Unsupervised Link Prediction Using Aggregative Statistics on Heterogeneous Social Networks

Tsung-Ting Kuo*, Rui Yan†, Yu-Yang Huang*, Perng-Hwa Kung*, Shou-De Lin*
* National Taiwan University, Taiwan
† Peking University, China
{d97944007, b98901083, r00922048, sdlin}@csie.ntu.edu.tw, r.yan@pku.edu.cn

ABSTRACT

The concern of privacy has become an important issue for online social networks. In services such as Foursquare.com, whether a person likes an article is considered private and cannot be disclosed; only the aggregative statistics of articles (i.e., how many people like this article) is revealed. This paper tries to answer a question: can we predict the opinion holder in a heterogeneous social network without any labeled data? This question can be generalized to an *unseen-type link prediction with aggregative statistics* problem. This paper devises a novel unsupervised framework to solve this problem, including three main components: (1) a three-layer factor graph model and three types of potential functions; (2) a ranked-margin learning algorithm for parameter tuning; and (3) a two-stage inference algorithm for link prediction. Finally, we evaluate our method on four diverse prediction scenarios using four datasets: preference (Foursquare), repost (Twitter), response (Plurk), and citation (DBLP). We further exploit nine unsupervised models to solve this problem as baselines. Our approach not only wins out in all scenarios, but on the average achieves 9.90% AUC and 12.59% NDCG improvement over the best competitors.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications – *data mining*; J.4 [**Computer Applications**]: Social and Behavioral Sciences; E.2 [**Data**]: Data Storage Representations – *linked representations*.

General Terms

Algorithms, Experimentation

Keywords

Link prediction, Social network mining, Heterogeneous social network, Probabilistic graphical model

1. INTRODUCTION

Most of the social network services allow users to express their opinions (e.g., "like" or "+1") to messages posted by other people. Such individual opinions are usually valuable: companies can identify a specific customer's preference, and government can recognize the will or desire of target influential person.

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

However, due to privacy concern, opinion holders are sometimes hard to be determined. An example is Foursquare, one of the most popular location-based social network websites. In Foursquare, users can post tips to certain venues of their interest, and other people may "*like*" the tips. Nevertheless, the information about which user likes which tip is generally not available to public due to the website's privacy policy.

Another example is Pinterest.com, which is a pinboard-style photo sharing website. In Pinterest, users can "*like*" or "*repin*" others' images, but only a little portion of such information is available due to internal limitation of Pinterest (only first 24 "*like*" and first 8 "*repin*" are shown on the webpage). Thus, it is difficult to gather information about each individual's opinion under such circumstances.

Fortunately, *aggregative statistics* of opinions are usually available in such websites. For example, the total count of "*like*" of each tip in Foursquare is accessible, and the total count of "*like*" and "*repin*" of an image in Pinterest is also obtainable. Such aggregative statistics are important because it is usually the only available clue to understand the quality of certain item without violating the policy rule. Hence, this paper tries to address a problem: can we predict the individual opinions (e.g., whether a user likes a tip) using the aggregative statistics together with other information in a heterogeneous social network?

We generalize the question to an *unseen-type link prediction with aggregative statistics* problem. The term *unseen* is used because we assume it is not possible to obtain which person likes which tip from data (therefore, such "*like*" link can be regarded as a kind of relationship that is previously unseen). From link prediction point of view, one can assume there is *no* labeled training data available, to predict the type of a link.

An example we use through this paper is a network gathered from Foursquare (Figure 1). There are 7 nodes and 7 links with 3 node types (users, items, and categories) and 3 link types (be-friend-of, own, and belong-to). We want to predict the existence of "*like*" links (e.g., whether user u_2 likes item r_2 or not) using the aggregative statistics (e.g., total like count of the item r_2 is $t(r_2) = 1$). Note that the links of "*like*" type is unseen, which means we do not see such link at all in training data.

Most of the link prediction literatures aim at predicting links of *seen* types (i.e., some labeled historical links are observable as the training data) [18, 20, 33], thus cannot be applied to our problem. Some researchers predict links of unseen types using external node group information [15], but those information are not always available. As in the Foursquare example, the only available information in our problem is the aggregative statistics.

Nevertheless, our problem is non-trivial due to the following three challenges:

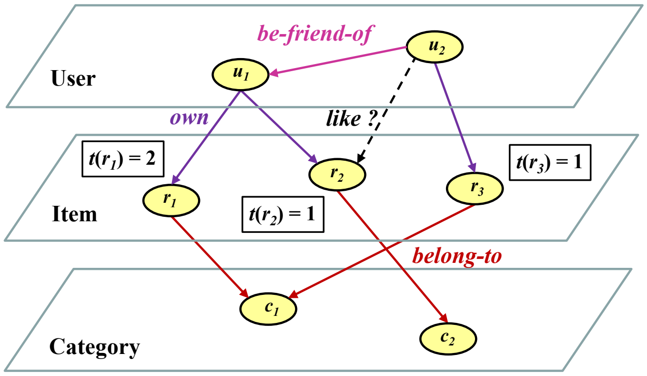

Figure 1. The unseen-type link prediction with aggregative statistics problem in a heterogeneous social network.

- **Lack of labeled data**. The absence of labeled training data prevents us from performing parameter learning in a straightforward way.
- **Diverse information**. In a heterogeneous social network, the information of different types of nodes and links are diverse but correlated with each other. A suitable model has to carefully model such correlation with aggregative statistics.
- **Sparsity of links**. Since the type is unseen, presumably the possible candidate-link count approaches $O(n^2)$ where n is the total number of nodes. When n is large, this can cause serious sparsity problem, while finding the links in such a large space can be very challenging.

In this paper, we try to address these challenges by proposing a novel unsupervised probabilistic graphical model. First, we devise a factor graph model with three layers of random variables (candidate, attribute, and count) to infer the existence of unseen-type links. Second, we define three types of potential functions (attribute-to-candidate, candidate-to-candidate, and candidate-to-count) to integrate diverse information into the factor graph model. Third, we design a ranked-margin learning algorithm to automatically tune the parameters using aggregative statistics. Finally, we design a two-stage inference algorithm to update the candidate-to-count potential functions, and calculate the prediction results as close to the given aggregative statistics as possible.

The main contributions of this study are as below:

- We propose and formulate a novel yet practical problem to predict the links of unseen-type using aggregative statistics in heterogeneous social networks.
- We devise an unsupervised learning framework to solve the above-mentioned problem. Note that the framework we proposed can be exploited not only for probabilistic graphical models, but for all kinds of general situations where only aggregative statistics are available for learning.
- We evaluate our method on four diverse scenarios using different heterogeneous social network datasets: preference prediction (Foursquare), repost prediction (Twitter), response prediction (Plurk), and citation prediction (DBLP). We also apply nine unsupervised models for this problem as baseline. Our model not only wins in all scenarios, but also achieves on the average 9.90% AUC and 12.59% NDCG improvement over the best comparison methods.

2. PROBLEM FORMULATION

We start by formulating the problem.

Definition 1. ***Heterogeneous social network*** *$N = (V, E, \Omega_V, \Omega_E)$ is a directed graph, where V is a set of nodes, Ω_V is a set of node labels, Ω_E is a set of link labels,* and *$E \subseteq V\times \Omega_E \times V$ is a set of links.*

The function $type(v) \rightarrow l_V$ maps node v onto its node label $l_V \in \Omega_V$. Similarly, given a triplet < *source*, *link-label*, *target* > as a link, the function $type(e) \rightarrow l_E$ maps link e onto its link label $l_E \in \Omega_E$.

For the example shown in Figure 1, there are 7 nodes and 7 links, with Ω_V = { "*user*", "*item*", "*category*" } and Ω_E = { "*be-friend-of*", "*own*", "*belong-to*" }. For brevity, we denote $U \subseteq V$ as the set of node for type = "*user*", $R \subseteq V$ for type = "*item*", and $C \subseteq V$ for type = "*category*".

The relationship between node labels and link labels can be enumerated. For instance, a user u may "*be-friend-of*" another user v (i.e., < u, "*be-friend-of*", v >); a user u may "*own*" an item r (i.e., < u, "*own*", r >), and an item r may "*belong-to*" a category c (i.e., < r, "*belong-to*", c >).

It should be noted that the number of items, $|R|$, is equivalent to the total number of "*own*" links, and is also equivalent to the total number of "*belong-to*" links (i.e., each item can only be owned by one user, and can only belong to one category).

Definition 2. ***Unseen-type links*** *is a set of links with a special type "?"; links of such type do not appear in a given heterogeneous social network. That is, unseen-type links Φ = { φ | φ = < source, "?", target >, type(source) $\in \Omega_V$, type(target) $\in \Omega_V$, "?" $\notin \Omega_E$ }.*

For the example in Figure 1, the unseen-type links denote the "*like*" behavior. That is, Φ = { < u, "*like*", r > } denotes the set of links that user u likes item r. We use < u, r > to denote the candidate pairs of unseen-type links, and there are $|U| \cdot |R| = 6$ plausible candidate pairs in Figure 1.

Definition 3. ***Aggregative statistic*** *is the total unseen-type link count of a target node. In other words, the aggregative statistic of a node $v \in V$ is $\sigma(v, \Phi)$ = | { φ | φ = < source, "?", target > $\in \Phi$, target = v } |, which is a non-negative integer.*

In our example, the aggregative statistic of an item $r_2 \in R$ is $\sigma(r_2, \Phi)$ = | { φ | φ = < u, "*like*", r > $\in \Phi$, $r = r_2$ } | = 1.

Definition 4. ***Aggregative statistics for heterogeneous social network*** *$T(N, \Phi)$ = { < v, $\sigma(v, \Phi)$ > | $v \in V$ } is the set of aggregative statistics of the unseen links for a heterogeneous social network N.*

In Figure 1, the aggregative statistics for heterogeneous social network N is $T(N, \Phi)$ = { < r_1, 2 >, < r_2, 1 >, < r_3, 1 > }.

Based on above definitions, we formulate the *unseen-type link prediction with aggregative statistics* problem as follows: *given a heterogeneous social network N and corresponding aggregative statistics $T(N, \Phi)$, predict the existence of unseen-type links Φ.*

The relational schema for our example is shown in Figure 2: given the heterogeneous social network (3 types of nodes and 3 types of edges) and aggregative statistics of "*like*", predict whether each < u, "*like*", r > exists or not, where $u \in U$ *and* $r \in R$.

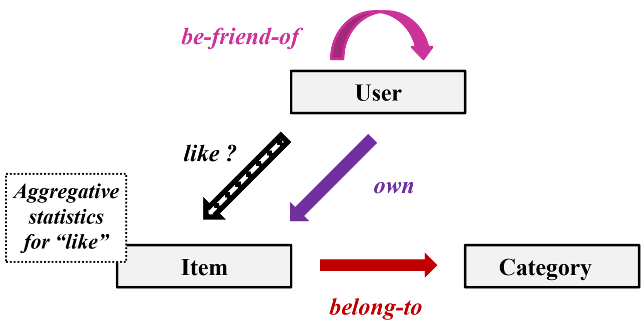

Figure 2. Relational schema of the unseen-type link prediction with aggregative statistics problem shown in Figure 1.

3. METHODOLOGY

In the first subsection, we propose to solve this problem using a probabilistic model. Then, we use an illustrative example to demonstrate our model. Finally, we describe a novel learning algorithm utilizing the aggregative statistics to learn the model parameters, as well as a two-stage inference algorithm to predict unseen-type links.

3.1 Factor Graph Model with Aggregative Statistics (FGM-AS)

To handle this problem, we propose a novel probabilistic graphical model: *factor graph model with aggregative statistics* (FGM-AS), as shown in Figure 3. There are three layers of variables in FGM-AS:

- **Candidate:** the binary random variables Y in the *candidate* layer represent all unseen-type links to be predicted. They either exist (positive) or not exist (negative). Each candidate y_i can be regarded as a pair of user and item, $< u, r >$. Also note that some y's might point to the same users while some might share the same item.
- **Attribute:** the random variables A in the *attribute* layer carry attribute information (e.g., a_1 represents the degree of the source node and a_2 represents the degree of the target node) of the candidate links.
- **Count:** the random variables T in the *count* layer encode the aggregative statistics of the items. Note that t is a one-to-one mapping of an item r, but a one-to-many mapping of y because there are some y's sharing the same item (e.g., candidate y_1 and y_2 point to the same t_1 as they have the same item r).

Together with the random variables, we also propose three types of potential functions:

- **Attribute-to-candidate functions:** we define this type of potential function as a linear exponential function

$$f(A, y_i) = \frac{1}{Z_\alpha} \exp\{\alpha \cdot f'(A, y_i)\} \quad (1)$$

  where $f'(A, y_i)$ is a vector of functions representing the associations between a candidate and its attributes (see subsection 3.2.1 for a detailed example), α is a vector of the corresponding weights, and Z_α is a normalization factor. Note that each candidate y can connect to multiple attributes.

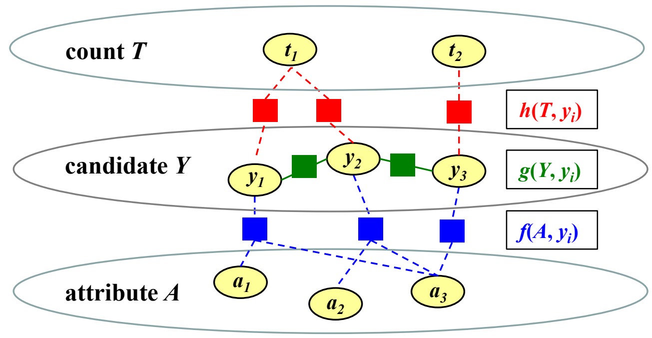

Figure 3. Factor graph model with aggregative statistics (FGM-AS)

- **Candidate-to-candidate functions:** this type of potential function is defined as

$$g(Y, y_i) = \frac{1}{Z_\beta} \exp\{\beta \cdot g'(Y, y_i)\} \quad (2)$$

  where $g'(Y, y_i)$ is a vector of functions representing the relationships between candidate random variables (see subsection 3.2.2 for a detailed example), β is a vector of weights, and Z_β is a normalization factor.

- **Candidate-to-count functions:** this type of potential function is defined as

$$h(T, y_i) = \frac{1}{Z_\gamma} \exp\{\gamma \cdot h'(T, y_i)\} \quad (3)$$

  where $h'(T, y_i)$ is a vector of functions representing the constraints of aggregative statistics (see subsection 3.2.3 for a detailed example), γ is a vector of weights, and Z_γ is a normalization factor. To be more precise, this type of potential functions adhere to the following condition: the sum of predicted marginal probability of the candidate random variables of each item should be as close to the total count of that item as possible.

According to the FGM-AS model, when the candidates, attributes and counts are known, we can define the joint distribution as

$$P(A, T, Y) = \prod_i f(A, y_i) \cdot g(Y, y_i) \cdot h(T, y_i) \quad (4)$$

Therefore, the marginal probability of candidate random variable y_i being positive (e.g., *like*) is

$$P(A, T, Y, y_i) = \sum_j P(A, T, Y, y_j), y_j \in Y / \{y_i\} \quad (5)$$

The marginal probability $P(A, T, Y, y_i = 1)$ is the desired output in our problem, as it tells us for $y_i = < u, r >$, how likely u likes r.

3.2 An Illustrative Example of FGM-AS

We believe that FGM-AS is a general graphical model for solving the unseen-type links prediction problem. The three layers of random variables and the three types of potential functions can be flexibly defined for different application context. Here we use FGM-AS to predict whether a user likes an item or not. Figure 4 illustrates an example of FGM-AS, which is built from the heterogeneous social network shown in Figure 1. The three layers of random variables are defined as:

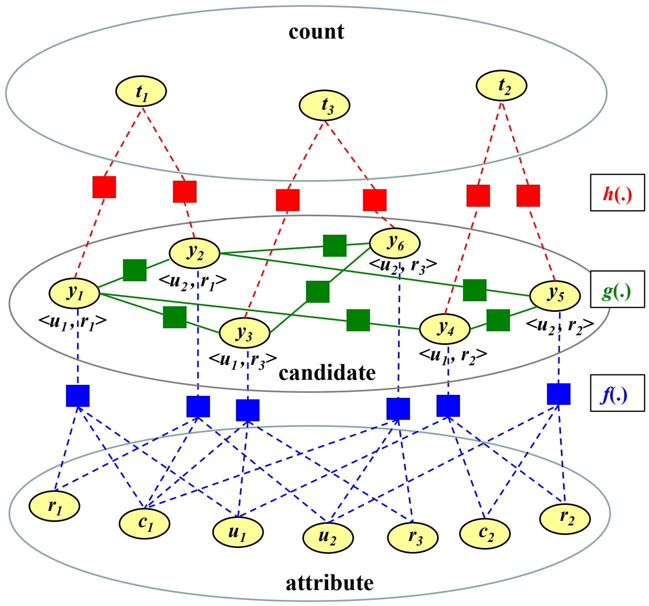

Figure 4. An example of FGM-AS based on Figure 1's network.

- **Candidate:** candidate random variables $Y = \{ y_i \mid i = 1, 2, \ldots, |U| \cdot |R| \}$ represent the set of plausible links $< u, r >$ to be predicted. In other words, each pair $y_i = < u, r >$ indicates whether the user u likes the item r. For example, $y_1 = < u_1, r_1 >$ represents whether user u_1 likes item r_1. Note that u_1 is not necessarily the owner of r_1.

- **Attribute:** attribute random variables $A = U \cup R \cup C$ contain three groups of information: users $U = \{ u_1, u_2, \ldots, u_{|U|} \}$, items $R = \{ r_1, r_2, \ldots, r_{|R|} \}$, and categories $C = \{ c_1, c_2, \ldots, c_{|C|} \}$. We use $u(y_i)$ to denote the corresponding user, $r(y_i)$ to denote the corresponding item, and $c(y_i)$ to denote the corresponding category of y_i.

- **Count:** count random variables $T = \{t_1, t_2, \ldots, t_{|R|} \}$ represent the aggregative statistics (total like count) of each item. Note that $|T| = |R|$ because t is a one-to-one mapping of r. We use $t(y_i)$ to denote the corresponding count of y_i.

The design of the three potential functions is described in the following three subsections.

3.2.1 *Attribute-to-Candidate Function*

According to Equation (1), we define $f'(A, y_i) = < f_{UF}(u(y_i)), f_{IO}(u(y_i), r(y_i)), f_{CP}(c(y_i)) >$. The functions f_{UF}, f_{IO} and f_{CP} are based on user friendship, item ownership, and category popularity, which are defined below:

- **User friendship (UF) function:** $f_{UF}(u(y_i))$ = the number of friends of $u(y_i)$. The intuition behind UF is that we believe the number of friends of a user can influence his / her tendency to like an item. In Figure 1, $f_{UF}(u(y_1)) = f_{UF}(u_1) = 1$, because user u_1 has only one friend (which is u_2).

- **Item ownership (IO) function:** $f_{IO}(u(y_i), r(y_i)) = 1$ if $r(y_i)$ is owned by $u(y_i)$, otherwise 0. The intuition behind IO is that we believe whether a user likes an item or not depends significantly on whether this item is owned by this user. In Figure 1, $f_{IO}(u(y_1), r(y_1)) = f_{IO}(u_1, r_1) = 1$, because u_1 owns r_1.

- **Category popularity (CP) function:** $f_{CP}(c(y_i))$ = the number of items in the whole dataset that belongs to the same category as $c(y_i)$. The intuition behind CP is that users tend to like items belonging to a hot category (i.e., category which contains many items). In Figure 1, $f_{CP}(c(y_1)) = f_{CP}(c_1) = 2$, because there are two items belonging to c_1.

3.2.2 *Candidate-to-Candidate Function*

According to Equation (2), we define $g'(Y, y_i) = < \Sigma_j g_{OI}(y_i, y_j), \Sigma_j g_{FI}(y_i, y_j), \Sigma_j g_{OF}(y_i, y_j), \Sigma_j g_{CC}(y_i, y_j) >, y_j \in Y / \{y_i\}$. The functions g_{OI}, g_{FI}, g_{OF} and g_{CC} are based on owner, friend, owner-friend, and co-category relationships, which are defined as follows:

- **Owner-identification (OI) function**: $g_{OI}(y_i, y_j) = 1$ if $< u(y_i)$, "*own*", $r(y_i) > \in E$, $< u(y_j)$, "*own*", $r(y_j) > \in E$, and $u(y_i) = u(y_j)$; otherwise 0. The intuition is that an owner tends to like all his / her items. For example in Figure 1, u_1 likes both r_1 and r_2, because u_1 owns both items. Therefore, there will be a relation between y_1 and y_4 in Figure 4.

- **Friend-identification (FI) function**: $g_{FI}(y_i, y_j) = 1$ if $< v$, "*own*", $r(y_i) > \in E$, $< v$, "*own*", $r(y_j) > \in E$, $u(y_i) = u(y_j)$, and $v \in friend(u(y_i))$; otherwise 0. The intuition is that a person may like friend's items. For example, u_2 likes both r_1 and r_2, because u_2's friend u_1 owns both items. Therefore, there will be a relation between y_2 and y_5.

- **Owner-friend (OF) function:** $g_{OF}(y_i, y_j) = 1$ if $< u(y_i)$, "*own*", $r(y_i) > \in E$, $r(y_i) = r(y_j)$, and $u(y_i) \in friend(u(y_j))$; otherwise 0. The intuition is that if an owner likes his / her own item, his / her friends tend to like the item too. For example, if u_1 likes his / her item r_2, then his / her friend u_2 tends to like r_2 as well. In other words, there will be a relation between y_4 and y_5.

- **Co-category (CC) function**: $g_{CC}(y_i, y_j) = 1$ if $< u(y_i)$, "*own*", $r(y_i) > \in E$, $u(y_i) = u(y_j)$, and $c(y_i) = c(y_j)$; otherwise 0. The intuition is: the extent an owner likes the item will be similar to the extent of the owner likes other items in the same category. For example, if u_1 tends to like item r_1, then u_1 may also like r_3, because r_1 and r_3 are in the same category c_1. Thus, there is a relation between y_1 and y_3.

3.2.3 *Candidate-to-Count Function*

According to Equation (3), we define $h'(T, y_i) = < h_{CT}(y_i, t(y_i)) >$. The function h_{CT} *is* defined as:

$$h_{CT}(y_i, t(y_i)) = 1 - \left| \frac{t(y_i) - \sum_{y_j \in Y, r(y_j) = r(y_i)} P(A, T, Y, y_j = 1)}{|U|} \right| \qquad (6)$$

The summation term in Equation (6) sums up all the probabilities of a certain item $r(y_i)$ being liked by each user, which we hope to be as close to the observed "*like*" count of this item as possible. Thus, the difference of this term and $t(y_i)$ represents how close the prediction to the known aggregative statistics is. We divide this difference by $|U|$ for normalization purpose. Ideally, the difference is 0, and thus $h_{CT}(y_i, t(y_i)) = 1$. Also, $0 \leq h_{CT}(y_i, t(y_i)) \leq 1$.

It should be noted that $P(A, T, Y, y_j = 1)$ are not random variables anymore but the posterior probability of them. Therefore, the conventional exact or approximated inference methods cannot be applied directly. To update accordingly, we design a two-stage inference algorithm, which is described at the end of section 3.3.

3.3 Ranked-Margin Learning for FGM-AS

The key factor that contributes to the success of FGM-AS lies in the algorithm's capability of learning the parameters without labeled data. Here we discuss the main idea. Given a parameter configuration $\theta = (\alpha, \beta, \gamma)$ and based on Equation (1) – (4), the joint probability $P(A, T, Y)$ can be written as

$$P(A,T,Y) = \frac{1}{Z}\prod_i \exp\{\theta \cdot (f'(A,y_i), g'(Y,y_i), h'(T,y_i))\}$$

$$= \frac{1}{Z}\exp\{\theta \cdot \sum_i s(y_i)\} = \frac{1}{Z}\exp\{\theta \cdot S\} \tag{7}$$

where all potential functions for a y_i is written as $s(y_i) = < f'(A, y_i), g'(Y, y_i), h'(T, y_i) >$, $Z = Z_\alpha Z_\beta Z_\gamma$, and $S = \Sigma_i s(y_i)$.

Now, we will discuss how to learn the parameters of the model. Traditionally the idea of *maximum-likelihood estimation* (MLE) can be exploited and algorithms such as EM can be applied to achieve this goal. Alternatively for a factor graph, algorithms such as gradient decent can be exploited to greedily search in the parameter space. However, in our scenario, the absence of labels eliminates the possibility of exploiting MLE strategy for learning. Moreover, even if one can somehow come up with certain approximated objective to be maximized in the M-step of EM, the total number of hidden variables in this graph grows to $|U| \cdot |R|$, which can lead to very high computational cost for models such as EM or other sampling methods for parameter learning.

To effectively and efficiently perform the learning task, we propose a novel idea to maximize the *ranked-margin* of the instances, incorporating the aggregative statistics into the objective function. The intuition is to assume the count for an item $r(y_i)$ is $t(y_i)$, which means that among all candidate users, only $t(y_i)$ of them like this object.

Therefore, during learning we want to adjust the parameter so that the top $t(y_i)$ users have very high probabilities of liking this item while the rest have very low probabilities of liking it. To realize this idea, we propose to do the following. For each item r, first rank each user u_i based on the marginal probability of $y = < u_i, r >$. Then, let $P(Y_r^{upper})$ be the average positive marginal probabilities for the top $t(y_i)^{th}$ candidate pairs, and $P(Y_r^{lower})$ be the average marginal probabilities for the rest of the candidate pairs, for all y_i of which $r(y_i) = r$. Finally, given $t(y_i)$, we want to adjust the parameters to maximize

$$Diff(Y_r^{margin}) = P(Y_r^{upper}) - P(Y_r^{lower}) \tag{8}$$

An extreme example is that the marginal probability of the top $t(y_i)$ candidate pairs are all 1, while the rest are all 0. In this case $Diff(Y_r^{margin}) = 1 - 0 = 1$. Another extreme example is the opposite, which results in $Diff(Y_r^{margin}) = -1$. Thus, $-1 \le Diff(Y_r^{margin}) \le 1$.

Based on the above idea and Equation (8), we define the log-likelihood objective function to be maximized as

$$O(\theta,r) = \log P(Y_r^{upper}) - \log P(Y_r^{lower})$$

$$= \log \sum_{Y_r^{upper}} \frac{1}{Z}\exp\{\theta \cdot S\} - \log \frac{1}{Z}\sum_{Y_r^{lower}} \exp\{\theta \cdot S\}$$

$$= \log \sum_{Y_r^{upper}} \exp\{\theta \cdot S\} - \log \sum_{Y_r^{lower}} \exp\{\theta \cdot S\} \tag{9}$$

Input: FGM-AS, learning rate η

Output: $P(A, T, Y, y_i = 1)$ for all $y_i \in Y$

Initialize all elements in parameter configuration $\theta = 1$

repeat

Run inference method using current θ to obtain $P(A, T, Y, y_i = 1)$

Compute potential function values S according to Eq. (1) – (7)

foreach $r \in R$ do

Compute gradient $\frac{\partial O(\theta,r)}{\partial \theta}$ using S according to Eq. (10)

$\theta = \theta + \eta \cdot \frac{\partial O(\theta,r)}{\partial \theta}$

end

until *convergence*

Algorithm 1. Ranked-margin learning algorithm.

Besides the intuitiveness of Equation (8) with respect to the count as mentioned, there are two other advantages of using Equation (9) as our objective function. First, it should be noted that computing the normalization factor Z in Equation (7) is very time-consuming. But by using Equation (9), we can essentially eliminate Z to avoid the high computational cost during learning. Second, the gradient of Equation (9) can be obtained through sampling using any inference algorithm (as shown below).

To maximize the objective function, we exploit an idea similar to the Stochastic Gradient Descent (SGD) method, as shown in Algorithm 1. We calculate the gradient and update the parameters for each item iteratively until convergence, then move on to the next item (η is the learning rate of our algorithm). The gradient for each parameter θ and item r is

$$\frac{\partial(\theta,r)}{\partial\theta} = \frac{\partial\left(\log \sum_{Y_r^{upper}} \exp\{\theta \cdot S\} - \log \sum_{Y_r^{lower}} \exp\{\theta \cdot S\}\right)}{\partial\theta}$$

$$= \frac{\sum_{Y_r^{upper}} \exp\{\theta \cdot S\} \cdot S}{\sum_{Y_r^{upper}} \exp\{\theta \cdot S\}} - \frac{\sum_{Y_r^{lower}} \exp\{\theta \cdot S\} \cdot S}{\sum_{Y_r^{lower}} \exp\{\theta \cdot S\}}$$

$$= \mathbb{E}_{P\theta(Y_r^{upper})} S - \mathbb{E}_{P\theta(Y_r^{lower})} S \tag{10}$$

where $\mathbb{E}_{P\theta(Y_r^{upper})} S$ and $\mathbb{E}_{P\theta(Y_r^{lower})} S$ are two expectations of S. The value of S can be obtained naturally using approximated inference algorithms, such as Gibbs Sampling or Contrastive Divergence. It should be noted that the proposed ranked-margin algorithm can be exploited not just for graphical model, but also for other learning models as long as the gradient of the expected difference can be calculated.

In Algorithm 1, we need to perform an inference algorithm on the factor graph, to obtain the marginal probability of each candidate pair y. Also, after the parameters are learned, we need to apply the inference algorithm again to compute the marginal probability, representing how likely the person likes the item. Unfortunately, such inference cannot directly be done as $P(A, T, Y, y_i = 1)$ in Equation (6) requires the posterior probabilities of y.

Input: FGM-AS, parameter configuration θ

Output: $P(A, T, Y, y_i = 1)$ for all $y_i \in Y$

Initialize all $y_i = 0$, all $h(T, y_i) = 1$

stage 1

Calculate $f(A, y_i)$ and $g(Y, y_i)$ according to Eq. (1), (2)

Run an inference method using θ to obtain $P(A, T, Y, y_i = 1)$

stage 2

Calculate $h(T, y_i)$ using $P(A, T, Y, y_i = 1)$ according to Eq. (3), (6)

Run an inference method using θ to obtain final $P(A, T, Y, y_i = 1)$

Algorithm 2. Two-stage Inference algorithm.

Thus, we design a two-stage inference algorithm (Algorithm 2). In the first stage, we perform general inference method using $f(A, y_i)$ and $g(Y, y_i)$ only (by assigning all $h(T, y_i) = 1$) to initialize $P(A, T, Y, y_i = 1)$. In the second stage, we compute $h(T, y_i)$ using $P(A, T, Y, y_i = 1)$, and then perform inference one more time. This way, we integrate the posterior information into the inference process.

4. EXPERIMENTS

Here we want to verify the generalization of our model by testing whether it can be applied to dataset in four different scenarios. We also want to verify the usefulness of the potential functions.

4.1 Scenarios and Datasets

We study the following four types of scenarios of the unseen-type link prediction problem, each with a real-world publicly available dataset. The statistics of the datasets are shown in Table 1.

- **Preference prediction**. In location-based social network services, we are interested in predicting whether users will like a tip of venue (i.e., add the tip into their to-do list). We extract the social network website *Foursquare* as the dataset for evaluation and consider *to-do* as the unseen-type link. We select all venues located in New York, collect all tips for these venues, and identify users who posted the tips. We regard *venues* as categories, and *tips* as items. Note that due to the privacy policy in Foursquare, only the total to-do count of each tip is revealed. There is very limited number (i.e., 15,758) of unseen-type links revealed, which become ground truth for evaluation (not seen in training).
- **Repost prediction**. In social network websites, we are interested in predicting whether users will re-blog or retweet a post. Therefore, we use *Twitter* as the dataset, which is collected from [7]. Twitter is one of the most famous micro-blog website, and has been used to verify several models with different purposes [7, 8, 24]. In this study, we consider *retweet* as the unseen-type link. We keep users who have two or more friends, and have tweeted or retweeted more than once. Then, we perform stemming to identify 100 most popular *terms* in tweets as categories while each *tweet* is regarded as an item. For example, if a user v posts a tweet r, and later another user u retweets this tweet (with the "*RT@*" keyword), we consider an unseen-type link exists from u to r.
- **Response prediction**. In micro-blog services, we are interested in predicting whether users will respond to a post. We use *Plurk* dataset in this scenario. Plurk is popular micro-blog service in Asia with more than 5 million users, and has been used in studies of diffusion prediction [13], diffusion model evaluation [12], and mood classification [2]. This dataset is collected from 01/2011 to 05/2011. In this study, we consider *response-to-message* as the unseen-type link. We manually identify the 100 most popular *topics* as categories, and regard *messages* as items. For example, if a person v posts a message r, and later another person u responds to this message, we consider an unseen-type link exists from u to r.
- **Citation prediction**. In academic indexing and searching services, we are interested in predicting whether researchers will cite a paper. Therefore, we use *DBLP* [17] dataset collected from ArnetMiner [26], version 5. In this study, we consider *citation-to-paper* as the unseen-type link. We first perform stemming, and then identify the 100 most popular *terms-in-titles* as categories, and regard *papers* as items. For example, if a researcher v published a paper r, and later another researcher u cites r, we consider an unseen-type link exists from u to r. Also, we consider two researchers as friend if they have been co-authors of at least one paper in the past.

Table 1. Statistics of the datasets

Property		Foursquare	Twitter	Plurk	DBLP
Node	User	71,634	69,026	190,853	102,304
	Item	180,684	55,375	352,376	221,935
	Category	16,961	100	100	100
	Total	**269,279**	**124,501**	**543,329**	**324,339**
Link	Be-friend-of	724,378	21,979,021	2,151,351	245,391
	Own	180,684	55,375	352,376	221,935
	Belong-to	180,684	55,375	352,376	221,935
	Unseen	15,758	79,918	804,404	123,479
	Total	**1,101,504**	**22,169,689**	**3,660,507**	**812,740**

Table 2. Mapping of the random variables for the datasets

Random Variable		Foursquare	Twitter	Plurk	DBLP
Candidate	y	To-do	Retweet	Response	Citation
Attribute	u	User	User	User	User
	r	Tip	Tweet	Message	Paper
	c	Venue	Term	Topic	Keyword
Count	t	To-dos per tip	Retweets per tweet	Responses per message	Citations per paper

The mapping of the information in the four abovementioned datasets to the random variables in FGM-AS is shown in Table 2. Note that in the above four datasets (Foursquare, Twitter, Plurk, and DBLP), we hide all unseen-link information as ground truth to evaluate our proposed framework. Also note that we obfuscate personal information in all of the datasets.

It should be noted that the unseen-type links used as ground truth are actually sparse comparing to all nodes and relations. For example, in Twitter dataset, the unseen-to-candidate ratio, $|Unseen| / (|User| \cdot |Item|)$, is merely 0.00002. Thus, predicting unseen-type links for these datasets is a very challenging task.

4.2 Comparison Methods

We use nine unsupervised model for comparison. The first three methods are single attribute-to-candidate functions: UF, IO, and CP. Another six methods are as follows (note that all methods are computed on the whole heterogeneous social network):

- **Betweenness Centrality (BC)**. This method is used to measure an edge's importance in a network. The BC value of an edge equals to the number of shortest paths from all nodes to all others that pass through that edge. For each candidate pair, we add a *pseudo* unseen-type link in network. Then, we generate BC values of pseudo links as their prediction scores.

- **Jaccard Coefficient (JC)**. This method is used to directly compute the relatedness of an user u to an item r, which is defined as $|$ *neighbor*(u) ∩ *neighbor*(r) $|$ / $|$ *neighbor*(u) ∪ *neighbor*(r) $|$. This score is used to predict whether u likes r.

- **Preferential Attachment (PA)**. This method bases on an assumption that popular users tends to like popular items. Therefore, it is defined as $|$ *neighbor*(u) $|$ · $|$ *neighbor*(r) $|$, which is used as the prediction scores.

- **Attractiveness (AT)**. This method is designed to compute user-to-user attractiveness using aggregated count [32]. We transform it to predict unseen-type links. It first computes owner-item attractiveness P_{vr} from owner v to item r as

$$P_{vr} = \frac{\sigma(r,\Phi)}{\sum_{c(r')=c(r)} \sigma(r',\Phi)} \tag{11}$$

where Φ is the set of "*like*" links, and $\sigma(r, \Phi)$ is the aggregative statistic of item r, as defined in Section 2. Then, it compute the user-owner attractiveness P_{uv} from user u to v as

$$P_{uv} = 1 - \prod_{r} (g_{uv} \cdot (1 - P_{vr})) \tag{12}$$

where $g_{uv} = 1$ if u and v are friends, otherwise 0. To perform link prediction, we further compute user-item attractiveness P_{ur} (the probability of user u likes item r) as

$$P_{ur} = P_{uv} \cdot P_{vr} \tag{13}$$

- **PageRank with Priors (PRP)**. This method executes PageRank algorithm [31] for $|R|$ times, once for each item. For specific item r, we set the prior of the item node to 1, and priors of all other nodes to 0. Thus, the probability of user u likes item r is modeled using PageRank score of the user node u. We set the random restart probability as 0.15.

- **AT-PRP**. We combine the Attractiveness and PageRank with Priors methods by using the weight of the links. That is, in the heterogeneous social network, we add a link for each $< u, r >$ pair, with weight equals to P_{ur}. We then normalize all weights of outgoing links to sum up to 1, and run PageRank with Priors as mentioned above.

4.3 Settings

Because of the sparsity of unseen-links in ground-truth, we use Area Under ROC Curve (AUC) [5] [19] and Normalized Discounted Cumulative Gain (NDCG) [10] to evaluate our proposed method. For each item, we rank all the candidate pairs based on their predicted positive marginal probabilities, and then compare the rankings with the ground-truths to obtain AUC and NDCG scores. Finally, we average the scores over all items.

We select Loopy Belief Propagation (LBP) as our base inference method [23], utilize MALLET [21] for LBP inference, and apply LingPipe [1] for stemming. We use JUNG [22] to compute betweenness centrality and PageRank with Priors algorithms.

Table 3. Experiment results of our framework (FGM-AS) and all comparison methods (in percentage).

Method	Foursquare		Twitter		Plurk		DBLP	
	AUC	NDCG	AUC	NDCG	AUC	NDCG	AUC	NDCG
UF	76.74	21.66	73.49	18.87	71.08	35.01	70.28	25.07
IO	81.31	51.60	69.98	18.93	69.86	35.33	68.51	23.84
CP	74.03	20.56	67.38	17.15	70.69	36.13	69.52	24.22
BC	67.01	21.26	67.65	18.97	69.81	31.47	64.17	21.10
JC	64.30	26.75	65.65	21.05	70.05	35.40	69.96	28.24
PA	72.28	27.09	62.30	16.39	67.42	32.68	71.41	26.12
AT	82.57	44.54	76.95	20.28	69.62	39.29	70.95	28.48
PRP	57.27	17.93	62.41	16.56	69.12	33.64	61.83	21.25
AT-PRP	71.06	22.38	68.17	18.11	70.99	36.03	67.86	24.27
INFER	86.77	70.60	79.11	24.80	74.23	40.24	86.84	41.75
LEARN	**98.61**	**80.44**	**81.29**	**25.87**	**74.42**	**42.61**	**87.29**	**41.84**
Improve	**16.04**	**28.84**	**4.34**	**4.82**	**3.34**	**3.32**	**15.88**	**13.36**

In FGM-AS, we set all zero potential function values to a small constant (0.000001), and use learning rate $\eta = 0.0001$. We run all experiments on a Linux server with AMD Opteron 2350 2.0GHz Quad-core CPU and 32GB memory.

4.4 Results

The results of different methods using AUC and NDCG are shown in Table 3. The LEARN method is to exploit Algorithm 1 to perform learning and Algorithm 2 for inference, while INFER is to exploit Algorithm 2 for inference without learning. In all cases, LEARN performs best. Note that INFER outperforms all baselines, and LEARN provides further improvement than INFER. Averaging over the four datasets, our framework (LEARN) are 9.90% AUC and 12.59% NDCG better than the best comparison methods. LEARN achieves best result for Foursquare dataset, with improvement of 16.04% in AUC and 28.84% in NDCG.

From Table 3, we see that the performance distinction between the three attribute-to-candidate functions, UF, IO, and CP, varies depending on the dataset used. We believe that these three functions are complementary to each other, and can be ensemble to contribute to our integrated framework. BC does not work well in all experiments, JC performs well for Twitter in terms of NDCG, and PA performs well for DBLP in terms of AUC. On the other hand, AT is in general the strongest comparison method (performs best among comparison methods in both metrics for all four datasets); PRP in general does not perform well; AT-PRP ranks just between AT and PRP. Our framework consistently outperforms these comparison methods significantly. Based on the above experiment results, we believe our framework can be a general method to solve the unseen-type link prediction problem.

4.5 Candidate-to-Candidate Verification

In previous subsection, we evaluate the attribute-to-candidate functions and compare them to our proposed framework. However, the candidate-to-candidate functions cannot be evaluated independently (i.e., without attribute-to-candidate functions). Therefore, we verify the feasibility of the four functions, namely OI, FI, OF, and CC, by performing a simple analysis in our datasets. First, we set all "*own*" links as "*like*" links. As shown in Figure 1, we set $< u_1$, "*like*", $r_1 >$, $< u_1$, "*like*", $r_2 >$, and $< u_2$, "*like*", $r_3 >$, as positive prediction. Then, we apply the above four candidate-to-candidate functions to extend the predicted links.

Table 4. Verification results of candidate-to-candidate functions (in percentage), Pre. = precision, Rec. = recall.

Function	Foursquare		Twitter		Plurk		DBLP	
	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
OI	**2.14**	37.50	0.00	0.00	0.00	0.00	0.00	0.00
FI	0.33	55.00	0.00	0.00	**3.25**	33.55	1.53	60.68
OF	0.35	40.00	0.21	20.00	3.23	37.31	1.53	60.68
CC	0.20	2.50	**0.74**	20.00	1.36	18.76	**2.64**	86.65
All	0.48	**95.00**	0.31	**40.00**	2.02	**51.43**	1.64	**94.66**

For example, considering OF function, there will be a link between < u_1, "*like*", r_2 > and < u_2, "*like*", r_2 >. Because < u_1, "*like*", r_2 > is positive (i.e., it is originally an "*own*" link), we predict < u_2, "*like*", r_2 > as positive based on OF.

We compare the result of candidate-to-candidate functions using precision and recall with the unseen-type links in ground-truth, as shown in Table 4. We also ensemble the four functions and examine the effectiveness of the combination (the *All* row). All of the candidate-to-candidate functions has low precision (less than 4%), but have some extend of recall (especially *All*). For Foursquare and DBLP datasets, the recall of *All* reaches as high as 95.00% and 94.66%, respectively. It should be noted that OI performs bad for Twitter, Plurk and DBLP datasets, but provides some improvement for Foursquare dataset. On the other hand, FI seems to be of little use for Twitter dataset, but it does provide information for other three datasets. Therefore, we regard these four candidate-to-candidate functions as complementary to each other, and can be ensemble to contribute to our framework.

5. RELATED WORK

In this section, we discuss some of works related to unsupervised unseen-link prediction framework using aggregative statistics.

5.1 Link Prediction

Our problem is effectively link prediction in heterogeneous social network. Link prediction is a well-studied task in social network analysis, and is characterized by graph topology, testing how *proximal* nodes are to each other [18]. Many features have been tested and developed for homogeneous network, using different graph topological properties [20]. However, such approaches do not consider the sparsity and diversity of heterogeneous social network. Feature design for heterogeneous social network was recently explored [33], casting as a supervised learning task [14]. One area of research interest is to predict actual popularity of a microblog (e.g., tweet) in a social media. In this case, the task is formulated as a supervised learning problem, where it can be binary (e.g., whether a tweet will be retweeted or not) or multi-class (e.g., assign the prediction of how a tweet will be retweeted by popularity category) classification problem [8] [24]. Another approach applies probabilistic model on social media response prediction [35]. This work essentially incorporates collaborative filtering accounting user and item (i.e., tweets) features, but still require training data. Another related area is to predict the link from user to venue (i.e., point of interest recommendation) using geographic information [34]. However, such method fails to utilize effects of information propagation in social network.

Regarding unsupervised link prediction, there have been works such as cold-start link prediction [15], transfer learning [6], and triad census [4]. They are fundamentally different from this work. Cold-start link prediction requires category information, and works only on homogeneous network. Transfer learning assumes another domain of labeled data is available. Triad census does not consider the aggregative statistics information in the networks. Pure unsupervised heterogeneous social network link prediction explores different context of the data by examining probabilistically the topological features of the reweighed path [3] [33]. However, these works usually predict links between two entities of the same type, holding the underlying assumption that birds of a feather flock together. Our work tries to predict links between two different types (usually users and items) where such assumption is not likely to hold.

5.2 Factor Graph and Max-Margin Learning

Factor graph [11] is a unified framework for general probabilistic graphical models. Recently, factor graphs have been widely adopted to resolve various problems [9] [25] [29] [30]. Among these applications, factor graphs are suitable for social relationship prediction tasks. [29] proposed a time-constrained unsupervised probabilistic factor graph (TPFG) to model the advisor-advisee relationship using time information. Triad Factor Graph (TriFG) model [9] incorporates the factor graph representations and social theories over triads into a semi-supervised model. [25] investigates the relationship prediction problem on heterogeneous social networks. Previous attempts are extended and integrated into a transfer-based factor graph (TranFG) model. However, these methods either need additional external information or do not consider the aggregation of statistics during computation.

Several margin-based learning methods on probabilistic graphical models have been proposed. Previous methods require the ground-truth labels to figure out the proper direction of parameter update. For example, [27] formulates the parameter fitting problem as a quadratic program and performs Sequential Minimal Optimization (SMO) learning to solve the problem. For max-margin methods solving similar problems such as structural support vector machines [28], the ground-truth is also needed to fit these models. However, in our problem, it is the aggregative statistics instead of the ground-truth labels that are given. Therefore, our framework maximizes the *ranked-margin* instead of traditional margin.

6. CONCLUSION AND FUTURE WORK

Mining on social networks using incomplete information has gained its own value due to its applicability, as in the real world we cannot always expect all the information to be observable. In this paper, we demonstrate that the unseen-type link prediction can be solved using an unsupervised framework through exploiting the aggregative statistics. We show how various information sources in the heterogeneous social network can be modeled all together in a factor graph, propose a novel learning algorithm to learn the parameters using aggregated counts, and devise an inference algorithm to predict unseen-type links using learnt parameters. With such framework, one can now derive hypotheses on the individual behavior using the group statistics. Especially, under the growing concern of personal privacy preservation, we believe our framework provides a mean for applications that tries to distill personal preference information from the statistics. On the other hand, in the area of biomedicine, our framework can be applied to identify novel protein-disease relationships, given clinical aggregated observations.

Future work includes extending the current ranked-margin learning framework to other types of models such as discriminant classification and clustering. Also, for some networks (e.g., Foursquare), the very small amount of observable links may be utilized to extend our framework to a semi-supervised setting to further improve the prediction accuracy. Our framework can also be applied to more application scenarios and networks. Next, temporal information may be considered, which further empowers our framework to deal with dynamic networks. Finally, our work may also be extended to predict positive / negative links (e.g., applying methods described in [16]) using aggregative statistics.

7. ACKNOWLEDGMENTS

We thank anonymous reviewers for valuable time, comments, and suggestions. This work was supported by National Science Council, National Taiwan University and Intel Corporation under Grants NSC 101-2911-I-002-001, NSC 101-2628-E-002-028-MY2 and NTU 102R7501.

8. REFERENCES

[1] Alias-i. (2008). *LingPipe 4.1.0*. Available: http://alias-i.com/lingpipe

[2] M.-Y. Chen, *et al.* 2010. Classifying mood in plurks. in *Proceedings of the 22nd Conference on Computational Linguistics and Speech Processing (ROCLING 2010)*, 2010.

[3] D. Davis, *et al.* 2011. Multi-relational Link Prediction in Heterogeneous Information Networks. in *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining*, ed. Washington, DC, USA: IEEE Computer Society, 2011, pp. 281-288.

[4] D. Davis, *et al.* 2011. Multi-relational Link Prediction in Heterogeneous Information Networks. presented at the Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining, 2011.

[5] J. Davis and M. Goadrich. 2006. The relationship between Precision-Recall and ROC curves. presented at the Proceedings of the 23rd international conference on Machine learning, Pittsburgh, Pennsylvania, 2006.

[6] Y. Dong, *et al.* 2012. Link Prediction and Recommendation across Heterogeneous Social Networks. in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, ed, 2012, pp. 181 -190.

[7] W. Galuba, *et al.* 2010. Outtweeting the twitterers - predicting information cascades in microblogs. presented at the Proceedings of the 3rd conference on Online social networks, Boston, MA, 2010.

[8] L. Hong, *et al.* 2011. Predicting popular messages in Twitter. presented at the Proceedings of the 20th international conference companion on World wide web, Hyderabad, India, 2011.

[9] J. Hopcroft, *et al.* 2011. Who will follow you back?: reciprocal relationship prediction. in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 1137-1146.

[10] K. Jrvelin and J. Keklinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.,* vol. 20, pp. 422-446, 2002.

[11] F. R. Kschischang, *et al.* 2001. Factor Graphs and the Sum-Product Algorithm. *IEEE TRANSACTIONS ON INFORMATION THEORY,* vol. 47, 2001.

[12] T.-T. Kuo, *et al.* 2011. Assessing the Quality of Diffusion Models Using Real-World Social Network Data. in *Conference on Technologies and Applications of Artificial Intelligence*, 2011.

[13] T.-T. Kuo, *et al.* 2012. Exploiting latent information to predict diffusions of novel topics on social networks. presented at the Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2, Jeju Island, Korea, 2012.

[14] T.-T. Kuo and S.-D. Lin. 2011. Learning-based concept-hierarchy refinement through exploiting topology, content and social information. *Information Sciences,* vol. 181, pp. 2512-2528, 2011.

[15] V. Leroy, *et al.* 2010. Cold start link prediction. in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, DC, USA, 2010, pp. 393-402.

[16] J. Leskovec, *et al.* 2010. Predicting positive and negative links in online social networks. presented at the Proceedings of the 19th international conference on World wide web, Raleigh, North Carolina, USA, 2010.

[17] M. Ley. 2002. The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives. in *SPIRE*, 2002.

[18] D. Liben- Nowell and J. Kleinberg. 2007. The link- prediction problem for social networks. *Journal of the American society for information science and technology,* vol. 58, pp. 1019-1031, 2007.

[19] R. Lichtenwalter and N. V. Chawla. 2012. Link Prediction: Fair and Effective Evaluation. presented at the ASONAM, 2012.

[20] L. Lu and T. Zhou. 2011. Link prediction in complex networks: a survey *Physica A: Statistical Mechanics and its Applications,* vol. 390, pp. 1150-1170, 2011.

[21] A. K. McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. ed, 2002.

[22] J. O'Madadhain, *et al.* 2003. The JUNG (Java Universal Network/Graph) Framework. 2003.

[23] J. Pearl. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*: Morgan Kaufmann, 1988.

[24] S. Petrovic, *et al.* 2011. Rt to win! predicting message propagation in twitter. in *5th ICWSM*, 2011.

[25] J. Tang, *et al.* 2012. Inferring social ties across heterogenous networks. in *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012, pp. 743-752.

[26] J. Tang, *et al.* 2008. ArnetMiner: extraction and mining of academic social networks. presented at the Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, Las Vegas, Nevada, USA, 2008.

[27] B. Taskar, *et al.* 2004. Max-margin Markov networks. in *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*, 2004, p. 25.

[28] I. Tsochantaridis, *et al.* 2005. Large Margin Methods for Structured and Interdependent Output Variables. *J. Mach. Learn. Res.,* vol. 6, pp. 1453-1484, 2005.

[29] C. Wang, *et al.* 2010. Mining advisor-advisee relationships from research publication networks. in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 203-212.

[30] Z. Wang, *et al.* 2012. Cross-lingual knowledge linking across wiki knowledge bases. in *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 459-468.

[31] S. White and P. Smyth. 2003. Algorithms for estimating relative importance in networks. in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 266-275.

[32] H.-H. Wu and M.-Y. Yeh. 2013. Influential Nodes in One-Wave Diffusion Model for Location-Based Social Networks. in *Proc. of the 17th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD-2013)*, 2013.

[33] Y. Yang, *et al.* 2012. Link Prediction in Heterogeneous Networks: Influence and Time Matters. in *ICDM*, ed, 2012.

[34] M. Ye, *et al.* 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. presented at the Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, Beijing, China, 2011.

[35] T. R. Zaman, *et al.* 2010. Predicting information spreading in twitter. in *Workshop on Computational Social Science and the Wisdom of Crowds, NIPS*, 2010, pp. 17599-601.