model. We choose the average rank (*avgRank*) and *MAP* as the evaluation metric for predicting award winners and PC members respectively.

6.1.4. *Prediction results.* Tables 4 and 5 show the results of predicting award winners, as compared with the baseline algorithms, in both ACM and ArnetMiner data sets respectively. We show the avgRank for each community as well as the overall average rank across communities.

Predicting Award Winners We test RankSVM and Coordinate Ascent with pure content as well as additional features. For sLDA, we only work on word count features. AdaRank applies a different learning mechanism, where we took each of the 42 distinct features as one 'weak learner'. There is no word count information used in AdaRank algorithm. Several observations can be made from the results in Tables 4 and 5:

- The results are consistent across the two data sets; however, results are very sensitive to individual communities, as models that work well in some communities do not perform well in others.
- RankSVM still performs the best in terms of overall performance; however, this is not always true looking at individual communities. Our model can outperform RankSVM in 6 out of 14 individual cases (considering 6 communities with either

content-based features results or results with additional features incorporated plus two overall performances).

- Our model is the second best model since it works better than sLDA, AdaRank and Coordinate Ascent under most circumstances.
- Incorporating features cannot guarantee improved performance on individual communities. This is true not only for our models, but also for RankSVM model. Taking our models (LtoRTM and LtoRTMF) as the example, for six communities in both the ACM and ArnetMiner data set, three of them (sigkdd, sigcomm and simod community for the ACM data set, and sigarch, sigsoft and sigmod community for the ArnetMiner data set) can have their performance enhanced with additional features. However, the overall performance can always be improved with additional features included.

Table 4. Award winner prediction: ACM avgRank

Algorithm	arch	soft	kdd	ir	comm	mod	Overall
RankSVM (C)	**35.0**	123.7	120.0	6.7	80.3	**49.3**	75.22
RankSVM (C+F)	41.4	121.1	119.0	**5.7**	48.6	49.7	**70.03**
AdaRank	43.7	201.1	161.0	36.7	113.2	78.6	113.19
sLDA (C)	137.7	126.2	98.5	42.3	**35.8**	129.4	104.5
CA (C)	132.37	107.47	106.47	158.50	107.70	102.29	119.13
CA (C+F)	89.71	115.75	120.85	161.77	112.60	109.41	118.35
LtoRTM	108.2	**95.7**	82.6	22.3	109.8	136.0	97.05
LtoRTMF	120.0	101.0	**81.7**	24.8	98.2	87.4	90.86

Table 5. Award winner prediction: ArnetMiner avgRank

Algorithm	arch	soft	kdd	ir	comm	mod	Overall
RankSVM (C)	**37.0**	122	138.0	**5.7**	**46.0**	49.7	69.67
RankSVM (C+F)	69.3	**56.3**	67.1	97.8	109.7	39.2	**63.89**
AdaRank	194.8	127.4	63.9	22.4	52.2	65.7	96.35
sLDA (C)	99.7	105.9	105.3	166.0	149.4	108.9	115.12
CA (C)	145.20	107.02	142.01	189.10	95.89	112.46	131.95
CA (C+F)	169.83	83.73	122.96	141.67	169.68	74.65	127.09
LtoRTM	141.9	76.2	**47.8**	117.3	91.4	128.4	103.31
LtoRTMF	118.5	74.9	48.2	138.9	204.4	**34.0**	91.21

Predicting PC Members Results on predicting PC members are reported in Tables 6 and 7 for ACM data set and ArnetMiner data set respectively. Several observations can be made as follows:

- Results are also sensitive to individual communities as well as to different data sets.
- For the ACM data set, we can see that RankSVM still works the best; our model is the second best model as it outperforms AdaRank and Coordinate Ascent and shows competitive results with sLDA.
- For the ArnetMiner data set, however, our models can outperform those of RankSVM in two individual cases (LtoRTM outperforms RankSVM (C) for sigir community, and LtoRTMF outperforms RankSVM (C+F) for sigkdd community). LtoRTMF is also superior to RankSVM (C+F) in overall performance.
- We can also observe that incorporating features does not provide performance improvement for all communities, but it does generate improved overall performance.

TABLE 6. PC member prediction: ACM MAP

Algorithm	sigkdd	sigir	sigmod	Overall
RankSVM (C)	0.5966	**0.5952**	**0.2303**	0.4740
RankSVM (C+F)	**0.6110**	0.5942	0.2267	**0.4773**
AdaRank	0.5997	0.2168	0.0261	0.2808
sLDA (C)	0.3358	0.4150	0.1814	0.3107
CA (C)	0.2974	0.4625	0.0851	0.2817
CA (C+F)	0.4611	0.2972	0.1558	0.3047
LtoRTM	0.3201	0.5146	0.0958	0.3102
LtoRTMF	0.4909	0.3372	0.1738	0.3340

TABLE 7. PC member prediction: ArnetMiner MAP

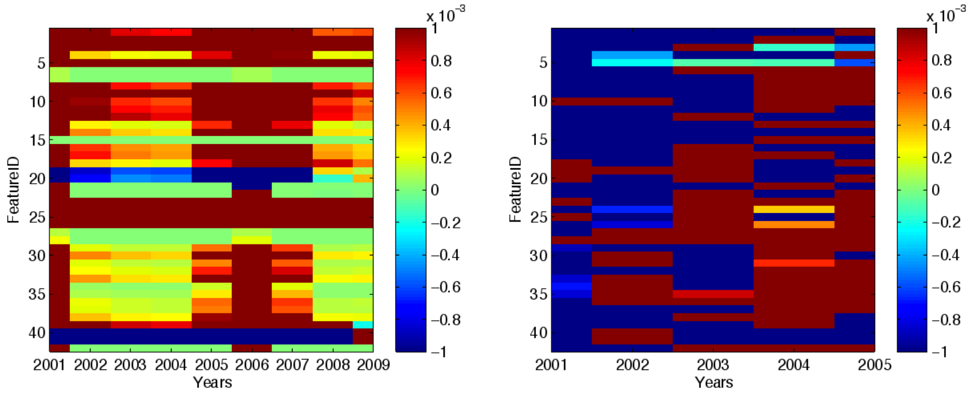
Algorithm	sigkdd	sigir	sigmod	Overall
RankSVM (C)	0.0692	0.0590	0.0479	0.0586
RankSVM (C+F)	0.0742	0.0632	**0.0513**	0.0629
AdaRank	0.1075	0.0411	0.0130	0.0539
sLDA (C)	0.0489	0.0809	0.0418	0.0571
CA (C)	0.0424	0.0416	0.0364	0.0401
CA (C+F)	0.0697	0.0495	0.0394	0.0529
LtoRTM	0.0496	**0.0821**	0.0424	0.0580
LtoRTMF	**0.1200**	0.0545	0.0393	**0.0712**

6.1.5. *Feature analysis.* In LtoRTMF model, η_2^c is the coefficient vector associated with the feature vector. By checking the coefficient value associated with each feature, we can determine its contribution (importance) to the overall performance. Figures 3(a) and 3(b) illustrate the results for predicting award winners and PC members of the SIGKDD community respectively. In both of these figures, we use different colors to represent features' importance. Compared with the right-side indicator bar, colors closer to '0' indicate less important features. Colors with corresponding values greater than 0 indicate positive correlations, and colors with corresponding values less than 0 indicate negative correlations.

We can observe that most of the features perform consistently across different years. Some features (i.e., feature#4: overall average citation number) keep on contributing positively, while others contribute (in-domain pub-interval (#40)) negatively. in-domain avgPubNo (#24), in-domain avgCiteNo (#25), and in-domain citation-network based PageRank (#26) are the three most important features in award winner prediction. Similar trend can be observed in Figure 3(b), where features show even more consistent performance than in award winner predictions.

6.2. **Qualitative topic modeling results.** We are also interested in evaluating the ability of our model in discovering latent topics in the author profile collections. Based on the learned results from the training set of predicting 2009 award winners for the sigir community (working on ACM data set), we report the Top 10 returned words for two identified topics, and compare them with the results obtained from the original LDA.

As shown in Table 8, we intend to retrieve more coherent topic-related words. For example, for topic on 'information retrieval', we can identify words like 'search', 'terms', which are relevant words but not ranked with Top 10 using LDA. On topic 'hardware', we can retrieve some relevant words as 'circuit' and 'clock'.



(a) (b)

Figure 3. Feature analysis (SIGKDD 2009) for award winner prediction (3(a)) and PC member prediction (3(b))

Table 8. Topic modeling results

LDA	LtoRTM	LDA	LtoRTM
Topics: Information retrieval		Topics: Hardware	
information	information	design	hardware
retrieval	retrieval	hardware	circuit
systems	query	level	circuits
query	document	architecture	delay
based	language	processor	architecture
model	model	paper	processor
document	text	data	routing
database	search	computer	bounds
language	terms	based	clock

Perplexity [8] is a standard measure to estimate the performance of topic modeling. Lower perplexity score indicates better generalization performance. Given a set of test words, perplexity can be defined as the exponential of the negative normalized predictive likelihood as follows:

$$P(d_i^{test}|\theta,\beta) = \prod_{w=1}^{V}\left(\sum_{z=1}^{K}\theta_{iz}\beta_{zw}\right)^{s_{iw}^{test}} \tag{9}$$

$$Perplexity = \exp - \frac{\sum_{i=1}^{M^{test}} \log(P(d_i^{test}|\theta,\beta))}{\sum_{i=1}^{M^{test}} N_i^{test}} \tag{10}$$

where M^{test} is the number of author profiles in testing set, and N_i^{test} is the number of words in profile d_i^{test}. s_{iw}^{test} indicates the word frequency of word w in testing profile i.

In order to test the generalization performance of our topic model, we vary the number of topics from 10 to 50, and compute the perplexity score for SIGKDD community on predicting award winners for years 2009 and 2006 on ACM data set. We compared our performance with that of sLDA.

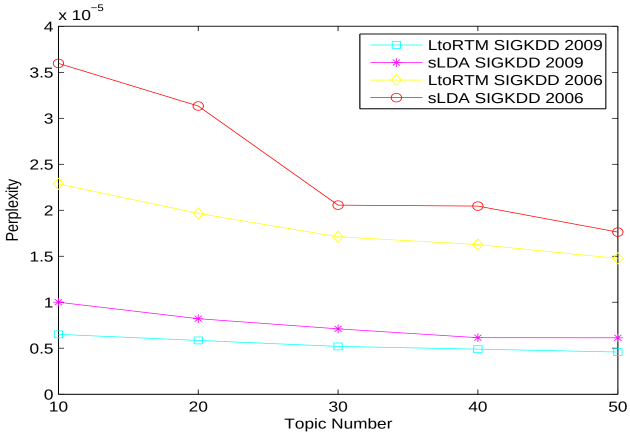


Figure 4. Perplexity

As shown in Figure 4, our LtoRTM model can achieve lower perplexity score, and therefore better generalization performance than sLDA for both years 2009 and 2006 under all different topic numbers.

7. **Discussions.** As demonstrated via experimental studies, our models can produce lower 'perplexity' scores than the supervised LDA model, which indicates that our models have stronger ability in predicting the word content of unseen data. This is a very important performance measurement for topic models. For both award winner prediction and PC member prediction tasks, our model works as the second best model since it outperforms AdaRank, Supervised LDA and Coordinate Ascent under most circumstances. Even though RankSVM still works the best in terms of overall performance, our models can outperform it in several individual communities.

Even though the theoretical inference of this model is complicated, it is easy to be applied to practical applications. The model is suitable for any ranking-oriented tasks. Suppose we have a ranking task in which we want to rank entities with regard to a certain query (a domain, a question or a community), to use the LtoRTM model in the training process, the only inputs we need to have from users are: the textual term frequency of each entity, the number of ⟨positive, negative⟩ pair-wise relationships in the corpus, and the list of those entity pairs. 'positive' here indicates the entity which is known to be relevant to a query and 'negative' is the entity known to be irrelevant to the query. For example, in the award winner prediction task, 'positive' entities are those existing award winners. To use the LtoRTMF model, users need to input the feature files for each entity as an addition. All this information is available and convenient to achieve in the experimental corpus.

Moreover, even though our model is proposed for expert ranking task in this paper, it can be applied to all other ranking-oriented research tasks, for example, to find the most popular tweets/twitter users, blogs/bloggers, and best answers/answer providers. We give two such examples in more detail as follows.

In Twitter search for influential twitter users, topic models can help to achieve users' expertise distributions over topics as represented in their posted tweets; other user-specific features, like user's age, gender, occupation, interests, geographical locations, number of followers/followees, number of tweets can be incorporated by the learning-to-rank scheme. This can help to identify the most influential twitter users in a specific community (domain-specific or community-related influential twitter user identification). It would also be helpful in finding the most popular tweets over topics as there are additional metadata on tweets, such as hash-tags and thematic labels provided by users. All these metadata can be well incorporated by the learning-to-rank scheme. If we further incorporate authors' information, we can develop systems that can retrieve the most popular tweets for specific group of users, as users of different types may care about different topics.

Similar mechanism can be applied to $Q\&A$ systems, where question-related, user-related and answer-related features can be explicitly represented and incorporated into the learning and ranking process. Therefore, we can return best answers for different groups of questions and users.

As a result, the model we developed is of strong capability for a wide range of applications. Expert ranking discussed in this paper is one typical representative of them.

8. **Conclusions.** In this paper, we propose a novel topic model that incorporates the preference between pairs of authors in terms of their authority of a specific domain into topic modeling process. It borrows the essential idea of pair-wise learning-to-rank algorithm and is particularly designed for modeling authors' authority (interests or influence) in academic environment. We further extend the model by introducing additional features related with authors' expertise beyond pure content. We provide introduction on model inference, parameter estimation, as well as the ranking scheme on new authors. Experiments conducted on two real world data sets have demonstrated our model to be either competitive or better than some state-of-the-art algorithms.

REFERENCES

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

[2] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov and S. Luo, Expertise retrieval, *Found. Trends Inf. Retr.*, vol.6, pp.127-256, 2012.

[3] P. Batista, M. Campiteli and O. Kinouchi, Is it possible to compare researchers with different scientific interests? *Scientometrics*, vol.68, pp.179-189, 2006.

[4] D. M. Blei and J. D. Lafferty, Dynamic topic models, *ICML*, pp.113-120, 2006.

[5] D. M. Blei and J. D. McAuliffe, Supervised topic models, *NIPS*, 2007.

[6] D. M. Blei, A. Y. Ng and M. I. Jordan, Latent dirichlet allocation, *Journal of Machine Learning Research*, pp.993-1022, 2003.

[7] J. Chang and D. M. Blei, Hierarchical relational models for document networks, *Annals of Applied Statistics*, 2010.

[8] X. Chen, X. Hu, Z. Zhou, C. Lu, G. Rosen, T. He and E. K. Park, A probabilistic topic-connection model for automatic image annotation, *CIKM*, pp.899-908, 2010.

[9] H. Deng, J. Han, M. Lyu and I. King, Modeling and exploiting heterogeneous bibliographics networks for expertise ranking, *JCDL*, 2011.

[10] D. Duan, Y. Li, R. Li, R. Zhang and A. Wen, Ranktopic: Ranking based topic modeling, *ICDM*, pp.211-220, 2012.

[11] E. Erosheva, S. Fienberg and J. Lafferty, Mixed-membership models of scientific publications, *Proc. of the National Academy Sciences*, pp.5220-5227, 2004.

[12] L. Egghe, Theory and practice of the g-index, *Scientometrics*, vol.69, pp.131-152, 2006.

[13] Y. Fang, L. Si and A. Mathur, Discriminative models of integrating document evidence and document-candidate associations for expert search, *SIGIR*, 2010.

[14] A. Harzing, *The Publish or Perish Book*, Tarma Software Research Pty Ltd, Melbourne, Australia, 2010.

[15] J. Hirsch, An index to quantify an individual's scientific research output, *Proc. of the National Academy of Sciences*, vol.102, no.46, pp.16569-16572, 2005.

[16] P. D. Hoff, A. E. Raftery and M. S. Handcock, Latent space approaches to social network analysis, *Journal of the American Statistical Association*, vol.97, pp.1090-1098, 2002.

[17] T. Hofmann, Probabilistic latent semantic indexing, *SIGIR*, pp.50-57, 1999.

[18] B. Jin, The *AR*-index: Complementing the h-index, *Intl. Society for Scientometrics and Informetrics Newsletter*, 2007.

[19] T. Joachims, Optimizing search engines using clickthrough data, *KDD*, pp.133-142, 2002.

[20] S. Kataria, P. Mitra, C. Caragea and C. Giles, Context sensitive topic models for author influence in document networks, *IJCAI*, 2011.

[21] T.-Y. Liu, Learning to rank for information retrieval, *Found. Trends Inf. Retr.*, vol.3, no.3, pp.225-331, 2009.

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

[23] C. Macdonald and I. Ounis, Learning models for ranking aggregates, *Proc. of the 33rd European Conference on Advances in Information Retrieval*, pp.517-529, 2011.

[24] D. Metzler and W. B. Croft, Linear feature-based models for information retrieval, *Inf. Retr.*, vol.10, no.3, pp.257-274, 2007.

[25] T. Minka and J. Lafferty, Expectation-propagation for the generative aspect model, *UAI*, 2002.

[26] C. Moreira, *Learning to Rank Academic Experts*, Master Thesis, Universidade Tecnica de Lisboa, 2011.

[27] R. Nallapati, A. Ahmed, E. Xing and W. Cohen, Joint latent topic models for text and citations, *KDD*, 2008.

[28] L. Nie, B. D. Davison and X. Qi, Topical link analysis for web search, *SIGIR*, pp.91-98, 2006.

[29] L. Page, S. Brin, R. Motwani and T. Winograd, The PageRank citation ranking: Bringing order to the Web, *Stanford InfoLab, Technical Report 1999-66*, 1998.

[30] D. Ramage, D. Hall, R. Nallapati and C. D. Manning, Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora, *EMNLP*, pp.248-256, 2009.

[31] M. Rosen-Zvi, T. Griffiths, M. Steyvers and P. Smyth, The author-topic model for authors and documents, *UAI*, 2004.

[32] F. Ruane and R. Tol, Rational (successive) h-indices: An application to economics in the Republic of Ireland, *Scienctometrics*, 2008.

[33] A. Sidiropoulos, D. Katsaros and Y. Manolopoulos, Generalized hirsch h-index for disclosing latent facts in citation networks, *Scientometrics*, vol.72, pp.253-280, 2007.

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

[35] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang and Z. Su, ArnetMiner: Extraction and mining of academic social network, *KDD*, 2008.

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

[37] J. Wang, X. Hu, X. Tu and T. He, Author-conference topic-connection model for academic network search, *CIKM*, pp.2179-2183, 2012.

[38] J. Xu and H. Li, Adarank: A boosting algorithm for information retrieval, *SIGIR*, pp.391-398, 2007.

[39] Z. Yang, L. Hong and B. D. Davison, Topic-driven multi-type citation network analysis, *RIAO*, 2010.

[40] Z. Yang, L. Hong and B. D. Davison, Academic network analysis: A joint topic modeling approach, *ASONAM*, pp.324-333, 2013.

[41] Z. Yang, D. Yin and B. D. Davison, Award prediction with temporal citation network analysis, *Proc. of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2011.

[42] Z. Yin, L. Cao, J. Han, C. Zhai and T. Huang, Geographical topic discovery and comparison, *WWW*, pp.247-256, 2011.

[43] C. Zhang, The e-index, complementing the h-index for excess citations, *PLos One*, vol.4, no.5, pp.1-4, 2009.

[44] J. Zhu, A. Ahmed and E. P. Xing, Medlda: Maximum margin supervised topic models for regression and classification, *ICML*, pp.1257-1264, 2009.