Patent Partner Recommendation in Enterprise Social Networks

Sen Wu
Department of Computer Science
Tsinghua University
Beijing 100084, China
ronaldosen@gmail.com

Jimeng Sun
IBM T. J. Watson Research Center
USA
jimeng@us.ibm.com

Jie Tang
Department of Computer Science
Tsinghua University
Beijing 100084, China
jietang@tsinghua.edu.cn

ABSTRACT

It is often challenging to incorporate users' interactions into a recommendation framework in an online model. In this paper, we propose a novel interactive learning framework to formulate the problem of recommending patent partners into a factor graph model. The framework involves three phases: 1) candidate generation, where we identify the potential set of collaborators; 2) candidate refinement, where a factor graph model is used to adjust the candidate rankings; 3) interactive learning method to efficiently update the existing recommendation model based on inventors' feedback. We evaluate our proposed model on large enterprise patent networks. Experimental results demonstrate that the recommendation accuracy of the proposed model significantly outperforms several baselines methods using content similarity, collaborative filtering and SVM-Rank. We also demonstrate the effectiveness and efficiency of the interactive learning, which performs almost as well as offline re-training, but with only 1 percent of the running time.

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Miscellaneous; H.3.3 [**Information Search and Retrieval**]: Text Mining

General Terms

Algorithms, Experimentation

Keywords

Cross collaboration, Social network, Predictive model

1. INTRODUCTION

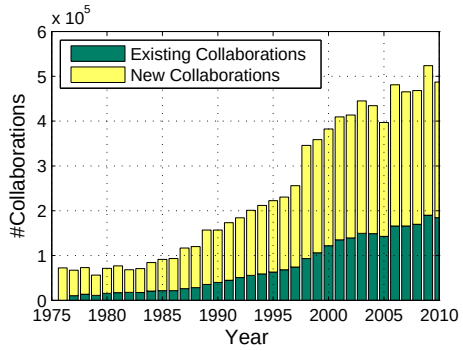
Collaboration exists almost everywhere, for example patent application, academic research, product development, and decision making. 98% of US patents are results of collaborations. Figure 1(a) shows a rapid increase of the co-invention relationships between inventors in the past 35 years (1975-2010). Today's patent collaborations are almost 6 times higher than that of 35 years before. In particular, the number of new collaborations has an even more rapid increase. Figure1(b) shows another interesting analysis, the changes of average number of patents by each individual and the average number of inventors for each patent over the past 35 years. We see the number of inventors for each patent has a clear increasing trend.

Indeed, collaboration is becoming so important that the Open Government Initiative has given high priority to increase the use of collaboration in the federal government[1]. In most of the industries, employees only use part of their time to create patents. Time constraint often limits their scope for choosing new collaborators. In general, collaborators should be ideally in closer proximity (i.e, within same company and geography), engaged in similar interests, and working on similar product. However, it is often challenging for researchers to establish successful collaborations. This is sometimes even more challenging within a company. IBM has 300,000 employees distributed in many different locations across the globe. How can healthcare researchers find the right data mining experts with right skills and are willing to help?

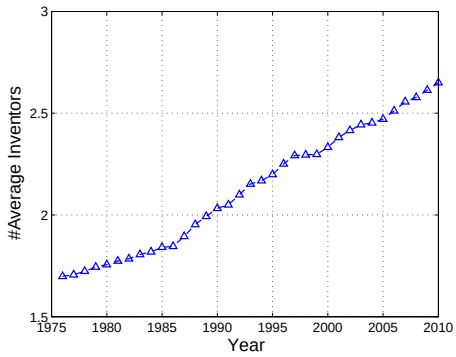
There are a few systems for patent search and analysis such as Google Patent, WikiPatent, PatentMiner, FreePatentsOnline, Patents, PatentLens, and PriorArtSearch. However, most of these systems focus on search and provide limited micro-level analytic functions. Few systems provide the function of patent partner recommendations. For research on the patent data, Tang et al. [26] propose a topic-driven patent analysis and mining method and develop a system named as PatentMiner. Jin et al. [11] propose a method to evaluate the quality of patents. Liu et al. [18] and Mann [19] study how to estimate patent quality from the perspective of court validity rulings or the number of forward citations. However, all these works focus on analyzing patent content, but ignore the collaborative relationships between inventors. In the space of data mining and web search, a number of relevant works have been conducted, such as collaborator recommendations [13, 27], friends suggestions [22], and reviewer finding [21, 29]. Kautz et al. [13] introduce a system called ReferralWeb which attempts to combine social networks for collaborative filtering. None of the above works directly target at patent collaborations, which is the focus of this paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WSDM'13, February 4–8, 2013, Rome, Italy.
Copyright 2013 ACM 978-1-4503-1869-3/13/02 ...$15.00.

[1]http://www.crosscollaborate.com



(a) #co-invention relationships



(b) #average inventors

Figure 1: Trends of co-invention relationships from 1975 to 2010.

Challenges and Contributions. In this work, we study the problem of recommending patent partners in enterprise social networks. We address three challenges: First, what are the fundamental factors that influence the formation of co-invention relationships? Will inventors with similar interests tend to collaborate, or they collaborate because they have complementary research interest, or simply due to the geographical proximity? Second, how to design an interactive mechanism so that the user can provide feedback to the system to refine the recommendations? Third, technically, how to learn the interactive recommendation framework in an online mode?

We formulate the problem and propose a ranking model to recommend patent collaborators according to the user's profile. More specifically, we propose an interactive learning framework to formulate the problem of recommending patent partners into a factor graph model. The framework involves three phases:

- **Candidate generation**, where we identify the potential set of collaborators by performing similarity retrieval on various features including homophily, referral chaining, and recency;
- **Candidate refinement**, where a factor graph model RankFG is used to adjust the candidate rankings;
- **Interactive learning**, where we propose an efficient online update algorithm RankGF+ to adjust the existing recommendation model based on inventors' feedback.

We evaluate the proposed model on large patent data sets. We show that the proposed model can significantly improve (up to 30% in terms of Precision@5) the recommendation performance of collaborations, compared with several baseline methods.

Organization. Section 2 formulates the problem; Section 3 explains the proposed model and describes the algorithm for learning the model; Section 4 introduces the data set and our observations to verify the hypotheses. Section 5 presents the results; finally, Section 6 discusses related work and Section 7 concludes this paper.

2. PROBLEM DEFINITION

The patent data set can be considered as an inventor social network $G = (V, E)$, where V is a set of $|V| = N$ inventors and $E \subseteq V \times V$ is a set of co-invention relationships between inventors. Let $\mathbf{x}_i$ be a set of attributes associated with inventor v_i. An attribute can be the inventor's interest, her employed company or the number of published patents. We use $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ to denote the attributes of all inventors. Our goal is to suggest future collaborators for a specific inventor v_q, based on her historic attributes $\mathbf{x}_q$, her existing co-invention relationships, and the idea she wants to patent. More precisely, we are concerned with the following problem:

Problem 1. **Suggesting co-invention relationships.** Let $G = (V, E, X)$ be an attribute augmented inventor network. For a particular inventor $v_q \in V$ and the topic t she wants to find co-invention relationships, the task of suggesting co-invention relationships is to find a predictive function such that we can suggest (infer) co-invention relationships between v_q and other inventors on topic t, i.e.,

$$f : (G, v_q, t) \rightarrow Y,$$

where $Y = \{y_1, \cdots, y_{|V|}\}$ is a set of inferred results between inventor v_q and all inventors in the network G; $y_k \in \{0, 1\}$ is binary score indicating whether the corresponding inventor $v_k \in V$ will have a co-invention relationship with v.

In addition, we consider how to interactively refine the learned predictive function. Specifically, the inventor can provide feedback on the recommendation list and then the technical objective is to incrementally update the prediction at a real time. The problem can be also generalized to suggest top-K collaborators (under some constraints) for any social networks. To keep things concrete, we focus on studying this problem in the inventor social network.

The problem formulation is different from existing work on patent quality analysis [11, 18, 19, 26], which has focused on mining patent content. It is also different from existing works on collaborator recommendations [13] and friends suggestions [22]. Here we mainly consider how to provide an interactive mechanism for supporting inventors to provide feedback to the recommended collaborators so as to refine the recommendations at a real time.

3. MODEL FRAMEWORK

A major motivation for our work comes from the intuition that inventors' content and social networks are both important for establishing co-invention relationships. At a high level, the proposed approach framework consists of three stages.

- **Candidate generation.** First, given a user v_q and the topic t on which she/he wants to establish co-invention relationships, we extract potential collaboration candidates through the similarity on several features.
- **Candidate refinement.** Second, the candidate list is fed to a ranking factor graph (RankFG) model to refine the ranking. The model incorporates various factors and the correlation among the collaboration candidates.
- **Interactive feedback.** Third, users can provide feedback to the suggested co-invention relationships. An interactive learning algorithm is designed to update the ranking model incrementally based on the user's feedback.

3.1 Candidate Generation

For a given user v_q and the topic t (i.e., query keywords), we consider three factors to generate the candidate list, i.e., *Homophily*, *Referral chaining*, and *Recency*.

The idea of homophily comes from the principle of "birds of a feather flock together" [20], which suggests that "connected" inventors tend to have similar characteristics (e.g., social status or interests). Regarding the homophily between inventors, we consider the following inventors' attributes: *geographical proximity* and *interest homophily*. Because of the companies' specific policies, the inventors can barely collaborate with other inventors in other companies. So we consider that geographical proximity indicates that if two inventors in one company come from the same country or geographical close places, they are more likely to collaborate. In this work, we simply consider the company and country information extracted from patents, and accordingly define the geographical proximity as a binary score (which stands for whether the two inventors in one company are from the same country or not). Interest homophily represents that significant interest overlapping between two inventors implies a higher likelihood of the two inventors to have a co-invention relationship. Formally the interest homophily is defined based on Jaccard coefficient. We did try different similarity algorithms such as cosine similarity and found that the inferring accuracy is not very sensitive to the similarity algorithms.

$$CI(v_i, v_j) = \frac{I_{v_i} \cap I_{v_j}}{I_{v_i} \cup I_{v_j}} \tag{1}$$

where I_{v_i} is a set of keywords describing inventor v_i's interests.

Regarding to *referral chaining*, the idea is that if two inventors can be connected via a short referral chain, then they are more likely to collaborate. Regarding to *Recency*, the idea is that a more recent collaboration tends to be more important. To quantify the length of a chain, we define the closeness measure of a referral link by considering the concept of recency. For any two inventors (v_i and v_j) who co-invented a set of patents S, we define the closeness of their referral link (in this case their co-inventing relationship) as:

$$R(v_i, v_j) = \sum_{d_i \in S} e^{-(\frac{t_{now} - t_{d_i}}{\lambda})} \tag{2}$$

where t_{now} is the current year; t_{d_i} is the year when patent d_i has been published; and λ is a tunable parameter. (We empirically set λ as 0.5.) This definition comes from our observation that an inventor's recent collaborator is usually closer than those of many years ago. A similar definition is also used in [22]. Based on Eq. 2, given two inventors (v_i and v_j), we first find the shortest path in terms of accumulative closeness scores on the path.

Finally, given an attribute augmented inventor network $G = (V, E, X)$, and a particular inventor v_q and the topic t, we first use the keywords contained in topic t to retrieve a list of "related" inventors using vector space model or language model [2]. We then calculate the homophily scores and the referral chaining score between inventor v_q and all inventors in the list. We combine the different scores together with the same weight and use the combination score to select top K inventors as the candidate collaborators for suggesting to inventor v_q[2].

3.2 RankFG: Ranking Factor Graph Model

After the initial retrieval of potential collaborator candidates, we now propose a ranking factor graph (RankFG) model to refine the co-invention relationships. Figure 2 shows the graphical structure of the RankFG model. For a given inventor v_q, we feed the model with the candidate list $\{v_1, v_2, v_4, v_5\}$ obtained in the initialization stage; v_3 and v_6 are two existing collaborators of v_q. The graphical model has two layers of variables: observations and latent variables. The observations are a collection of inventor pairs $\{(v_q, v_i)\}$. The corresponding latent variable y_i to each inventor pair (v_q, v_i) represents whether the two inventors have a co-invention relationship. We define two types of functions to capture the underlying factors that may influence the formation of co-invention relationships.

- **Pairwise factor function:** It captures the characteristics of the two inventors, e.g., the relationships between attributes of the two inventors. It is defined as an exponential function

$$f(v_q, v_i, y_i) = \frac{1}{Z_a} \exp\{\sum_k \alpha_k \psi_k(\mathbf{x}_q, \mathbf{x}_i, y_i)\} \tag{3}$$

  where $\psi_k(.)$ is the k^{th} feature function defined between v_q and v_i with respect to the value of y_i; α_k is the weight of the feature; $\mathbf{x}_q$ and $\mathbf{x}_i$ are attributes associated with v_q and v_i. Z_a is a normalization factor.

- **Correlation factor function:** It captures the correlation between latent variables. It is also defined as an exponential function

$$g(y_i, y_j) = \frac{1}{Z_b} \exp\{\sum_l \beta_l \phi_l(y_i, y_j)\} \tag{4}$$

  where $\phi_l(y_i, y_j)$ is the l^{th} feature function defined between y_i and y_j; β_l is the weight of the feature.

[2]The number of K is empirically set as 50 in the experiments. We will study how the number affects the inferring accuracy.

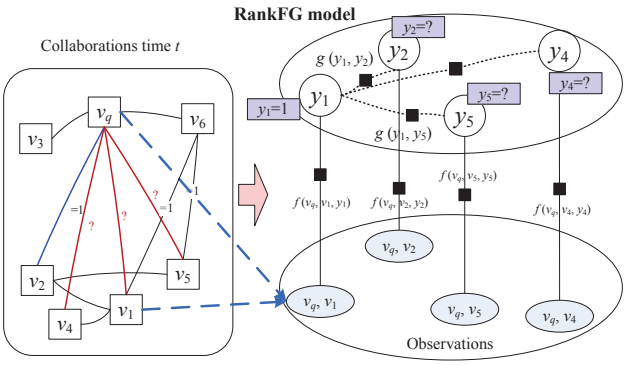


Figure 2: Graphical representation of the RankFG model. v_q **is the given inventor who intends to find patent partners;** $\{v_1, v_2, v_4, v_5\}$ **are four candidate collaborators obtained in the initialization stage;** v_3 **and** v_6 **are two existing collaborators of** v_q**;** $\{y_1, \cdots, y_4\}$ **are latent variables defined for pairs of inventors, with each representing whether the corresponding pair of inventors will have a co-invention relationship;** $f(.)$ **represents a factor function defined for each pair of inventors;** $g(.)$ **represents a correlation factor function defined between latent variables.**

In general, the pairwise factor captures the characteristics on each potential co-invention relationship and the correlation factor captures correlations between the suggested results. By integrating the defined factor functions, and also following the Markov assumption [8], we can define the following log-likelihood objective function:

$$\log P(Y|X,\theta) = \sum_{y_i \in Y} \sum_{k} \alpha_k \psi_k(\mathbf{x_q}, \mathbf{x_i}, y_i) + \sum_{v_i \sim v_j} \sum_{l} \beta_l \phi_l(y_i, y_j) - \log Z \tag{5}$$

where $Z = Z_a Z_b$ is the normalization factor; $v_i \sim v_j$ indicates that there is a (directed or indirected) correlation between v_i and v_j; $\theta = (\{\alpha\}, \{\beta\})$ are parameters to estimate.

Feature definitions. We now introduce possible ways to define the factor functions $\psi_k(\mathbf{x}_q, \mathbf{x}_i, y_i)$ and $\phi_l(y_i, y_j)$. In principle, the factor functions can be instantiated in different ways to reflect our prior knowledge (or intuitions) for different applications. It can be defined as either a binary function or a real-valued function. For example, for the pairwise factor function, we can define a binary feature function according to an attribute in X: if two inventors are from the same country and they have a co-invention relationship, then a feature $\psi(x_{qk} = x_{ik}, y_i) = 1$, where k is the attribute of country. More specifically, in total, we define nine pairwise factor functions which can be divided into four categories: Basic statistics of co-inventors, Link homophily, Interest homophily, and Correlation.

Basic statistics. We calculate a set of statistics for each potential co-inventor, the number of patents published by the inventor, the number of existing patent partners, and the number of patents published per year.

Table 1: Features defined for co-invention relationship (v_i, v_q)**.**

Feature	Description
#Patent	Number of patents published by v_i
#Co-inventor	Number of existing co-inventors of v_i
Ratio	#Patent / #Collaborator
Experience	Difference of the years of the first patent published by v_q and v_i
CS-interest	Cosine similarity of interests between v_q and v_i
#C-interests	Number of common interests between v_q and v_i
#P-interests	Percentage of common interests between v_q and v_i
Cat-similarity	Similarity between patents' categories of v_q and v_i
#C-neighbor	Number of common co-inventors of v_q and v_i
#2-C-neighbor	Number of 2-step common co-inventors of v_i and v_j
Referral	Referral chaining length between v_q and v_i
Recency	Difference of current year and last collaborated year between v_q and v_i over the referral chaining
Correlation	Represent whether two candidates have a co-invention relationship.

Input: Query inventors $Q = \{v_q\}$ with corresponding topics $\{q\}$, $G = (V, E, X)$, and the learning rate η;
Output: learned parameters θ;
$\theta \leftarrow \mathbf{0}$;
repeat
  foreach $v_q \in Q$ *and* q **do**
    //Initialization;
    $L \leftarrow$ initialization list;
    Factor graph $FG \leftarrow BuildFactorGraph(L)$;
    // Learn the parameter θ for factor graph model;
    repeat
      foreach $v_i \in order$ **do**
        Update the messages of v_i by Eqs. 8 and 9;
      end
    until *(all messages* μ *do not change)*;
    foreach $\theta_i \in \theta$ **do**
      Calculate gradient ∇_i according to Eq. 7;
      Update $\theta^{new} = \theta^{old} + \eta \cdot \nabla_i$;
    end
  end
until *converge*;

Algorithm 1: Learning algorithm for RankFG.

Link homophily. This feature represents the number of common co-inventors between the candidate inventor and the query user.

Interest homophily. We use keywords appearing in patents published by each inventor as her interest. Three features are then defined as: the number of common interests, percentage of common interests between each inventor and the query inventor, cosine similarity of the interests between two inventors.

Correlation. If two candidate inventors in the initial list have a co-invention relationship, we define a correlation factor here.

In addition, we define two features based on the referral chain length and the recency. Table 1 summarizes the main features defined in the RankFG model.

Model learning. Learning the RankFG model is to find a parameter configuration $\theta = (\{\alpha\}, \{\beta\})$ from a given historic data set, such that the log-likelihood objective function $L(\theta) = \log P(Y|X, \theta)$ can be maximized,

$$\theta^\star = \arg\max_\theta \log P(Y|X,\theta) \tag{6}$$

We can use a gradient ascent algorithm (or a Newton-Raphson method) to solve the objective function $L(\theta)$. The gradient of each parameter θ wrt $L(\theta)$ is:

$$\begin{aligned}\frac{\partial L(\theta)}{\partial \theta_i} &= \sum_j f(\mathbf{x}_i, \mathbf{y}_i) - \frac{Z(\theta)'}{Z(\theta)} \\ &= \mathbb{E}[\phi(\mathbf{x}_i, \mathbf{y}_i)] - \mathbb{E}_{P(\mathbf{x}_i, \mathbf{y}_i)}[\phi(\mathbf{x}_i, \mathbf{y}_i)]\end{aligned} \tag{7}$$

Here, we use ϕ to indicate both ϕ and ψ. The first term $\mathbb{E}[\phi(\mathbf{x}_i, \mathbf{y}_i)]$ in Eq. 7 is easy to calculate. However, in the second term, it is intractable to estimate the marginal probability $P(\mathbf{x}_i, \mathbf{y}_i)$ as the graph structure may contain cycles. There are several methods to approximately solve the problem. In our work, we choose the sum product algorithm [14] (also known as Loopy Belief Propagation (LBP) [31]). To perform the sum product algorithm, we first derive a factor graph from the original graph G. The likelihood $P(Y|X)$ is a factorization of the graph and we can flexibly add factor nodes for feature functions. Then we perform the sum-product algorithm on the factor graph to compute the approximate marginal distributions. In particular, the sum-product algorithm operates according to the sum-product update rule: the message sent from variable v to factor f (or g) is the product of local function at v with all messages received at v from all factors other than f and the message sent from factor f to variable v is the sum of all factor functions associated with v, i.e.,

$$\mu_{v\to f}(x_v) = \prod_{f^*\in N(v)\setminus f} \mu_{f^*\to v}(x_v) \tag{8}$$

$$\mu_{f\to v}(x_v) = \sum_{\sim x_v} f(\mathbf{x}_f) \prod_{v^*\in N(f)\setminus\{v\}} \mu_{v^*\to f}(v^*) \tag{9}$$

where $N(v)$ are neighborhood variables of factor f and $N(f)$ are neighborhood factors of variable v; $\mathbf{x}_f$ is the set of arguments of $f(.)$; $\sum_{\sim x}$ indicates the summation over all variables except x. Algorithm 1 describes the learning process in the RankFG model. In each iteration, the messages are updated sequentially in a certain order. We randomly select a node as the root and perform breadth-first search on the factor graph to construct a tree. We update the messages from the leaves to the root, then from the root to the leaves. The process repeats updating messages until the convergence or until the number of iterations is large enough. Based on the received messages from factors, we can calculate the marginal probabilities for each variable. Then we compute the gradient according to Eq. 7 and update the parameters by $\theta_j^{new} = \theta_j^{old} + \eta \cdot \frac{\partial L(\theta)}{\partial \theta_j}$, with η the learning step.

Recommending patent partners. Given the observed value $\mathbf{x}$ and the learned parameters θ, the patent partner recommendation is to find the most likely configuration of Y_q for a given inventor v_q. This can be obtained by:

$$Y_q = \arg\max_{Y_q} P(Y_q|X,\theta) \tag{10}$$

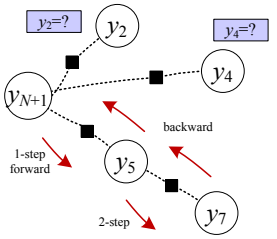


Figure 3: Illustration of the l-step message passing update.

For inference, we use the max-sum algorithm (the max version of Eqs. 8 and 9) to find the values of Y_q that maximizes the likelihood. This max-sum algorithm is similar with the sum-product algorithm, except for the message passing functions, which calculate the message according to *max* instead of *sum*.

3.3 Interactive Learning

Now, we introduce how to interactively update the ranking model based on the feedback given by the inventor. The idea is to allow the user to provide feedback to the recommended patent partners and then use an efficient algorithm to refine the ranking model by leveraging the user feedback. The algorithm supports both online interactive update and offline complete update.

For the interactive learning, the technical challenge is how to update the learned RankFG model efficiently and effectively. We design an algorithm to incrementally update the parameters. The key is still how to calculate the gradient. According to Eq. 7, for the first term, it is easy to obtain an incremental estimation, i.e.,

$$\mathbb{E}^{new}[.] = \frac{N}{N+1}\mathbb{E}^{old}[.] + \frac{1}{N+1}\sum_k \theta_k\phi_k(\mathbf{x}_{N+1}, \mathbf{y}_{N+1})$$

where $\{\theta_k\phi_k(\mathbf{x}_{N+1}, \mathbf{y}_{N+1})\}$ denotes a set of k factor functions defined for the new learning instance with the inventor's feedback ($y_{N+1} = 1$ for relevant or 0 for not). For the second term, it is again intractable to compute the marginal probabilities. Performing message passing on the complete factor graph is obviously time-consuming. We approximate this by performing a local message passing. Specifically, we first add new factor nodes (variable node and factor nodes) to the factor graph built in the model learning process. Then an l-step message passing is performed on the new factor graph starting from the new variable node y_{N+1}. More precisely, we take the new variable node y_{N+1} as the root node, begin by calculating messages $\mu_{y_{N+1}\to f}$, and then send messages to all of its neighborhood factors. We propagate the messages according to a function similar to Eqs. 8-9 up to l-depth, and then perform a backward messages passing, which propagates all messages back to the root note y_{N+1}. In this way, based on the messages sent between variables and factors, we can calculate an approximate value of the marginal probabilities of the newly added factors/variables. Accordingly, we can estimate the second term of Eq. 7.

Figure 3 illustrates the update algorithm with an l-step message passing. The interactive update is approximate and may lead to biased results. To avoid this, we perform an offline update (a complete model learning) when the number of inventors' feedback reaches a threshold.

Table 2: Statistics of four test data: IBM and Sony have much higher numbers in both measures than Intel and Exxon. Values in parenthesis are the standard deviation

Data	#inventors per patent	#patents per inventor
IBM	3.06(+/-1.68)	7.57(+/-0.76)
Intel	2.65(+/-1.49)	2.63(+/-0.69)
Sony	2.94(+/-1.47)	5.66(+/-0.82)
Exxon	2.42(+/-1.66)	3.00(+/-0.77)

4. DATA AND OBSERVATIONS

Before presenting the empirical results, we describe the datasets and observations on the datasets. All the datasets and codes used in this paper are publicly available.[3]

4.1 Data Collection

As patents represent intellectual properties of companies, co-invention relationships across different companies are merely impossible. Thus we mainly focus on analyzing co-invention relationships within companies. We have collected patent information from USPTO[4], which consists of 2,445,350 inventors and 3,770,411 patents with issued date from 1976 to 2010. We select four companies who own a large number of patents in our study: IBM, Intel, ExxonMobil and Sony. For each company, we select all its owned patents and extract all inventors and co-inventor relationships from the patents. More specifically, the data characteristics are:

IBM: with largest number of patents in the patent database, owns 55,967 patents published by 46,782 inventors. On average, each patent has three inventors, there are 269,333 co-invention relationships in total. It has a 8.26% average increase on the patent number and 11.9% increase on the number of co-invention relationships year over year.

Intel: one of the largest semiconductor chip maker corporations. We collect 18,264 patents for Intel, which include 54,095 co-invention relationships within Intel. Intel has a 18.8% increase on patent number and 35.5% average increase on the number of co-invention relationships.

ExxonMobil: the largest oil and gas corporation. For ExxonMobil, we have collected 8,505 patents and 31,569 co-invention relationships. Per day, ExxonMobil has a 11.7% increase on patent number and 13.0% increase on the number of co-invention relationships.

Sony: one of the largest electronics companies. For Sony, we obtain 19,174 patents and 53,671 co-invention relationships. Sony has a 10.6% average increase on the patent number and 14.7% average increase on the number of co-invention relationships.

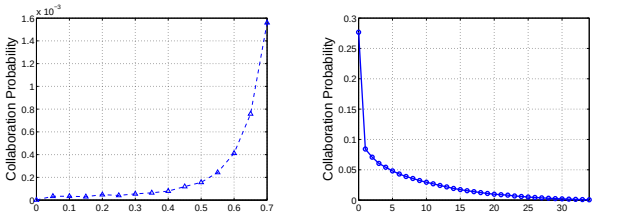
4.2 Observations

We focus on studying the interplay between several basic factors and the co-invention relationships via the following statistics:

1. Inventor history and collaboration scale vary across companies;
2. Probability that two inventors have similar interest, conditioned on whether or not they have a co-invention relationship;
3. Probability that two inventors have a co-invention relationship, conditioned on the referral chaining length;
4. Probability that two inventors maintain a co-invention relationship, conditioned on whether they have a co-invention relationship recently.

[3]http://arnetminer.org/patents/

[4]Home page of the United States Patent and Trademark Office's main web site. http://www.uspto.org/



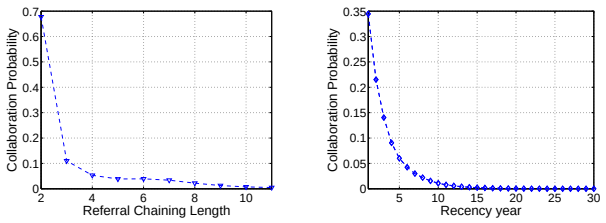
(a) Interest similarity (b) patent experience difference

Figure 4: Patent collaboration probability vs. similarity between inventors. (a) Interest similarity. X-axis: interest similarity between co-inventors; Y-axis: the collaboration likelihood of the two inventors. (b) patent experience difference: X-axis: time difference in years from the first patent published by the two inventors; Y-axis: the collaboration probability of the two inventors.

Inventor history and collaboration scale. We present the average number of inventors per patent and the average number of patents per inventor in Table 2. The former shows the collaboration scale, i.e., how many collaborators work together on a single patent. The latter shows the invention history, i.e., how many patents an inventor typically generates. In particular, IBM and Sony have much higher measures than Exxon and Intel. Both measures turn out to affect the recommendation performance as we will present in the next section. Intuitively, the higher these two measures are, the better the recommendation will be.

Similar inventors tend to collaborate. Figure 4 shows that, in general, similar inventors tend to have a co-invention relationship. For each inventor v, a feature vector is generated based on the frequencies of words in those patents published by the inventor. The value of each element in the vector is normalized by TFIDF [2]. Then *Interest Similarity* is computed as Cosine similarity between the two feature vectors.

Figure 4(a) shows that the collaboration probability increases dramatically as the similarity increases, which confirms the homophily feature in our model. Figure 4(b) shows how the collaboration probability changes with the difference of patent experience. We use the year from the first patent published by the inventor as the estimate of her patent experience. We show that inventors seem to like to find others of with similar patent experiences to develop patent with. The likelihood of two inventors with little difference (difference=0) to collaborate is almost 10 times more than ones with 10 year difference.



(a) Referral chaining length (b) Recency

Figure 5: (a) Patent collaboration probability vs. referral chaining length; X-axis: referral chaining length between two inventors; Y-axis: collaboration likelihood of the two inventors. (b) Patent collaboration conditioned on recency; X-axis: the difference of the current year and the last collaborated year; Y-axis: collaboration likelihood of the two inventors.

Inventors with a short referral chain tend to collaborate. We study how likely two inventors collaborate conditioned on the referral chaining length between them. Figure 5(a) gives two observations. The first one is "Your friends' friends are very likely to be your friends", which is consistent with the phenomenon of in the social balance theory [6]. The collaboration probability drops sharply with the increase of the referral chaining length. When the length increases to 4, the likelihood becomes less than $\frac{1}{10}$ of that at the length of 2. Another observation is that the analysis validates the theory of "six degrees of separation". More than 96% of pairs of inventors can be connected in less than six referral steps.

Inventors tend to continue collaborating Figure 5(b) illustrates that people tend to maintain collaborations. The likelihood of two inventors to maintain a recent co-invention relationship (in recent 2 years) is more than 10 times higher than re-establish a co-invention relationship of five years before.

5. RESULTS AND ANALYSIS

We present quantitative performance of the proposed approach with the comparison in each settings. After that, we make some analysis and discussion.

5.1 Experimental Setup

We use the data set described in §4 to evaluate the proposed model and to compare with several baseline methods. We evaluate our models RankFG and RankFG+ on the patent datasets from each of the four companies. We partition the 35 years' data into the first 25 years as training and the last 10 years as testing.

Comparison methods. We compare the following methods for suggesting co-invention relationships:

Content Similarity (Content): it calculates similarity between inventors based on patents published by the two inventors. It uses word frequencies in one's published patents as features and calculates the similarity score $Sim(v_q, v)$ between the query inventor v_q and another inventor v based on the words. The patent partner recommendation is then made based on the similarity score.

Collaborative Filtering (CF): it leverages the existing co-invention relationships to make the recommendation. The basic idea is that if an inventor A has similar patent partner as another inventor B, A is likely to collaborate with B's other patent partners. We employ a memory-based collaborative filtering algorithm [5], in which recommendations are made for a query user v_q using the following formula:

$$CF\_score(v_q, v) = \frac{1}{N} \sum_{v_i \in V^S} I(v_i, v) r(v_q, v_i)$$

where $r(v_q, v_i)$ describes one element of the pairwise similarity between inventors, which is typically measured by Pearson correlation coefficient or cosine similarity based on links; the indicator variable $I(v_i, v')$ is 1 if the inventor v_i collaborated with v' and 0 otherwise; N denotes the size of $|V^S|$.

Hybrid: it considers a linear combination of the scores from the Content and the CF methods, specifically,

$$Hybrid(v_q, v) = \mu CF\_score(v_q, v) + (1 - \mu) Sim(v_q, v)$$

where μ is a balance parameter. We empirically set it as 0.5.

SVMRank: The above three methods do not use training data. We then consider a learning approach which use the same training data as our RankFG. As for the learning model, we use SVM Light [12].

RankFG: The proposed method, which trains a RankFG model to suggesting patent partners.

RankFG+: it uses the proposed RankFG model with 1% interactive users' feedbacks.

Evaluation measures. We use the following performance metrics: Precision(P): P@5, P@10, P@15, P@20, Mean Average Precision(MAP) and Recall(R): R@100 [2].

All codes are implemented in C++ and JAVA, and all the evaluations are performed on an x64 machine with E7520 1.87GHz Intel Xeon CPU and 128GB RAM. The operation system is Microsoft Windows Server 2008 R2 Enterprise. All methods yields good performances. The training time needed for prediction by all algorithm on all data sets less than 5 minutes.

5.2 Performance Analysis

We compare the performance of all methods for suggesting co-invention relationships in the four companies. Table 3 lists the performance of comparison methods. The proposed method (RankFG) shows clearly better performance than the baseline methods. On average, RankFG achieves a +3.2-20.3% improvement compared with other methods in terms of MAP. SVM-Rank also uses training data; however, it does not consider the correlation among suggested inventors, thus performes worse than our method RankFG and RankFG+. We can also see that RankFG+ obtains a clear improvement (2-4% in terms of MAP) than RankFG, which confirms the effectiveness of interactive learning. The results also suggest that in different companies the co-inventing patterns are very different. Some interesting observations include

- Content based recommendation method *Content* performs better than network based method *CF* in the patent recommendation, due to the sparsity of the underlying network. This is very different from other social networks such as publication collaboration network, where CF often performs better than Content.

Table 3: Performance of patent partner recommendation for different companies (%).

Data	Method	P@5	P@10	P@15	P@20	MAP	R@100
IBM	Content	23.0	23.3	18.8	15.6	24.0	33.7
	CF	13.8	12.8	11.3	11.5	21.7	36.4
	Hybrid	13.9	12.8	11.5	11.5	21.8	36.7
	SVM-Rank	13.3	11.9	9.6	9.8	22.2	43.5
	RankFG	31.1	**27.5**	25.6	22.4	40.5	46.8
	RankFG+	**31.2**	**27.5**	**26.6**	**22.9**	**42.1**	**51.0**
Intel	Content	16.4	12.6	11.3	10.3	20.1	22.9
	CF	4.8	6.0	5.6	5.6	10.1	18.3
	Hybrid	4.9	6.2	5.9	5.7	11.4	18.3
	SVM-Rank	8.4	13.6	14.5	14.1	20.4	32.0
	RankFG	**17.3**	**14.2**	12.9	12.1	26.0	**33.0**
	RankFG+	**17.3**	**14.2**	**14.7**	**14.9**	**29.3**	**33.0**
Sony	Content	23.0	16.5	13.5	11.6	23.9	34.9
	CF	10.0	17.0	16.0	16.5	5.2	20.9
	Hybrid	9.0	7.2	6.0	6.8	14.1	20.2
	SVM-Rank	28.0	24.0	16.5	14.2	34.8	44.2
	RankFG	37.0	34.5	30.7	**26.3**	39.9	48.8
	RankFG+	**38.4**	**38.5**	**31.0**	**26.3**	**43.3**	**55.5**
Exxon	Content	25.0	20.3	18.1	16.1	23.6	16.1
	CF	3.3	5.0	5.4	5.6	8.1	12.8
	Hybrid	3.3	5.3	5.4	5.9	8.9	20.7
	SVM-Rank	25.5	**23.3**	**21.6**	21.5	28.2	29.3
	RankFG	26.6	19.2	20.0	23.9	28.5	32.3
	RankFG+	**27.1**	19.4	20.7	**24.7**	**30.4**	**34.0**

Table 4: Performance comparison of patent partner recommendation with online interactive learning and offline complete learning for the four companies.

Data	Algorithm	P@5	P@10	P@20	MAP	R@100
IBM	Interactive	31.2	27.5	22.9	42.1	51.0
	Complete	31.6	27.9	23.1	42.3	52.1
Intel	Interactive	17.3	14.2	14.9	29.3	33.0
	Complete	17.3	15.2	15.4	29.5	34.4
Sony	Interactive	38.4	38.5	26.3	43.3	55.5
	Complete	40.0	39.7	26.5	43.5	58.4
Exxon	Interactive	27.1	19.4	24.7	30.4	34.0
	Complete	27.2	19.4	24.7	30.4	34.5

- All methods achieve better recommendation accuracy in IBM and Sony than in Intel and Exxon. This is because IBM and Sony have higher number of inventors per patent as well as the higher number of patents per inventor. Longer patent history per inventor and more common collaborations help improve the recommendation performance.

In all cases, the proposed methods RankFG and RankFG+, because of incorporating the data correlations and the user feedback, obtain consistent better performance than all the baselines.

In Figure 6, we give a detailed comparison between the online interactive learning and the offline complete learning for the proposed RankFG model. The offline complete learning re-trains the RankFG model and the interactive learning only performs the learning on neighborhood nodes. Thus the offline complete learning can be considered as the upper bound of the interactive learning. We see that the proposed interactive learning achieves a close performance to the complete learning. Notice that the interactive learning only is usually finished in 3 seconds, 1/100 of the running time used for complete training (even on a relative small data set). Figure 6 shows the performance of interactive learning by varying the number of user feedbacks. We see that the performance of online interactive learning becomes very close to the offline complete learning when the number of feedbacks increases to 1% of the total collaborations, which confirms the effectiveness of the proposed interactive learning method.

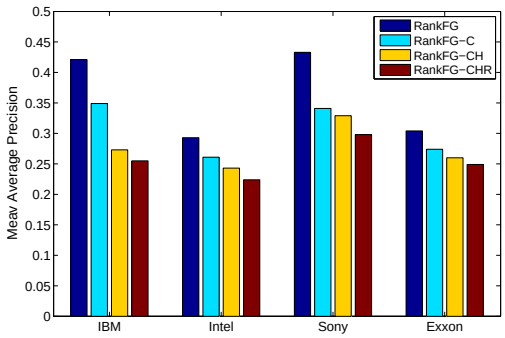
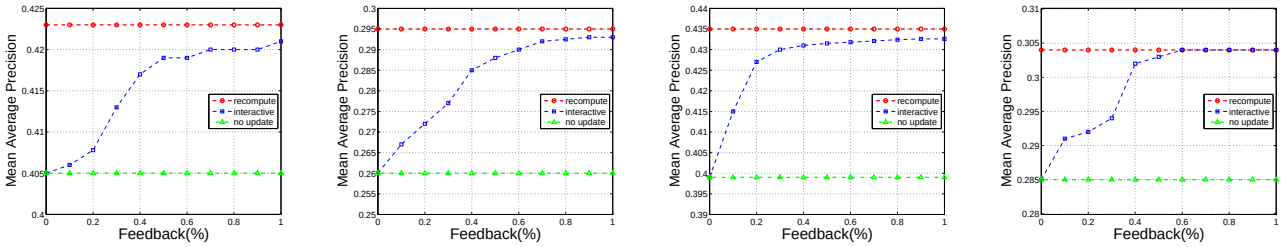


Figure 7: Factor contribution analysis: RankFG-C stands for ignoring referral chaining factor functions. RankFG-CH stands for ignoring both referral chaining and homophily. RankFG-CHR stands for further ignoring recency.

Factor contribution analysis. We now analyze how different factors can help us. In the RankFG model, we consider five major factors: homophily (H), referral chaining length (C), recency (R), basic statistics, and correlation. Here we examine the contribution of different factors defined in our RankFG model. Specifically, we take basic statistics and correlation as the basic features in the model and study the contribution of the other three factors. Figure 7 shows the Mean Average Precision score over the different data sets. In particular, RankFG-C represents that we remove referral chaining based features from our model and RankFG-CH denotes that we further remove homophily features. It can be clearly observed that the performance drops when ignoring each of the factors. We often observe that for recommending patent partners the referral chaining length is more important than others. The analysis confirms that our model works well when combining all the features together.

Convergence analysis. We conduct an experiment on the effect of the number of iterations of the loopy belief propagation. Figure 8 shows the convergence analysis results of the learning algorithm RankFG. We see on all the test cases, the learning algorithm can converge less than 10 iterations. After about 7 iterations, the performance becomes stable.



(a) IBM (b) Intel (c) Sony (d) Exxon

Figure 6: Interactive learning with different numbers of feedbacks.

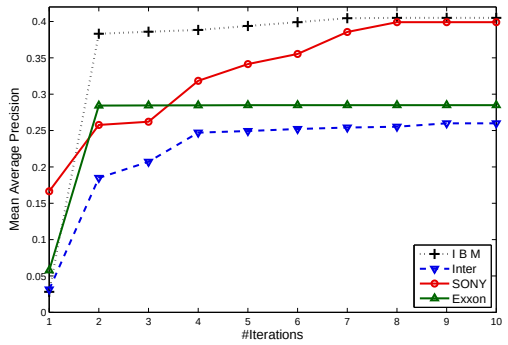


Figure 8: Convergence analysis of learning algorithm for different cross domains.

This suggests that learning algorithm is very efficient and has a good convergence property.

6. RELATED WORK

Collaboration plays an important role in social networks. While a large body of works focuses on expert finding [3, 28, 33], friends recommendations [9, 13, 22], and expertise matching [21, 29], few publications consider the problem of patenting partner recommendation. Kautz et al. [13] introduced a system called ReferralWeb which attempts to combine social networks for collaborative filtering. Roth et al. [22] leveraged the implicit social graphs that are formed by users' interactions to suggest recipients in the Google Gmail system. Shi et al. [24] introduced rank heterogeneous content to make recommendations, and Sculley et al. [23] presented a method to rank that combines regression and ranking. Quan et al. [32] considered social relations: membership, friendship to recommend top-n people. However, none consider the cross-field problem. Mimno et al. [21] and Tang et al. [29] studies the problem of paper reviewers recommendation, a subtask of expert finding. The proposed algorithms can be leveraged for collaborator recommendations. Lappas et al. [15] investigated the problem of finding a team of experts to fulfill a given task in social networks. They theoretically proved that the problem is NP-hard and propose two instantiation models to approximately solve this problem.

There are a few systems for patent search and analysis such as Google Patent, WikiPatent, PatentMiner, FreePatentsOnline, Patents, PatentLens, and PriorArtSearch. However, most of these systems focus on search and provide limited micro-level analytic functions. Few systems provide the function of patenting partner recommendations. For research on the patent data, Tang et al. [26] propose a topic-driven patent analysis and mining method. Jin et al. [11] proposed a method to evaluate the quality of patents. Liu et al. [18] and Mann [19] studied how to estimate patent quality from the perspective of court validity rulings or the number of forward citations. Tseng et al. [30] introduced a series of text mining techniques for patent analysis, including text segmentation and summary extraction. However, all these works have focused on analyzing patent content, but ignore the collaborative relationships between inventors.

Our work is also related to link prediction, which is one of the core tasks in social networks. For example, Liben-Nowell et al. [17] presented a unsupervised method for link prediction. Backstrom et al. [1] proposed a supervised random walk algorithm to estimate the strength of social links. Leskovec et al. [16] employed a logistic regression model to predict positive and negative links in online social networks. Cradall et al. [4] studied how to infer the friendship from geographic coincidence data. Hopcroft et al. [10] studied the extent to which the formation of a reciprocal relationship can be predicted in a dynamic network. Eric Gilbert et al. [7] presented a predictive model that maps social media data to tie strength. Tang et al. [25] developed a framework for classifying the type of social relationships by learning across heterogeneous networks. In this work, we focus on studying the underlying patterns that influence the formation of co-invention relationships and propose a novel rank factor graph model to incorporate the discovered patterns for recommending co-invention relationships.

7. CONCLUSION

In this paper, we study the problem of recommending patenting partners in enterprise social networks. We precisely define the problem and propose a ranking factor graph (RankFG) model for suggesting co-invention relationships. Through a careful observable investigation, we discover several interesting patterns. We incorporate the discovered patterns into the proposed RankFG model. We evaluate our proposed model on large patent data sets and the experimental results show that the proposed model can significantly improves the performance for recommending co-invention relationships compared with several alternative methods.

Finding the right patenting partner is an important step toward producing successful patents in an enterprise social network. There are many potential research topics in this di-

rection. It would be interesting to further consider subtopics in the recommendation. The user may be not aware of the subtopics of the query topic. It would be helpful to extract subtopics and to allow the user to refine the recommendation results according to the subtopics. It is also interesting to study how the topic trend of a company influence the enterprise co-invention relationships.

Acknowledgements. The work is supported by the National Basic Research Program of China (No. 2011CB302302), Natural Science Foundation of China (No. 61222212, No. 61073073), and a fund for Fast Sharing of Science Paper in Net Era by CSTD.

8. REFERENCES

[1] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM'11*, pages 635–644, 2011.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, 1999.

[3] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR'2006*, pages 43–55, 2006.

[4] D. J. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg. Inferring social ties from geographic coincidences. *PNAS*, 107:22436–22441, Dec. 2010.

[5] A. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. In *WWW'07*, pages 271–280,2007.

[6] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010.

[7] E. Gilbert and K. Karahalios. Predicting tie strength with social media. In *CHI'09*, pages 211–220, 2009.

[8] J. M. Hammersley and P. Clifford. Markov field on finite graphs and lattices. *Unpublished manuscript*, 1971.

[9] P. Heymann and H. Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical Report 2006-10, Stanford University, April 2006.

[10] J. E. Hopcroft, T. Lou, and J. Tang. Who will follow you back? reciprocal relationship prediction. In *CIKM'11*, pages 1137–1146, 2011.

[11] X. Jin, S. Spangler, Y. Chen, K. Cai, R. Ma, L. Zhang, X. Wu, and J. Han. Patent maintenance recommendation with patent information network model. In *ICDM'11*, pages 280–289, 2011.

[12] T. Joachims. *Making large-Scale SVM Learning Practical*. MIT-Press, 1999.

[13] H. Kautz, B. Selman, and M. Shah. Referral web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[14] F. R. Kschischang, B. J. Frey, and H. andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE TOIT*, 47:498–519, 2001.

[15] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *KDD'09*, pages 467–476, 2009.

[16] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *WWW'10*, pages 641–650, 2010.

[17] D. Liben-Nowell and J. M. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[18] Y. Liu, P.-y. Hseuh, R. Lawrence, S. Meliksetian, C. Perlich, and A. Veen. Latent graphical models for quantifying and predicting patent quality. In *KDD'11*, pages 1145–1153, 2011.

[19] R. Mann. A new look at patent quality. *American Law and Economics Association Annual Meetings*, 2008.

[20] M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[21] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *KDD'07*, pages 500–509, 2007.

[22] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom. Suggesting friends using the implicit social graph. In *KDD'10*, pages 233–242, 2010.

[23] D. Sculley. Combined regression and ranking. In *KDD'10*, pages 979–988, 2010.

[24] Y. Shi, D. Ye, A. Goder, and S. Narayanan. A large scale machine learning system for recommending heterogeneous content in social networks. In *SIGIR'11*, pages 1337–1338, 2011.

[25] J. Tang, T. Lou, and J. Kleinberg. Inferring social ties across heterogenous networks. In *WSDM'12*, pages 743–752, 2012.

[26] J. Tang, B. Wang, Y. Yang, P. Hu, Y. Zhao, X. Yan, B. Gao, M. Huang, P. Xu, W. Li, and A. K. Usadi. Patentminer: topic-driven patent analysis and mining. In *KDD '12*, pages 1366–1374, 2012.

[27] J. Tang, S. Wu, J. Sun, and H. Su. Cross-domain collaboration recommendation. In *KDD'12*, pages 1285–1293, 2012.

[28] J. Tang, J. Zhang, R. Jin, Z. Yang, K. Cai, L. Zhang, and Z. Su. Topic level expertise search over heterogeneous networks. *Machine Learning Journal*, 82(2):211–237, 2011.

[29] W. Tang, J. Tang, T. Lei, C. Tan, B. Gao, and T. Li. On optimization of expertise matching with various constraints. *Neurocomputing*, 76(1):71–83, 2012.

[30] Y.-H. Tseng, C.-J. Lin, and Y.-I. Lin. Text mining techniques for patent analysis. *Inf. Process. Manage.*, 43:1216–1247, September 2007.

[31] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS'01*, pages 689–695, 2001.

[32] Q. Yuan, L. Chen, and S. Zhao. Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In *RecSys'11*, pages 245–252, 2011.

[33] J. Zhang, J. Tang, and J. Li. Expert finding in a social network. In *DASFAA'07*, pages 1066–1069, 2007.