Neurocomputing 76 (2012) 71–83

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

On optimization of expertise matching with various constraints

Wenbin Tang [a,*], Jie Tang [a], Tao Lei [a], Chenhao Tan [b], Bo Gao [a], Tian Li [c]

[a] Department of Computer Science and Technology, Tsinghua University, Beijing 10084, China
[b] Department of Computer Science, Cornell University, Ithaca, NY 14850, USA
[c] Department of Computer Science, Beijing University of Aeronautics and Astronautics, Beijing 100084, China

ARTICLE INFO

Available online 31 July 2011

Keywords:
Expertise matching
Constrained optimization
Paper–reviewer assignment
Topic models

ABSTRACT

This paper studies the problem of expertise matching with various constraints. Expertise matching, which aims to find the alignment between experts and queries, is a common problem in many applications such as conference paper–reviewer assignment, product–reviewer alignment, and product-endorser matching. Most existing methods formalize this problem as an information-retrieval problem and focus on finding a set of experts for each query independently. However, in real-world systems, various constraints are often needed to be considered. For example, in order to review a paper, it is desirable that there is at least one senior reviewer to guide the reviewing process. An important question is: "Can we design a framework to *efficiently* find the *optimal solution* for expertise matching under various constraints?" This paper explores such an approach by formulating the expertise matching problem in a constraint-based optimization framework. In the proposed framework, the problem of expertise matching is linked to a convex cost flow problem, which guarantees an optimal solution under various constraints. We also present an online matching algorithm to support incorporating user feedbacks in real time. The proposed approach has been evaluated on two different genres of expertise matching problems, namely conference paper–reviewer assignment and teacher–course assignment. Experimental results validate the effectiveness of the proposed approach. Based on the proposed method, we have also developed an online system for paper–reviewer suggestions, which has been used for paper–reviewer assignment in a top conference and feedbacks from the conference organizers are very positive.

1. Introduction

The fusion of computer technology and human collective intelligence has recently emerged as a popular way for users to find and share information on the internet. For example, ChaCha.com, one of the largest mobile search engines, has already attracted users to answer over 300 million questions; Epinions.com, a consumer review site, has already collected thousands of millions of reviews for products. The human-based computation offers a new direction in search with its unique use of human intelligence; however, it also poses some brand new challenges. One key problem, referred to as expertise matching, is how to align human experts with questions (queries). Straightforwardly, we hope that the human experts who are assigned to answer a question have the specific expertise related to the question. But it is obviously insufficient. An ideal matching system should also consider various constraints in the real world, for example, an expert can only answer a certain number of questions (load balance); as the authoritative degree of different experts may vary largely, it is desirable that each question can be answered/reviewed by at least one senior expert (authority balance); a question may be related to multiple different aspects (topics), thus it is expected that the combined expertise of all assigned experts could cover all aspects of questions (topic coverage).

The problem has attracted considerable interests from different domains. For example, several works have been made for conference paper–reviewer assignment by using methods such as mining the web [1], latent semantic indexing [2], probabilistic topic modeling [3,4], integer linear programming [5], minimum cost flow [6] and hybrid approach of domain knowledge and matching model [7]. A few systems such as [8–13] have also been developed to help proposal–reviewer and paper–reviewer assignments. However, most existing methods mainly focus on improving the accuracy of measuring the relevance between queries and experts, i.e., how to find (or rank) relevant experts for each query, but ignore the different constraints or tackle the constraints using heuristics. Moreover, these methods usually do not consider user feedbacks. On the other hand, there are some methods focusing on expert finding. For example, Fang et al. [14] proposed a probabilistic model for expert finding, and Petkova et al. [15] employed

[*] Corresponding author.
E-mail address: tangwb06@gmail.com (W. Tang).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.04.039

a hierarchical language model in enterprise corpora. Balog et al. [16] employed probabilistic models to study the problem of expert finding, which try to identify a list of experts for a query. However, these methods retrieve experts for each query independently, and cannot be directly used to deal with the expertise matching problem. Thus, several key questions arise for expertise matching, i.e., how to design a framework for expertise matching to guarantee an optimal solution under various constraints? How to develop an online algorithm so that it can incorporate user feedbacks in real time?

Fig. 1 shows an example of paper–reviewer matching problem (assigning reviewers to each paper). In the problem, the topics corresponding to a reviewer (i.e., the expertise of the reviewer) can be "machine learning", "data mining", "computational theory", etc. Also, each paper has a distribution on different topics. There are some requirements to be a good assignment. First, for each paper, the assigned reviewers' expertise should cover the topics of the paper, and all the reviewers should have a load balance (each reviewer can only review a certain number of papers). In addition, some reviewers might be senior and some might be average. We always hope that the review process of each paper can be "supervised" by at least one senior reviewer. Another example is the patient–doctor matching case, the topics corresponding to the doctor's expertise include "pediatrics", "rheumatology", "neurology", etc. Each doctor has different expertise degrees on different topics, while the disease of a patient also has a relevance distribution on the topics. Ideally, when arranging a consultation for a patient, the topics of the assigned doctors should contain the potential causes of the patient's disease (e.g, a consultation for an SLE patient requires rheumatologist, nephrologist, cardiologist and neurologist), and all the doctors should have a load balance so that no doctor overstrains.

Contributions. In this paper, we formally define the problem of expertise matching and propose a constraint-based optimization framework to solve the problem. Specifically, the expertise matching problem is cast as a convex cost flow problem and the objective is then to find a feasible flow with minimum cost under certain constraints. We theoretically prove that the proposed framework can achieve an optimal solution under various constraints and develop an efficient algorithm to solve it. This paper is an extension and refinement of our previous conference paper [17], and differs the previous work in two aspects. (1) We reformalize our framework considering "multi-topic coverage" matching, which is very important to paper–reviewer assignment problem. We show that topic coverage measures (e.g. *Coverage* and *Confidence*) can also be incorporated into our framework. (2) An additional dataset is introduced to evaluate the performance of our approach on multi-topic coverage. Experimental results substantiate the effectiveness and efficiency of the proposed approach. We have applied the proposed method to help assign reviewers to papers for a top conference. Feedbacks from the conference organizers confirm the usefulness of the proposed approach.

The rest of the paper is organized as follows: Section 2 reviews the relevant literatures. Section 3 formally formulates the problem. Section 4 explains the proposed optimization framework. Section 5 gives experimental results that validate the effectiveness and the computational efficiency of our methodology. Finally, Section 6 concludes.

2. Related work

In general, existing methods for expertise matching mainly fall into two categories: probabilistic model and optimization model. The probabilistic model tries to improve the matching accuracy between experts and queries based on different probabilistic models such as keyword matching [1], latent semantic indexing [2], probabilistic topic modeling [3,4]. However, most of these methods do not consider the various constraints or simply consider the constraints by heuristics. The optimization model tries to incorporate the constraints as a component in an optimization framework such as integer linear programming [5] and minimum cost flow [6].

Most previous works cast expert matching or expert finding as an information-retrieval problem, in which every expert is represented as a "expertise" document and given a query, the goal is to retrieve most relevant experts. As a result, these methods mainly focus on two points: how to define the matching score between a query and a document; and how to represent each expert [18,19]. For example, Dumais and Nielsen [2] use latent semantic indexing (LSI) as the retrieval method and the abstracts provided by reviewer as expertise documents. Yu et al. [20] represent experts by analyzing text content and extracting related information. Basu et al. [1,21,22] integrate different sources of information for recommendation (e.g. publications, research interests, etc.). Yarowsky and Florian [23] assign a paper by computing its cosine similarity with a reviewer and choosing the one with the highest rank. Other expert finding work include [24,25].

In addition, different language models [14–16,26,27] and topic models [28] are used for expert matching/finding problem. In all of the language models, the matching score is the probability of a query given an expertise document, i.e., $p(q|d)$, but its definition varies. Mimno and McCallum [4] improve the matching accuracy by proposing a novel topic model Author-Persona-Topic (APT), in which experts are represented as independent distributions over topics. Karimzadehgan et al. also consider matching experts on multiple aspects of expertise [3]. Unlike the previous probabilistic

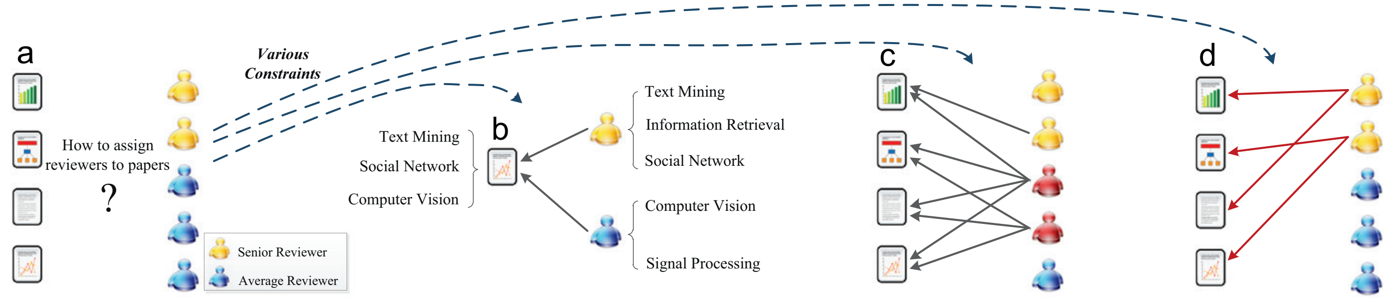


Fig. 1. (a) An illustration of the paper–reviewer assignment problem. To make an ideal assignment, some constraints should be considered. For example: (b) Each paper is related to one or several topics, each reviewer also has expertise on different topics. For a paper, the combination of the expertise of all assigned reviewers should cover all topics of the paper. (c) The assignment should be "load balanced". A bad example is shown, where the two reviewers marked with red are assigned with too many works, meanwhile the blue one does not get any assignment. (d) The authoritative degree of reviewers may vary largely, it is also desirable that each paper is assigned with at least one senior reviewer, so that she/he can guide the review process.

models in which a query is matched as a whole unit, it tries to find "comprehensive" matchings to cover all subtopics of a query. New measures *Coverage* and *Confidence* are defined to evaluate multi-aspect expertise matching results. Several matching methods are proposed and show better multi-aspect performance than traditional ones.

However, most of the aforementioned works treat each query independently and ignore the certain constraints (e.g. load balance), thus they cannot be directly adapted to an expertise matching system. In the real world, expertise matching is a highly constrained problem, and some existing works study this constrained optimization problem using various methods. For example, Guervós et al. [29] combine a greedy and an evolutionary algorithm [30,31] to assign papers to reviewers. Karimzadehgan et al. [5] and Taylor [32] cast it as an integer linear programming (ILP) problem so approximate solutions can be found by any ILP solver. Sun et al. [7] solves the reviewer assignment by hybrid approach of domain knowledge. Recently, a few systems [8–10,33,34,11–13] have also been developed to help proposal–reviewer and paper–reviewer assignments. However, the expertise matching problem is still treated as an information-retrieval problem, which obviously cannot result in an optimal solution.

In this paper, we aim to formalize the problem of expertise matching in a constraint-based optimization framework and propose an efficient algorithm to solve the framework. The differences of our work from existing work are: (a) we offer an optimization framework that incorporates the expertise matching and various constraints together; (b) the framework can be easily extended since new constraint can be combined into the optimization framework by simply defining a new (hard or soft) constraint; and (c) the framework can guarantee an optimal solution.

3. Problem formulation

In this section, we first give several necessary definitions and then present a formal definition of the problem.

Given a set of experts $V=\{v_i\}$, each expert has different expertise over all topics. Formally, we assume that there are T aspects of expertise (called topics) and each expert v_i has different expertise degrees on different topics. Further, given a set of queries $Q=\{q_j\}$, each query is also related to multiple topics. Given this, we first define the concept of topic model.

Definition 1 (*Topic model*). A topic model θ of an expert (or a query) is a multinomial distribution of words $\{p(w|\theta)\}$. Each expert (query) is considered as a mixture of multiple topic models. The assumption of this model is that words associated with the expert (query) are sampled according to the word distributions corresponding to each topic, i.e., $p(w|\theta)$. Therefore, words with the highest probability in the distribution would suggest the semantics represented by the topic.

Assuming we have T topics, the expertise degree of expert v_i on topic $z \in \{1 \cdots T\}$ is represented as a probability $\theta_{v_i z}$ with $\sum_z \theta_{v_i z} = 1$. Similarly, for each query, we also have a T-dimensional topic distribution with $\sum_z \theta_{q_j z} = 1$. Notations are summarized in Table 1.

It is easy to understand that each query q_j can be represented as a sequence of words, i.e., d_{q_j}. To represent every expert v_i, without loss of generality, we also consider it as a sequence of words, i.e., d_{v_i}. Based on this representation, we can calculate the similarity (or relevance score) between each query and every expert using measures such as cosine similarity or language model. Given this, we can define our problem of expertise matching with various constraints.

Table 1
Notations.

Symbol	Description
M	Number of experts
N	Number of queries
T	Number of topics
V	The set of candidate experts
Q	The set of queries
v_i	One expert
q_j	One query
$\theta_{v_i z}$	The probability of topic z given expert v_i
$\theta_{q_j z}$	The probability of topic z given query q_j
$T(v_i)$	The set of major related topics of expert v_i
$T(q_j)$	The set of major related topics of query q_j

Problem 1 (*Expertise matching with constraints*). Given a set of experts V and a set of queries Q, the objective is to assign m experts to each query by satisfying certain constraints, such as (1) the number of assigned queries with each expert should be in a range $[n_1,n_2]$, where $n_1 \le n_2$; (2) the experts' major topics should cover the query's related topics; (3) the assignment should avoid some conflict-of-interest (COI).

Actually in some applications, satisfying the constraints is more important than matching expertise with the queries. For example, in the conference paper–reviewer assignment, the authors of a paper should not be assigned to review their own papers. This must be a hard constraint. While in some other scenario, the constraint is relatively soft, for example the load balance among experts. The number of assigned queries to each expert can be in a range between n_1 and n_2. In existing works, Dumais et al. [2] and Mimno et al. [4] mainly focus on improving the accuracy of expertise matching, but ignore how to obtain an optimal matching satisfying the various constraints. Karimzadehgan et al. [5] use integer linear programming to find the solution for expertise matching with constraints. However, the proposed model cannot guarantee an optimal solution. In this work, we propose a generalizable optimization framework to solve this problem. Various constraints can also be incorporated in the framework.

4. The constrained optimization framework

In this section, we propose a constraint-based optimization framework for expertise matching. We develop an efficient algorithm to solve the optimization framework based on the theory of convex cost flow, and also present an online matching algorithm to incorporate user feedbacks in real time.

Basic idea. The basic idea of our approach is to formulate this problem in a constrained optimization framework. Different constraints can be formalized as penalty in the objective function or be directly taken as the constraints in the optimization solving process. For solving the optimization framework, we transform the problem to a convex cost network flow problem, and present an efficient algorithm which guarantees an optimal solution.

4.1. The framework

Now, we explain the proposed approach in detail. In general, our objective can be viewed from two perspectives: maximizing the matching score between experts and queries and satisfying the given constraints. Formally, we denote the set of experts to answer query q_j as $V(q_j)$, and the set of queries assigned to expert v_i as $Q(v_i)$. Further, we denote the matching score between expert v_i and query q_j as R_{ij}. Therefore, a basic objective function can be defined as follows:

$$\text{Max} \sum_{v_i \in V} \sum_{q_j \in Q(v_i)} R_{ij} \tag{1}$$

The objective function can be equivalently defined as $\sum_{q_j \in Q} \sum_{v_i \in V(q_j)} R_{ij}$. In different applications, the constraints will be different. Here we use several general constraints to explain how the proposed framework can incorporate different constraints.

The first constraint is that each query should be assigned with exactly m experts. For example, in the paper–reviewer assignment task, each paper should be assigned with 3 or 5 reviewers. This constraint can be directly added into the optimization problem. Formally, we have

$$\textbf{ST1}: \quad \forall q_j \in Q, \ |V(q_j)| = m \tag{2}$$

The second constraint is called as *expert load balance*, indicating that each expert can only answer a limited number of queries. There are two ways to achieve this purpose: define a *strict* constraint or add a *soft penalty* to the objective function.

For *strict*, we add a constraint indicating that the number of assigned queries to every expert v_i should be equal or larger than a minimum number n_1, but be equal or smaller than a maximum number n_2. The *strict* constraint can be written as

$$\textbf{ST2}\ (\textit{strict}): \quad \forall v_i \in V, \quad n_1 \le |Q(v_i)| \le n_2 \tag{3}$$

The other way is to add a soft penalty to the objective function (Eq. (1)). For example, we can define a square penalty as $|Q(v_i)|^2$. By minimizing the sum of the penalty $\sum_i |Q(v_i)|^2$, we can achieve a *soft* load balance among all experts, i.e.:

$$\text{softpenalty}: \quad \text{Min} \sum_{v_i \in V} |Q(v_i)|^2 \tag{4}$$

These two methods can be also used together. Actually, in our experiments, soft penalty method gives better results than strict constraint. Combining them together can yield a further improvement.

The third constraint is called *authority balance*. In real application, experts have different expertise level (authoritative level). Take the paper–reviewer assignment problem as an example. Reviewers may be divided into two levels: senior reviewers and average reviewers. Intuitively, we do not hope that the assigned reviewers to a paper are all average reviewers. It is desirable that the senior reviewers can cover all papers to guide (or supervise) the review process. Without loss of generality, we divide all experts into K levels, i.e., $V^1 \cup V^2 \cup \cdots \cup V^k = V$, with V^1 representing experts of the highest authoritative level. Similar to *expert load balance*, we can define a strict constraint like $|V^1 \cap V(q_j)| \ge 1$, and also add a penalty function to each query q_j over the k-level experts. Following, we give a simple method to instantiate the penalty function:

$$\textbf{ST3}: \quad \text{Min} \sum_{k=1}^{K} \sum_{j=1}^{N} |V^k \cap V(q_j)|^2 \tag{5}$$

Besides the above constraints, we also wish to assign experts that can cover all topics in the query. In the paper–reviewer assignment problem, for example, a paper may be related to several research areas, thus ideally in a comprehensive assignment, the paper is reviewed by a group of experts which cover all of the topics.

We introduce *related topics* of queries/experts. The related topics of query q_j and expert v_i are denoted as $T(q_j)$ and $T(v_i)$, indicating the most relevant aspects to the query and expert, respectively. $T(q_j)$ and $T(v_i)$ can be determined in different ways. Again in the paper–reviewer assignment problem, authors may be required to select the related topics of their papers from predefined categories in the submission, and reviewers can also select their expertise topics. In addition, it is also possible to estimate related topics from the learned topic distributions: (a) select top-k topics in θ_{v_i}, θ_{q_j} as the corresponding related topics; or (b) use thresholds τ_v and τ_q to prune topics, i.e., related topics are determined by $\theta_{v_i z} > \tau_v$ and $\theta_{q_j z} > \tau_q$.

Follow the work [3], we can incorporate different evaluation measures for topic covering. One measure is called *Coverage*, as we hope that assigned experts can cover different topics of a given query, i.e.,

$$\textit{Coverage}(q_j) = \frac{|T(q_j) \cap \bigcup_{v_i \in V(q_j)} T(v_i)|}{|T(q_j)|} \tag{6}$$

In this way, an optimal assignment should maximize the *Coverage*, e.g., $T(q_j) \subseteq \bigcup_{v_i \in V(q_j)} T(v_i)$. But this becomes the NP-hard set cover problem which is intractable. Consequently, we choose to find less optimal solutions by making further assumptions: for a specific query q_j, each assigned expert $v_i \in V(q_j)$ can select only one *responsible topic* $\hat{T}_{q_j}(v_i) \in T(v_i)$, and covers this topic for the query. The optimal solution under the assumption provides a lower bound of the original problem. Finding the matching maximizing the coverage with responsible topics actually opens another optimization problem, but fortunately this can be incorporated into our framework. We leave the discussion to Section 5.2.

Another measure proposed is called *Confidence*, as we prefer the related topics to be covered by as many experts as possible, i.e.,

$$\textit{Confidence}(q_j) = \frac{1}{m} \sum_{v_i \in V(q_j)} \frac{|T(q_j) \cap T(v_i)|}{|T(q_j) \cap \bigcup_{v_i \in V(q_j)} T(v_i)|} \tag{7}$$

Generally there is a *Coverage-Confidence* tradeoff. To achieve both high coverage and high confidence, a measure *Average Confidence* is accordingly defined as:

$$\textit{AverageConfidence}(q_j) = \frac{1}{m} \sum_{v_i \in V(q_j)} \frac{|T(q_j) \cap T(v_i)|}{|T(q_j)|} \tag{8}$$

We finally choose the average confidence as the fourth constraint in our optimization framework.

$$\textbf{ST4}: \quad \text{Max} \sum_{q_j \in Q} \sum_{v_i \in V(q_j)} \frac{|T(q_j) \cap T(v_i)|}{|T(q_j)|} \tag{9}$$

The last constraint is called *COI avoidance*. In many cases, we need to consider the conflict-of-interest (COI) problem. For example, an author, of course, should not review his own or his coauthors' paper. This can be accomplished through employing a binary $M \times N$ matrix U. An element with value of 0, i.e., $U_{ij}=0$, represents expert v_i has the conflict-of-interest with query q_j. A simple way is to multiply the matrix U with the matching score R in (Eq. (1)).

Finally, by incorporating different constraints in Eqs. (4)–(9) and the COI matrix U into the basic objective function (Eq. (1)), we can result in a constraint-based optimization framework, e.g.:

$$\begin{aligned}
\text{Max} \quad & \sum_{v_i \in V} \sum_{q_j \in Q(v_i)} U_{ij} R_{ij} - \sum_{k=1}^{K} \left(\mu_k \sum_{j=1}^{N} |V^k \cap V(q_j)|^2 \right) \\
& -\beta \sum_{v_i \in V} |Q(v_i)|^2 + \lambda \sum_{q_j \in Q} \sum_{v_i \in V(q_j)} \frac{|T(q_j) \cap T(v_i)|}{|T(q_j)|} \\
\text{s.t.} \quad & \forall q_j \in Q, \ |V(q_j)| = m \\
& \forall v_i \in V, \ n_1 \le |Q(v_i)| \le n_2
\end{aligned} \tag{10}$$

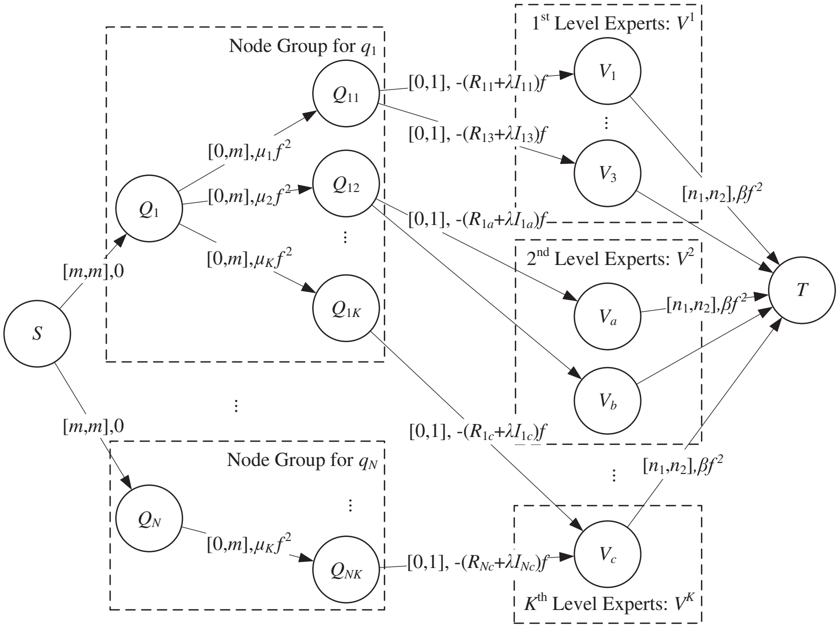

1st Level Experts: V^1
Node Group for q_1
Q_{11}
V_1
[0,1], -(R_{11}+λI_{11})f
[0,1], -(R_{13}+λI_{13})f
[0,m],$\mu_1 f^2$
Q_{12}
V_3
Q_1
[0,m],$\mu_2 f^2$
[n_1,n_2],βf^2
[0,1], -(R_{1a}+λI_{1a})f
2nd Level Experts: V^2
[m,m],0
[0,m],$\mu_K f^2$
Q_{1K}
V_a
[n_1,n_2],βf^2
T
S
V_b
[m,m],0
[0,1], -(R_{1c}+λI_{1c})f
Node Group for q_N
[n_1,n_2],βf^2
Q_N
[0,m],$\mu_K f^2$
Q_{NK}
[0,1], -(R_{Nc}+λI_{Nc})f
V_c
Kth Level Experts: V^K

Fig. 2. The construction of convex cost network flow according to objective function (10).

where λ, β and μ_k are lagrangian multipliers, used to tradeoff the importance of different components in the objective function.

Now the problem is how to define the topic distribution θ, how to calculate the pairwise matching score R_{ij}, and how to optimize the framework.

4.2. Modeling multiple topics

The goal of topic modeling is to associate each expert v_i with a vector $\theta_{v_i} \in \mathbb{R}^T$ of T-dimensional topic distribution, and as well to associate each query q_j with a vector $\theta_{q_j} \in \mathbb{R}^T$. The topic distribution can be obtained in many different ways. For example, in the paper–reviewer assignment problem, each reviewer can select their expertise topics from a predefined categories. In addition, we can use statistical topic modeling [35,36] to automatically extract topics from the input data. In this paper, we use the topic modeling approach to initialize the topic distribution of each expert and each query.

To extract the topic distribution, we can consider that we have a set of M expert documents and N query documents (each representing an expert or a query). An expert's document can be obtained by accumulating the content information related to the expert. For example, we combine all publication papers as the expert document of a reviewer, thus expert v_i's document can be represented as $d_{v_i} = \{w_{ij}\}$. Each query can also be viewed as a document. Then we can learn these T topic aspects from the collection of expert documents and query documents using a topic model such as LDA [36]. Specifically, let $D = \{d_{v_1}, \ldots, d_{v_M}\}$ be the set of experts' documents. The log-likelihood of the whole collection according to LDA is

$$\log p(D|\theta,\phi) = \sum_{d \in D}\sum_{w \in d} c(w,d) \log\left(\sum_{z=1}^{T} p(w|z,\phi_z)p(z|d,\theta_d)\right) \tag{11}$$

where $c(w,d)$ is the count of word w in document d, $p(w|z,\phi_z)$ is the probability of topic z generating word w, and $p(z|d,\theta_d)$ is the probability of document d containing topic z.

We use the Gibbs sampling algorithm [37,38] to learn the topic distribution θ_{v_i} for each expert. The topic distribution of query θ_{q_j} can be accordingly inferred from the produced θ_{v_i}.

4.3. Pairwise matching score

We employ a language model-based retrieval method to calculate the pairwise matching score. With language model, the matching score R_{ij} between expert v_i and query q_j is interpreted as a probability $R^{LM}_{ij} = p(q_j|d_i) = \prod_{w \in q_j} p(w|d_i)$, where

$$p(w|d_i) = \frac{N_{d_i}}{N_{d_i}+\lambda_D} \cdot \frac{tf(w,d_i)}{N_{d_i}} + \left(1 - \frac{N_{d_i}}{N_{d_i}+\lambda_D}\right) \cdot \frac{tf(w,\mathbf{D})}{N_{\mathbf{D}}} \tag{12}$$

where N_{d_i} is the number of word tokens in document d_i, $tf(w,d_i)$ is the number of occurring times of word w in d_i, $N_{\mathbf{D}}$ is the number of word tokens in the entire collection, and $tf(w,\mathbf{D})$ is the number of occurring times of word w in the collection $\mathbf{D}$. λ_D is the Dirichlet smoothing factor and is commonly set according to the average document length in the collection [26].

Our previous work extended LDA and proposed the ACT model [39] to generate a topic distribution. By considering the learned topic model, we can define another matching score as

$$R^{ACT}_{ij} = p(q_j|d_i) = \prod_{w \in q_j}\sum_{z=1}^{T} p(w|z,\phi_z)p(z|d,\theta_{d_i}) \tag{13}$$

Further, we can define a hybrid matching score by combining the two probabilities together

$$R^{H}_{ij} = R^{LM}_{ij} \times R^{ACT}_{ij} \tag{14}$$

4.4. Optimization solving

In order to maximize the objective function (Eq. (10)), we construct a convex cost network with lower and upper bounds imposed on the arc flows. Fig. 2 illustrates the constructing process as described in Algorithm 1.[1] Convex cost flow problem can be solved by transforming to an equivalent minimum cost flow problem [40]. The minimum cost flow of the network gives an optimal assignment with respect to (Eq. (10)).

[1] Every arc in the network is associated with lower and upper bound denoted as [l, u] and a convex function of the arc flow f.

Algorithm 1. Optimization solving algorithm.

```
     Input: The set of experts $V$; the set of queries $Q$; the
     matching score matrix $R_{M \times N}$; the COI matrix $U_{M \times N}$;
     Number of expertise
     level $K$; $m$, $n_1$, $n_2$ as described above.
     Output: An assignment of experts to queries maximizing
     objective function (10).
1.1  Create a network $G$ with source node $S$ and sink node $T$;
1.2  foreach $q_j \in Q$ do
1.3      Create $K+1$ nodes, denoted as $Q_j, Q_{j1}, \ldots, Q_{jK}$
1.4      respectively;
1.5      Add an arc from source node $S$ to node $Q_j$,
         with zero cost and flow constraint $[m,m]$;
         Add an arc from node $Q_j$ to $Q_{jk}$, with square cost function
         $\mu_k f^2$ and flow constraint $[0,m]$;
1.6  foreach $v_i \in V$ do
1.7      Create a node $V_i$;
1.8      Add an arc from $V_i$ to sink node $T$, with square cost function
         $\beta f^2$ and flow constraint $[n_1,n_2]$;
1.9  foreach $v_i \in V, q_j \in Q$, s.t. $U_{ij} = 1$ do
1.10     $k$ = expert level of $v_i$;
1.11     Add an arc from $Q_{jk}$ to $V_i$, with linear cost function
         $-(R_{ij} - \lambda I_{ij}) f$ and flow constraint $[0,1]$;
1.12 Compute the minimum cost flow on $G$;
1.13 foreach $v_i \in V, q_j \in Q$, s.t. $U_{ij} = 1$ do
1.14     $k$ = expert level of $v_i$;
1.15     if flow $f(Q_{jk}, V_i) = 1$ then Assign query $q_j$ to expert $v_i$;
```

Theorem 1. *Algorithm 1 based on minimum convex cost flow gives an optimal solution.*

Proof. First the minimum convex cost flow problem (MCCF) can be formulated as the following optimization problem:

$$\begin{aligned} \text{Min} \quad & \sum_{(a,b) \in E(G)} C_{ab}(f(a,b)) \\ \text{s.t.} \quad & \forall a \in V(G), \quad \sum_{b:(a,b) \in E(G)} f(a,b) = \sum_{b:(b,a) \in E(G)} f(b,a) \\ & \forall (a,b) \in E(G), \quad l_{ab} \le f(a,b) \le u_{ab} \end{aligned} \tag{15}$$

The model is defined on a directed network $G = (V(G), E(G))$ with lower bound l_{ab}, upper bound u_{ab} and a convex cost function $C_{ab}(f(a,b))$ associated with every arc (a,b).

Now we prove that minimizing (Eq. (15)) on the graph G constructed in Algorithm 1 is equivalent to maximizing (Eq. (10)). For simplicity, we use I_{ij} to denote $|T(q_j) \cap T(v_i)| / |T(q_j)|$. For the constructing process, we see a feasible flow on G is mapping to a query–expert assignment. The flow from S to Q_j indicates the number of experts assigned with query q_j, and the flow from V_i to T indicates the number of queries assigned to expert v_i. And the cost between V_i and T is corresponding to the *load balance* soft penalty function (Eq. (4)). The meaning of the flow from Q_j to Q_{jk} is the number of kth-level experts assigned to q_j, thus we impose a square cost function $\mu_k \cdot f^2$ on the arcs which is equivalent to the negative of the *authority balance* penalty. The flow from Q_{jk} to V_i means we assign query q_j to expert v_i, it is easy to find that no query will be assigned to the same expert twice since we give an upper bound of 1 on the arc, while the cost is equivalent to the negative of matching score and topic average confidence score. Therefore, our problem can be reduced to an equivalent MCCF problem, where the objective function of MCCF problem (Eq. (15)) is the negative form of (Eq. (10)).

In practice, it is not necessary to add all (Q_{jk}, V_i) arcs. To further reduce the complexity of the algorithm, we first greedily generate an assignment and preserve corresponding arcs, then keep only $c \cdot m$ arcs for Q_{jk} and $c \cdot n_2$ arcs for V_i which have highest matching score (c is a fixed constant). We call this process *Arc-Reduction*, which will reduce the number of arcs in the network without influencing the performance too much. To process large scale data, we can leverage the parallel implementation of convex cost flow [41].

4.5. Online matching

After an automatic expertise matching process, the user may provide feedbacks. Typically, there are two types of user feedbacks: (1) pointing out a false match; (2) specifying a new match. Online matching aims to adjust the matching result according to the user feedback. One important requirement is how to perform the adjustment at real time. In our framework, we provide online interactive adjustment without recalculating the whole cost flow. For both types of feedbacks, we can easily accomplish online adjustment by canceling some flows and augmenting new assignments in our framework. We give Algorithm 2 to consider the first type of feedback, which still produces an optimal solution.

Algorithm 2. Online matching algorithm.

```
     Input: A minimum cost network flow $f$ on $G$
     corresponding to the current assignment;
     an inappropriate match $(v_i, q_j)$.
     Output: A new assignment.
2.1  $k$=expert level of $v_i$;
2.2  if $f(Q_{jk}, V_i) = 1$ then
2.3      Construct the residual network $G(f)$;
2.4      Compute the shortest path $P_{back}$ from $T$ to $S$ on $G(f)$
         which contains backward arc $(V_i, Q_{jk})$;
2.5      Cancel(rollback) 1 unit of flow along $P_{back}$ and update $G(f)$;
2.6      Remove arc $(Q_{jk}, V_i)$ from $G$ and update $G(f)$;
2.7      Compute shortest augmenting path $P_{aug}$ from $S$ to $T$;
2.8      Augment 1unit of flow along $P_{aug}$;
```

Lemma 1 (*Negative cycle optimality conditions*). *Ahuja et al.* [40] *A feasible solution f^* is an optimal solution of the minimum cost flow problem if and only if it satisfies the negative cycle optimality conditions: namely, the residual network $G(f^*)$ contains no negative cost cycle.*

Theorem 2. *Algorithm 2 produces an optimal solution in the network without assignment (q_j, v_i).*

Proof. According to Lemma 1, the residual network $G(f)$ contains no negative cost cycle since the given flow f has the minimum cost. In Algorithm 2, we remove the inappropriate match (v_i, q_j) and adjust the network flow in line (2.3)–(2.5). Denote the feasible flow in the network after line (2.5) as f'. According to the SAP (short augmenting path) algorithm of cost flow, if f' has the minimum cost(i.e., $G(f')$ contains no negative cycle), the algorithm will give the optimal solution. We show the optimality of f' by contradiction. Assume $G(f')$ contains a negative cycle C, C must intersect with the shortest path P_{back} computed online (2.3), since the original $G(f)$ contains no negative cycle. Thus merging C into path P_{back} will generate a shorter path, which contradicts with the assumption that P_{back} is shortest. Therefore, f' has the minimum

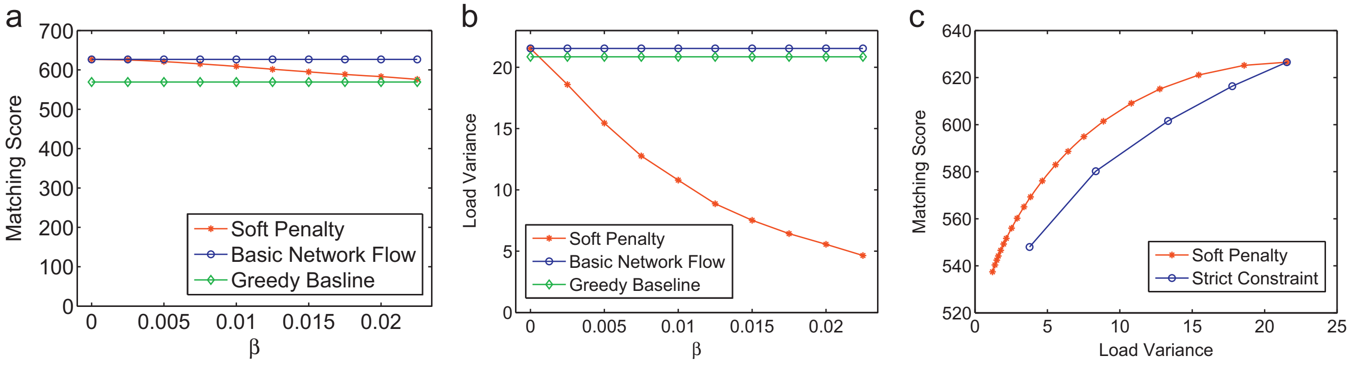


Fig. 3. (a) and (b) illustrate how soft penalty function influences the matching score (MS) and load variance with different β respectively. (c) gives a comparison between soft penalty function and strict constraint methods towards load balance.

cost. Accordingly, Algorithm 2 gives the optimal solution after augmenting the new assignment. □

5. Experimental results

The proposed approach for expertise matching is very general and can be applied to many application to align experts and queries. We evaluate the proposed framework on two different genres of expertise matching problems: paper–reviewer assignment and course–teacher assignment. Three experiments on different datasets are conducted to show the effectiveness of the proposed method. All datasets, code, and detailed results are publicly available.[2] All the experiments are carried out on a PC running Windows XP with Intel Core2 Quad CPU Q9550 (2.83 GHz), 4G RAM.

5.1. Paper–reviewer assignment experiment

Dataset. The paper–reviewer dataset consists of 338 papers and 354 reviewers. The reviewers are the program committee members of KDD'09 and the 338 papers are those published on KDD'08, KDD'09, and ICDM'09. For each reviewer, we collect her/his all publications from academic search system Arnetminer[3] [42] to generate the expertise document. As for the COI problem, we generate the COI matrix U according to the coauthor relationship in the last five years and the organization they belong to. Finally, we set that a paper should be reviewed by $m=5$ experts, and an expert at most reviews $n_2=10$ papers.

Baseline methods and evaluation metrics. We employ a greedy algorithm as the baseline. The greedy algorithm assigns experts with highest matching score to each query, while keeping the load balance for each expert (i.e., $|Q(v_i)| \le n_2$) and avoiding the conflict-of-interest.

As there are no standard answers, in order to quantitatively evaluate our method, we define the following metrics:

Matching score (*MS*): It is defined as the accumulative matching score.

$$MS = \sum_{v_i \in V} \sum_{q_j \in Q(v_i)} U_{ij} R_{ij}$$

Load variance (*LV*): It is defined as the variance of the number of papers assigned to different reviewers.

$$LV = \sum_{i=1}^{M} \left(|Q(v_i)| - \frac{\sum_{i=1}^{M} |Q(v_i)|}{M} \right)^2$$

Expertise variance (*EV*): It is defined as the variance of the number of top level reviewers assigned to different papers.

$$EV = \sum_{j=1}^{N} \left(|V(q_j) \cap V^1| - \frac{\sum_{j=1}^{N} |V(q_j) \cap V^1|}{N} \right)^2$$

Results. In this experiment, we tune different parameters to analyze the influence on the accumulative matching score. We also evaluate the efficiency of our proposed approach.

We first set $\mu = \mathbf{0}$ and tune the parameter β to find out the effects of soft penalty function. Fig. 3(a) illustrates how soft penalty function influences the matching score with different β. We see that the matching score decreases slightly with β increasing. Fig. 3(b) shows the effects of load variance with β varied. We see that the load variance changes very fast toward balance.

In Fig. 3(c), we compare the two different methods to achieve load balance, namely, strict constraint and soft penalty. The two LV–MS curves are respectively generated by setting different minimum numbers n_1 for strict constraint and varying the weight parameter β for soft load balance penalty. The curves show that soft penalty outperforms strict constraint towards load balance.

Then we set β to 0 to test the effects of authority balance. Experts are divided into two levels base on their H-index, and we set $\mu_2 = 0$ to consider the balance of the senior reviewers only. Fig. 4 presents the accumulative matching score (a) and expertise variance (b) with μ_1 varied.

Further, we analyze the effects of different constraints. Specifically, we first remove all constraints (using Eq. (1) only), and then add the constraints one by one in the order (load balance, authority balance, and COI). In each step, we perform expertise matching using our approach. Table 2 lists the accumulative matching score obtained in each step. We see that the load balance constraint will reduce the expertise matching score, while the other constraints have little negative effect. This is because senior experts are often good at many aspects (the matching score between them and many queries are large), thus assigned with heavy load in traditional matching. In our method, we try to get a more reasonable assignment by adding load balance constraint, which will restrict the work load of those senior experts. As a result, the matching score decreases.

To clearly illustrate the effect of load-balance constraint, we present Fig. 5, in which we see that traditional information-retrieval based method assigns many papers to senior reviewers, while some reviewers do not get any work. The load-balance constraint is necessary to generate a reasonable matching.

Finally, we evaluate the efficiency performance of the proposed algorithm. We compare the CPU time of the original optimal algorithm and the version with *Arc-Reduction*. As shown in Fig. 6, the *Arc-Reduction* process can significantly reduce the time consumption.

[2] http://www.arnetminer.com/expertisematching
[3] http://arnetminer.org

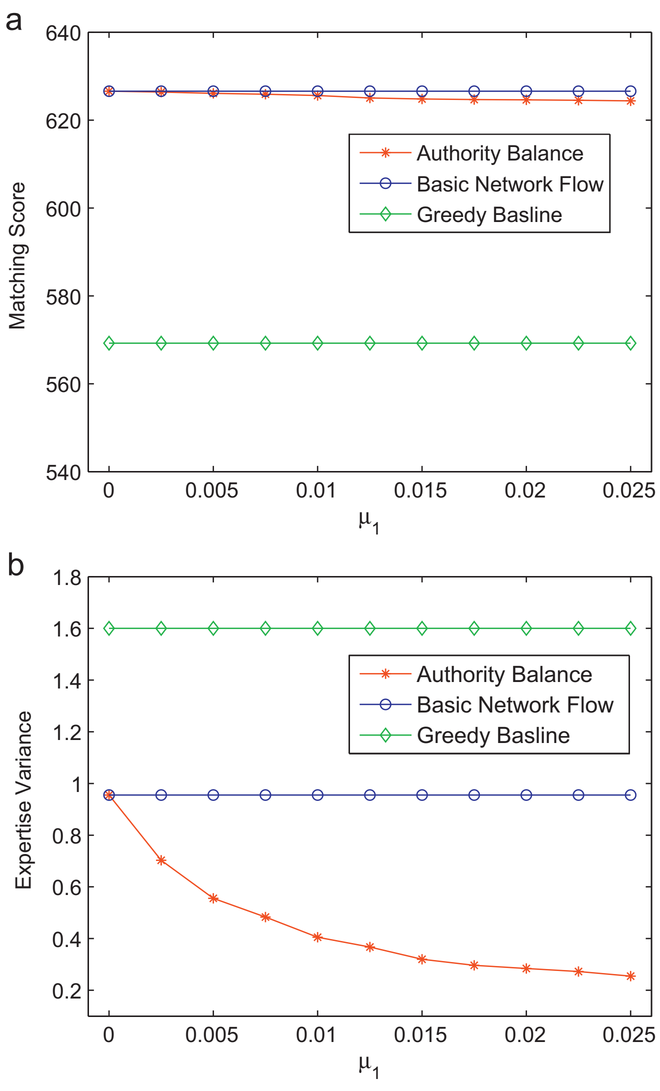


Fig. 4. Matching score (MS) and expertise variance (EV) with μ_1 varied.

Table 2
Effects of different constraints on matching score.

Constraint	Matching score
Basic objective function (Eq. (1))	635.51
+ Load balance *soft penalty* with $\beta = 0.02$	592.83
+ Authority balance with $\mu = (0.02,0)^T$	599.37
+ COI	590.14

For example, when setting $c = 12$ in this problem, we can achieve a $> 3\times$ speedup without any loss in matching score.

We further use a case study (as shown in Tables 3 and 4) to demonstrate the effectiveness of our approach. We see that the result is reasonable. For example, Lise Getoor, whose research interests include relational learning, is assigned with a lot of papers about social network.

5.2. Multi-topic paper–reviewer assignment experiment

Dataset. We use another dataset (D2) to verify the performance on "topic coverage". The dataset D2 is provided by [3],[4] consisting of 73 queries and 189 reviewers. The 73 queries are paper abstracts from SIGIR'07, where each of them is related to at least two topics. The documents of reviewers are the combination of all

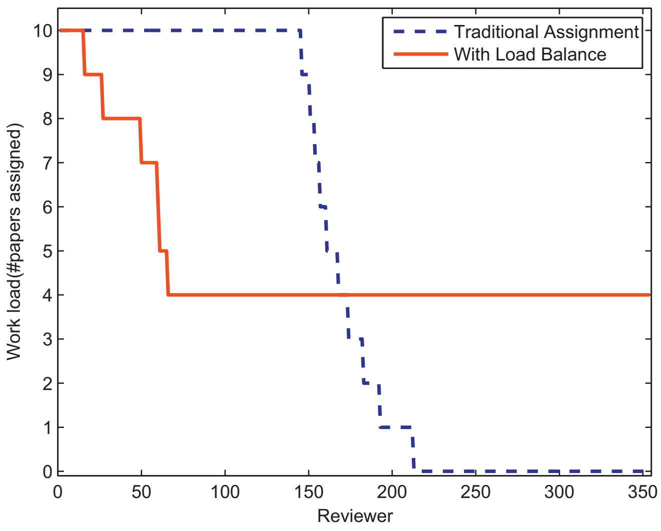


Fig. 5. The work load (number of paper assigned) of every reviewer.

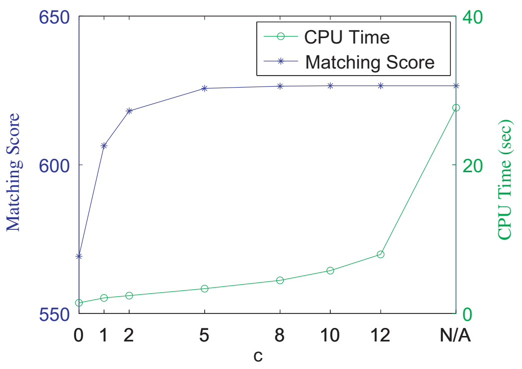


Fig. 6. Efficiency performance (s).

Table 3
Example assigned papers to three reviewers.

Reviewer	Assigned papers
Lise Getoor	Evaluating statistical tests for within-network classifiers of… Discovering organizational structure in dynamic social network Connections between the lines: augmenting social networks with text MetaFac: community discovery via relational hypergraph factorization Relational learning via latent social dimensions Influence and correlation in social networks
Wei Fan	Mining data streams with labeled and unlabeled training examples Vague one-class learning for data streams Active selection of sensor sites in remote sensing applications Name-ethnicity classification from open sources Consensus group stable feature selection Categorizing and mining concept drifting data streams
Jie Tang	Co-evolution of social and affiliation networks Influence and correlation in social networks Feedback effects between similarity and social influence… Mobile call graphs: beyond power-law and lognormal distributions Audience selection for online brand advertising: privacy-friendly…

[4] http://timan.cs.uiuc.edu/data/review.html

abstracts published in SIGIR1971-2006 by the corresponding reviewers. In this dataset, 25 topics such as “text mining”, “clustering” and “language models” are identified by human experts, and the related topics of each query and each reviewer are manually labeled as a gold standard for evaluation. We use the Lemur toolkit [43] for pre-processing to tokenize each query and document, removing common stop words.

Settings. Since the authors and reviewers in the dataset are anonymous, we do not consider COI and authoritative balance in this experiment. In order to be consistent with the setting in [3], we set a paper to be reviewed by $m=3$ experts, and use *Coverage* (Eq. (6)), *Confidence* (Eq. (7)), *AverageConfidence* (Eq. (8)), and F_{score} as the measures. In all the methods, the top topics with largest θ value are selected as the related topics for each query and expert, respectively.

Comparison methods. We implement three methods in [3] as comparison methods. The first baseline approach use a language model to retrieve reviewer documents for each query. Specifically, given a query q_j, one way to rank reviewers is to use the probability of query q_j given reviewer document d_i, i.e.,

$$p(q_j|d_{v_i}) = \prod_{w \in q_j} \frac{tf(w,d_{v_i}) + \mu p(w|D)}{N_{d_{v_i}} + \mu} \tag{16}$$

where $N_{d_{v_i}}$ is the document size of d_{v_i}, $tf(w,d_{v_i})$ is the number of occurring times of word w in d_{v_i}, $p(w|D)$ is the unigram language model of the entire reviewer document collection and μ is a Dirichlet smoothing factor which empirically set to 1000. In addition, we use a smoothed version of $p(w|D)$ so it would not return zero when word w is unseen in the entire document collection, i.e.,

$$p(w|D) = \frac{tf(w,D)+1}{N_D+V} \tag{17}$$

where V is the estimated vocabulary size (i.e., the number of distinct words).

Table 4
List of reviewers for five random papers.

Paper	Assigned reviewers
Audience selection for online brand advertising: privacy-friendly social network targeting	C. Lee Giles, Jie Tang, Matthew Richardson, Hady Wirawan Lauw, Elena Zheleva
Partitioned logistic regression for spam filtering	Rong Jin, Chengxiang Zhai, Saharon Rosset, Masashi Sugiyama, Annalisa Appice
Structured learning for non-smooth ranking losses	Xian-sheng Hua, Tie-yan Liu, Hang Li, Yunbo Cao, Lorenza Saitta
Unsupervised deduplication using cross-field dependencies	Chengxiang Zhai, Deepak Agarwal, Max Welling, Donald Metzler, Oren Kurland
The structure of information pathways in a social communication network	C. Lee Giles, Wolfgang Nejdl, Melanie Gnasa, Michalis Faloutsos, Cameron Marlow

The second baseline method uses KL-divergence for reviewer retrieval [26]:

$$\sum_{w \in q_j, tf(w,d_i)>0} p(w|q_j)\log\left(1+\frac{tf(w,d_i)}{\mu p(w|D)}\right) + \log\frac{\mu}{\mu+N_{d_i}} \tag{18}$$

We select the best method proposed in [3] as the third comparison method. The intuition of this method is to select a group of reviewers that work together to achieve a most similar topic distribution to the topics of a query. Specifically, given a paper query, this method picks reviewers one by one by choosing new reviewer with minimum KL-divergence value:

$$D(\theta_q || \theta^{r_k}_{r_1,\ldots,r_{k-1}}) \tag{19}$$

where θ_q is the topic distribution of the query and $\theta^{r_k}_{r_1,\ldots,r_{k-1}}$ is an averaged topic distribution of previously picked reviewers $r_1,\ldots,r_k$, i.e.,

$$p(z|\theta^{r_k}_{r_1,\ldots,r_{k-1}}) = \frac{\sigma}{k-1}\sum_{i=1}^{k-1} p(z|\theta_{r_i}) + (1-\sigma)p(z|\theta_{r_k}) \tag{20}$$

where σ indicates how much to rely on the previously picked reviewers $r_1,\ldots,r_{k-1}$.

The three methods will be referred as Baseline-Pr, Baseline-KL and Topic-KL in following discussion.

Matching with responsible topics. Maximizing the global *Coverage* actually is an NP-hard set cover problem. Thus we add an assumption to make it tractable, in which for a specific query q_j, each assigned expert $v_i \in V(q_j)$ can select only one *responsible topic* $\hat{T}_{q_j}(v_i) \in T(v_i)$, and covers this topic for the query. The optimization problem with the assumption provides a lower bound of the original one, and can be easily incorporated into our framework. As shown in Fig. 7, we create $|T_{q_j}|+1$ nodes for a query q_j, where Q_j is corresponding to the query, Q_{jk} represents the k-th related topic of q_j. We add two arcs from Q_j to Q_{jk} for every k, with costs of $-\sigma\theta_{jk}$ and 0 respectively. The costs of the arcs is mapping to the topic coverage, where σ is a multiplier used to adjust the importance. Then, for each expert v_i whose related topics include k, we add an arc from Q_{jk} to node V^j_i, which means the expert is able to be responsible for the topic. V^j_i is then connected to the expert node V_i, where node V^j_i is used to avoid the query to be assigned to the same expert twice. Combining with the matching

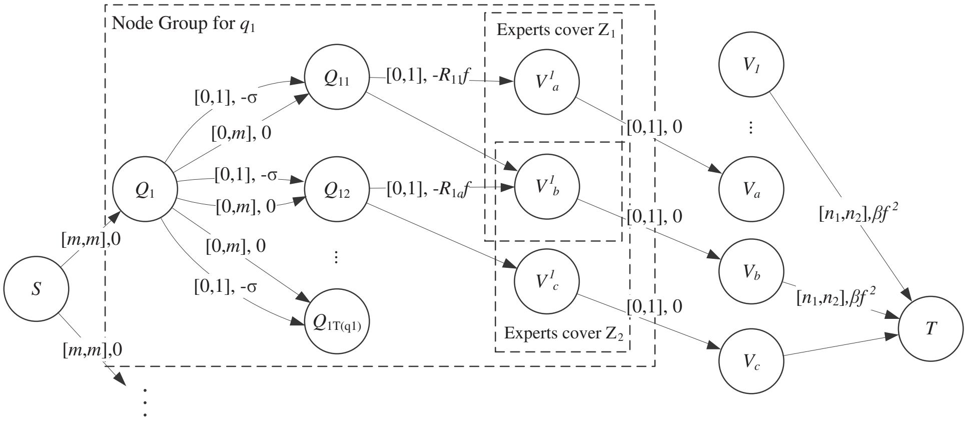


Fig. 7. The configuration of the network for matching with responsible topics.

score and the load balance constraint, we get the final configuration as shown Fig. 7.

Results. We exam our approach both with and without load balance constraint. The results of all methods on dataset D2 is presented in Table 5, from which we see that both our approach and Topic-KL outperform the two baseline methods on *Coverage*, *Confidence*, *AverageConfidence* and F_{score}. Interestingly, the baseline methods achieve better *Confidence*. This may be because the assigned reviewers of a paper are more likely to cover the same topic many times. Our approach without load balance achieves comparable performance with Topic-KL. It's noticeable when adding load balance constraint, the proposed method still produces good results, which validates the usefulness of our approach. In Table 5, we use our approach-L10, our approach-L5 to represent the proposed method with different load balance settings, i.e., $n_2=10$ and $n_2=5$.

Parameter sensitivity. Now we investigate the influences of the parameters in the framework. We first consider the multiplier σ, which controls the importance of *Coverage* score. When setting $\sigma=0$, the approach degenerates into the greedy baseline with poor performance. With a larger σ, both *Coverage* and *Confidence* increase and archive the best result at $\sigma=0.2$. It is also noticeable that the approach is not very sensitive to the parameter, as both *Coverage* and *Confidence* become stable with a relatively large σ (Fig. 8).

Table 5
Comparison of all methods on four different measures: *Coverage*, *Confidence*, *AverageCoverage* and F_{Score}. The number of topics is set to 20 when learning topic model.

Methods	**Coverage** (%)	**Confidence** (%)	**AverageConfidence** (%)	F_{score} (%)
Baseline-Pr	74	62	46	63
Baseline-KL	75	62	45	63
Topic-KL	87	58	53	67
Our approach	87	60	51	67
Our approach-L10	86	58	49	66
Our approach-L5	80	59	48	65

The number of topics also affects the performance. We employ the Gibbs sampling algorithm to learn topic models for different number of topics (e.g. 10, 30, 50 topics). In addition, we vary the number of related topics (i.e., $|T(q_j)|$ and $|T(v_i)|$) from 1 to 7, and $T(q_j)$, $T(v_i)$ are determined by selecting top-k topics in θ_{q_j} and θ_{v_i} as we have discussed in Section 4.1. The sensitivity curves of *Coverage* and *Confidence* are plotted in Fig. 8 and Fig. 9. With the help of pairwise matching scores, *Coverage* and *Confidence* remain $>80\%$ and $>54\%$ even we set $T(q_j)=T(v_i)=1$ and topic number to be 10. Moreover, we see that too large or too small number of related topics do not produce good results. The appropriate number of related topics is about 3 or 4, which is fairly near the ground truth. Setting too few topics (e.g. 10) may hurt the final performance, but using enough topics (20, 30, 50) would not make big differences. One explanation is, small number of topics may limit the discriminative power of topics.

5.3. Course–teacher assignment experiment

Dataset. In the course–teacher assignment experiment, we manually crawled graduate courses from the department of Computer Science (CS) of four top universities, namely CMU, UIUC, Stanford, and MIT. In total, there are 609 graduate courses from the fall semester in 2008 to 2010 spring, and each course is instructed by 1–3 teachers. Our intuition is that teachers' research interest often match the graduate courses he/she is teaching. Thus we still use the teachers' recent (five years) publications as their

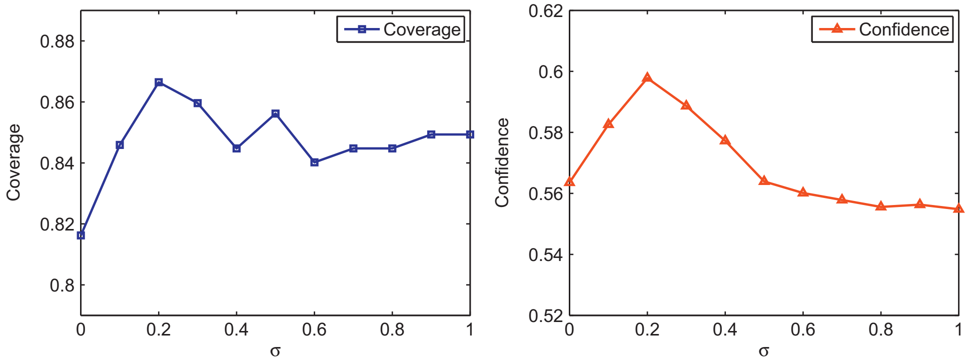


Fig. 8. Sensitivity of *Coverage* and *Confidence* to different Lagrangian multiplier σ.

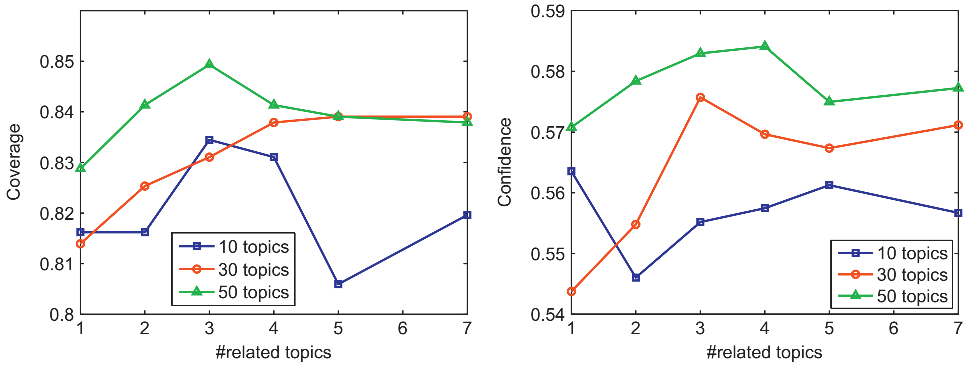


Fig. 9. Sensitivity of *Coverage* and *Confidence* to the number of related topics for different topic models, with $\sigma=0.3$.

expertise documents, while the course description and course name are taken as the query.

Baseline methods and evaluation metrics. We employ the same greedy method used in experiment 5.1 as baseline. The real assignment is extracted as the ground truth. Thus, we perform the evaluation in terms of precision.

Results. Fig. 10(a) shows the assignment precision in the course–teacher assignment task by our approach and the baseline method, and (b) shows the effects of the parameter β on the precision on UIUC data. The precision is defined as the ratio of the number of correct assignments(consistent with the ground truth data) over total number of assignments. As Fig. 10(a) shows, in all the datasets we collect from top universities, our algorithm outperforms the greedy method greatly. And in Fig. 10(b), as the β increases, the precision of our approach increases in general and decreases slowly after it exceeds the peak value. The peak value is more than 50% larger than the initial precision, which validates the effectiveness of the soft penalty approach.

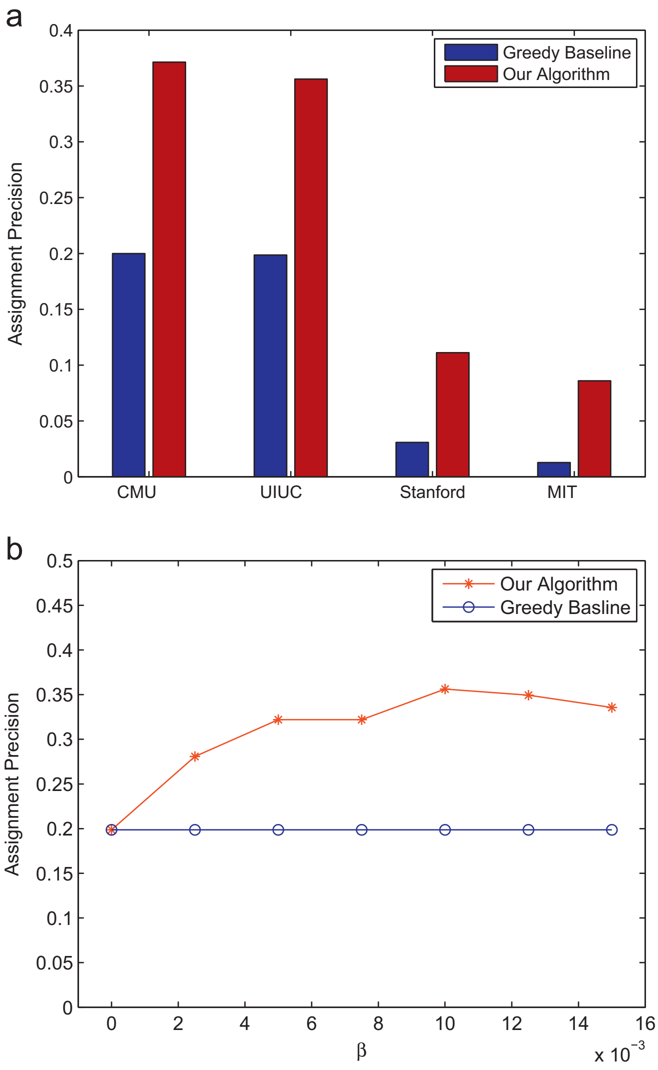


Fig. 10. Course–teacher assignment performance (%). (a) Course assignment results and (b) precision vs. β on UIUC data.

We conduct a further analysis on the UIUC dataset. As Table 6 shows, some professors with publications in various domains, are likely to be assigned with many courses in the baseline algorithm. But in real situation, most professors, though with various background, want to focus on several directions. Thus some courses should be assigned to younger teachers. While in our algorithm, the situation is much better. And we can see that each teacher is assigned with a reasonable load as well as a centralized interest.

5.4. Online system

Based on the proposed method, we have developed an online system for paper–reviewer suggestions, which is available at http://review.arnetminer.org/. Fig. 11 shows a screenshot of the system. The input is a list of papers (with titles, abstracts, authors, and organization of each author) and a list of conference program committee (PC) members. We use the academic information stored in ArnetMiner to find the topic distribution for each paper and each PC member [42]. With the two input lists and the topic distribution, the system automatically finds the match between papers and authors. As shown in Fig. 11, there are 5–7 papers assigned to each PC member and the number of reviewers for each paper is set as 3. The system will also avoid the conflict-of-interest (COI) according to the coauthorship and co-organization relationship. In addition, users can provide feedbacks for online adjustment, by removing or confirm (fix) an assignment.

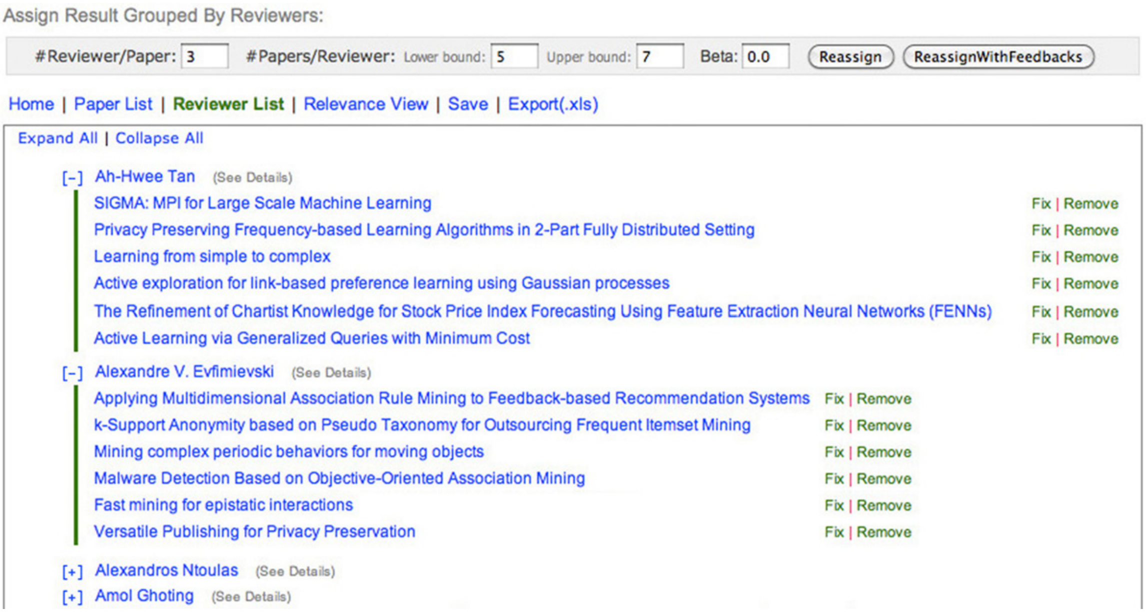
6. Conclusion and future work

In this paper, we study the problem of expertise matching in a constraint-based framework. We formalize the problem as a minimum convex cost flow problem. We theoretically prove that the proposed approach can achieve an optimal solution and develop an efficient algorithm to solve it. Experimental results on two different types of datasets demonstrate that the proposed approach can effectively and efficiently match experts with the queries. Also we provide an algorithm to consider user feedbacks in real time. We are now applying the proposed method to several real-world applications. Feedbacks from the users are very positive.

The general problem of expertise matching represents a new and an interesting research direction. There are many potential

Table 6
Case study: professors with many courses assigned in UIUC (2008, fall - 2010, spring).

Professor	Pub papers	Courses assigned (baseline)	Courses assigned (our approach)
Jose Meseguer	237	23 courses Database systems (2008, spring) Programming languages and compilers (2008, spring) Iterative and multigrid methods (2009, spring) Programming languages and compilers (2009, spring)	7 courses Programming languages and compilers (2008, spring) Programming language semantics (2008, spring) Programming languages and compilers (2008, fall) Programming languages and compilers (2009, spring)
ChengXiang Zhai	117	18 courses Computer vision (2009, spring) Text information systems (2009, spring) Stochastic processes and applic (2009, fall) Computer vision (2008, spring)	7 courses Text information systems (2008, spring) Stochastic processes and applic (2008, fall) Text information systems (2009, spring) Stochastic processes and applic (2009, fall)


Assign Result Grouped By Reviewers:
Reassign
ReassignWithFeedbacks
Home | Paper List | Reviewer List | Relevance View | Save | Export(.xls)
Expand All | Collapse All

#Reviewer/Paper: 3 #Papers/Reviewer: Lower bound: 5 Upper bound: 7 Beta: 0.0

[-] Ah-Hwee Tan (See Details)
- SIGMA: MPI for Large Scale Machine Learning — Fix | Remove
- Privacy Preserving Frequency-based Learning Algorithms in 2-Part Fully Distributed Setting — Fix | Remove
- Learning from simple to complex — Fix | Remove
- Active exploration for link-based preference learning using Gaussian processes — Fix | Remove
- The Refinement of Chartist Knowledge for Stock Price Index Forecasting Using Feature Extraction Neural Networks (FENNs) — Fix | Remove
- Active Learning via Generalized Queries with Minimum Cost — Fix | Remove

[-] Alexandre V. Evfimievski (See Details)
- Applying Multidimensional Association Rule Mining to Feedback-based Recommendation Systems — Fix | Remove
- k-Support Anonymity based on Pseudo Taxonomy for Outsourcing Frequent Itemset Mining — Fix | Remove
- Mining complex periodic behaviors for moving objects — Fix | Remove
- Malware Detection Based on Objective-Oriented Association Mining — Fix | Remove
- Fast mining for epistatic interactions — Fix | Remove
- Versatile Publishing for Privacy Preservation — Fix | Remove

[+] Alexandros Ntoulas (See Details)

[+] Amol Ghoting (See Details)

Fig. 11. Screenshot of the online system.

future directions of this work. One interesting issue is to apply the proposed framework to question answer (e.g., Yahoo! Answer), where one of the most important issues is how to identify who can answer a new question. Another interesting issue is to incorporate some supervised information into our framework to further improve the performance of expertise matching. Finally, it is important to consider the influence between users when extending expertise matching to the social network.

Acknowledgments

The work is supported by the Natural Science Foundation of China (Nos. 61073073, 60703059, and 60973102), Chinese National Key Foundation Research (No. 60933013), and National High-tech R&D Program (No. 2009AA01Z138). The work was done when the last author was visiting Tsinghua University.

References

[1] C.B. Haym, H. Hirsh, W.W. Cohen, C. Nevill-manning, Recommending papers by mining the web, in: Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI'99), 1999, pp. 1–11.

[2] S.T. Dumais, J. Nielsen, Automating the assignment of submitted manuscripts to reviewers, in: SIGIR'92: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 1992, pp. 233–244.

[3] M. Karimzadehgan, C. Zhai, G. Belford, Multi-aspect expertise matching for review assignment, in: Proceedings of the 17th ACM International Conference on Information and Knowledge Management (CIKM'08), 2008, pp. 1113–1122.

[4] D. Mimno, A. McCallum, Expertise modeling for matching papers with reviewers, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'07), 2007, pp. 500–509.

[5] M. Karimzadehgan, C. Zhai, Constrained multi-aspect expertise matching for committee review assignment, in: Proceedings of the 17th ACM International Conference on Information and Knowledge Management (CIKM'09), 2009, pp. 1697–1700.

[6] D. Hartvigsen, J.C. Wei, R. Czuchlewski, The conference paper–reviewer assignment problem, Decision Sciences 30 (3) (1999) 865–876.

[7] Y.-H. Sun, J. Ma, Z.-P. Fan, J. Wang, A hybrid knowledge and model approach for reviewer assignment, in: Proceedings of the 40th Hawaii International Conference on Systems Science (HICSS-40 2007), 2007, p. 47.

[8] S. Hettich, M.J. Pazzani, Mining for proposal reviewers: lessons learned at the national science foundation, in: Proceeding of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06), 2006, pp. 862–871.

[9] D. Conry, Y. Koren, N. Ramakrishnan, Recommender systems for the conference paper assignment problem, in: RecSys'09: Proceedings of the Third ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2009, pp. 357–360.

[10] N.D. Mauro, T.M.A. Basile, S. Ferilli, Grape: an expert review assignment component for scientific conference management systems, in: IEA/AIE'05: Proceedings of the 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 2005, pp. 789–798.

[11] R. Van De Stadt, Cyberchair: A Web-Based Groupware Application To Facilitate The Paper Reviewing Process, URL 〈http://borbala.com/cyberchair/wbgafprp.pdf〉.

[12] Microsoft Conference Management Toolkit (cmt), URL 〈http://cmt.research.microsoft.com/cmt/〉.

[13] The Easychair Software, URL 〈http://www.easychair.org/〉.

[14] H. Fang, C. Zhai, Probabilistic models for expert finding, in: Proceedings of the 29th European Conference on Information Retrieval Research ECIR'07, 2007, pp. 418–430.

[15] D. Petkova, W.B. Croft, Hierarchical language models for expert finding in enterprise corpora, International Journal on Artificial Intelligence Tools (2008) 5–18.

[16] K. Balog, L. Azzopardi, M. de Rijke, Formal models for expert finding in enterprise corpora, in: Proceedings of the 29th ACM SIGIR International Conference on Information Retrieval (SIGIR'2006), 2006, pp. 43–55.

[17] W. Tang, J. Tang, C. Tan, Expertise matching via cosntraint-based optimization, in: Proceedings of 2010 IEEE/WIC/ACM International Conferences on Web Intelligence (WI'2010), 2010.

[18] D. Yimam, A. Kobsa, Demoir: a hybrid architecture for expertise modeling and recommender systems, in: Proceedings of the Ninth IEEE International Workshops on Enabling Technologies: Infrastructure for Colaborative Enterprises, 2000, pp. 67–74.

[19] Y. Cao, J. Liu, S. Bao, H. Li, Research on expert search at enterprise track of trec 2005, in: TREC, 2005.

[20] Y. Fu, W. Yu, Y. Li, Y. Liu, M. Zhang, S. Ma, Thuir At Trec 2005: Enterprise Track, 2005.

[21] C. Basu, H. Hirsh, W.W. Cohen, C. Nevill-Manning, Technical paper recommendation: a study in combining multiple information sources, Journal of Artificial Intelligence Research 14 (2001) 231–252.

[22] C. Basu, H. Hirsh, W. Cohen, Recommendation as classification: using social and content-based information in recommendation, in: Proceedings of the Fifteenth National Conference on Artificial Intelligence, AAAI Press, 1998, pp. 714–720.

[23] D. Yarowsky, R. Florian, Taking the Load Off The Conference Chairs: Towards A Digital Paper–Routing Assistant, 1999.

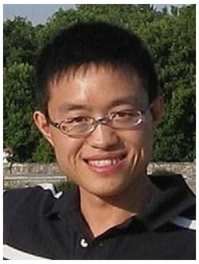
[24] J. Zhang, J. Tang, J. Li, Expert finding in a social network, Advances in Databases: Concepts Systems and Applications 23 (2010) 1066–1069.

[25] D. Yimam-Seid, A. Kobsa, Expert finding systems for organizations: problem and domain analysis and the demoir approach, Sharing Expertise: Beyond Knowledge Management 23 (2003) 327–358.

[26] C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to ad hoc information retrieval, in: Proceedings of the 24th ACM SIGIR International Conference on Information Retrieval (SIGIR'01), 2001, pp. 334–342.

[27] K. Balog, L. Azzopardi, M. de Rijke, A Language Modeling Framework For Expert Finding, 2008.
[28] X. Wei, W.B. Croft, Lda-based document models for ad-hoc retrieval, in: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR'06, ACM, New York, NY, USA, 2006, pp. 178–185.
[29] J.J.M. Guervós, P.A.C. Valdivieso, Conference paper assignment using a combined greedy/evolutionary algorithm, in: PPSN, 2004, pp. 602–611.
[30] E. Zitzler, L. Thiele, Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, 1999.
[31] M. Cámara, J. Ortega, F. de Toro, A single front genetic algorithm for parallel multi-objective optimization in dynamic environments, Neurocomputing 72 (2009) 3570–3579.
[32] C.J. Taylor, On the Optimal Assignment of Conference Papers to Reviewers, Technical Report, MS-CIS-08-30, Computer and Information Science Department, University of Pennsylvania, 2008.
[33] M.A. Rodriguez, J. Bollen, An algorithm to determine peer–reviewers, in: Paroceedings of the Conference on Information and Knowledge Management, ACM Press, Napa, California, 2008, pp. 319–328 doi: 10.1145/1458082.1458127.
[34] S. Benferhat, J. Lang, Conference paper assignment, International Journal of Computational Intelligence Systems 16 (10) (2001) 1183–1192.
[35] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), 1999, pp. 50–57.
[36] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.
[37] T.L. Griffiths, M. Steyvers, Finding scientific topics, Proceedings of the National Academy of Sciences PNAS'04 (2004) 5228–5235.
[38] J. Cao, T. Xia, J. Li, Y. Zhang, S. Tang, A density-based method for adaptive lda model selection, Neurocomputing 72 (2009) 1775–1781.
[39] J. Tang, R. Jin, J. Zhang, A topic modeling approach and its integration into the random walk framework for academic search, in: Proceedings of 2008 IEEE International Conference on Data Mining (ICDM'08), 2008, pp. 1055–1060.
[40] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, 1993.
[41] P. Beraldi, F. Guerriero, R. Musmanno, Parallel algorithms for solving the convex minimum cost flow problem, Computational Optimization and Applications 18 (2) (2001) 175–190.
[42] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, Z. Su, Arnetminer: Extraction and mining of academic social networks, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'08), 2008, pp. 990–998.
[43] The Lemur Toolkit For Language Modeling And Information Retrieval, URL 〈http://lemurproject.org/〉.

Wenbin Tang is a master student in Tsinghua University, supervised by Prof. Jie Tang. His research interests are text mining, social network and computer vision.



Jie Tang is an associate professor in Tsinghua University. His research interests are machine learning and text mining.

Tao Lei is currently a research assistant in KEG Group, Tsinghua University. He obtained a BS degree in Peking University. His research interests focus on machine learning and text mining.

Chenhao Tan is a Ph.D. student in Cornell University. His research interests are machine learning and social network.

Bo Gao is a software engineer in KEG Group, Tsinghua University. He is currently in charge of the development and maintenance of the academic social network ArnetMiner.

Tian Li is an undergraduate student in Beijing University of Aeronautics and Astronautics.