On Exploiting Social Relationship and Personal Background for Content Discovery in P2P Networks

Xiao Han[a,*], Ángel Cuevas[a,b], Noel Crespi[a], Rubén Cuevas[b], Xiaodi Huang[c]

[a] *Institut-Mines Télécom, Télécom SudParis, 9 rue Charles Fourier, 91011 Evry Cedex France*
[b] *Universidad Carlos III de Madrid, Av de la Universidad, 30 28911 Legans, Madrid, Spain*
[c] *School of Computing and Mathematics, Charles Sturt University, Albury, NSW, Australia*

Abstract

Content discovery is a critical issue in unstructured Peer-to-Peer (P2P) networks as nodes maintain only local network information. However, similarly without global information about human networks, one still can find specific persons via his/her friends by using social information. Therefore, in this paper, we investigate the problem of how social information (i.e., friends and background information) could benefit content discovery in P2P networks. We collect social information of 384,494 user profiles from Facebook, and build a social P2P network model based on the empirical analysis. In this model, we enrich nodes in P2P networks with social information and link nodes via their friendships. Each node extracts two types of social features - *Knowledge* and *Similarity* - and assigns more weight to the friends that have higher similarity and more knowledge. Furthermore, we present a novel content discovery algorithm which can explore the latent relationships among a node's friends. A node computes stable scores for all its friends regarding their weight and the latent relationships. It then selects the top friends with higher scores to query content. Extensive experiments validate performance of the proposed mechanism. In particular, for personal interests searching, the proposed mechanism can achieve 100% of Search Success Rate by selecting the top 20 friends within two-hop. It also achieves 6.5 Hits on average, which improves $8x$ the performance of the compared methods.

Keywords: Social P2P, Content Discovery, Similarity, Knowledge

*Corresponding author. Tel.:+33 01 60 76 41 65
Email addresses: han.xiao@telecom-sudparis.eu (Xiao Han), acrumin@it.uc3m.es (Ángel

1. Introduction

Unlike the traditional client/server model, each node[1] in Peer-to-Peer (P2P) networks acts both as a server and a client. Thus, the node is allowed to share resources (e.g., files, peripherals) directly with others, which makes P2P networks quite popular. A report from Palo Alto Network [40] shows that P2P file sharing consumes 14% of overall bandwidth between November 2011 and May 2012, surpassing other applications. Furthermore, with the increasing demand for multimedia entertainment, P2P networks are being broadly used in video streaming applications, such as PPstream, PPLive and UUSee.

In P2P networks, content discovery is a critical problem. There are two typical classes of its solutions: structured and unstructured. Structured P2P, using Distributed Hash Table (DHT) [24] [53] [37], is efficient but inflexible under a dynamic environment. Compared to unstructured P2P, it also produces more overheads for finding popular content. Unstructured P2P is widely used over the Internet [36]. Gnutella [18] is the first practical implementation of unstructured P2P. However, it applies flooding to search content and cannot adapt to the complex networks. Although many improved approaches [16] [62] [63] [35] have been proposed, content discovery still remains a challenge in unstructured P2P, especially for unpopular content which is stored by only a few nodes. This is due to the lack of global network topologies and content information.

Nevertheless, similarly without global information of complex human networks, humans can efficiently find out specific people by exploiting their own **Social Information** (i.e., *friends*, and friends' *background information* such as nationality, interests and city). On one hand, researchers tend to verify this through experiments. In 1950s, from real human networks, Milgram revealed that any randomly selected people can reach the others by about six people on average [38]. It has also been demonstrated that users on Facebook can reach others through 3.74 intermediaries [6]. On the other hand, researchers are also inspired to extract the underlying characteristics of people behavior (e.g., people communicate more

Cuevas), noel.crespi@telecom-sudparis.eu (Noel Crespi), rcuevas@it.uc3m.es (Rubén Cuevas), xhuang@csu.edu.au (Xiaodi Huang)

[1] nodes & users are exchangeable in this paper

with each other when they have more similarity [30]), and leverage them to enhance performance in diverse systems, such as prediction systems [7], recommendation systems [41], and advertisement systems [8].

In this paper, we are motivated to investigate how social information could benefit content discovery in unstructured P2P networks. In particular, by learning from humans' experience on finding people, we propose to exploit social information from real social networks and look for content via a subset of friends that are selected based on their social information. Our approach is different from the existing work. First, we do not infer nodes' preferences and social relationships by monitoring their behavior as suggested in [52] [33], since such information is explicitly exposed among friends on social networks. Either, we do not group nodes into communities by exploiting complex algorithms presented in [15] [46]; instead, we use the user-generated friendships which are straight-forward and reliable. In addition, we especially look into content discovery regarding users' personal interests (i.e., users' own interests which include both popular and unpopular content) rather than only focus on the popular ones.

1.1. Challenges

It is a non-trivial task of leveraging social information to improve content discovery in P2P networks. We encounter the following challenges:

First, to leverage social information into P2P network and verify the newly proposed social P2P mechanism, real social information data are required. Although the recent online social networks reflecting human networks provide plenty of users' social information, it is not easy to collect such social information.

Second, since the existing P2P platforms do not involve or exploit social information, how to associate the nodes in P2P networks with social information is another challenge.

Third, even if we are able to solve the second challenge and enrich nodes in P2P networks with their associated social information, it is still hard to properly exploit such information and achieve good performances (e.g., high success rate and low cost) for content discovery.

1.2. Method and Contributions

To solve the challenges, we first capture a large volume of social information from Facebook. The studies on these data reveal that: (1) a node shares higher similarity with its friends than with randomly selected nodes; (2) a node's friends present different degrees of *Similarity* to itself and report different amount of *Knowledge* (e.g., friends, interests). Intuitively, a node is more likely to find content from those nodes that present higher similarity and more knowledge. Therefore, we then build up a social P2P Network Model that connects nodes with their friends rather than randomly selected nodes. On top of this model, we propose a Top K *social-DRWR-P2P* Search Algorithm, which selects a subset of friends with higher similarity and more knowledge. The details are as follows:

Social P2P Network Model: The model projects users' social information in social networks into corresponding nodes of users in a P2P network, and links nodes according to users' friendships. In the model, a node estimates the weight of a link, which is defined as a friend's content discovery weight, by applying two types of social features: the friend's *Knowledge*; and the *Similarity* between the node and its friend.

Top K *social-DRWR-P2P* Search Algorithm: Based on the social P2P model, the algorithm extracts the latent friendships among a node's friends and computes scores for its friends according to their content discovery weights by using a modified Distributed Random Walks with Restart (DRWR) method. Eventually, by using the algorithm, a node ranks its friends based on the scores and forwards queries to its top K friends (receivers) on the ranking list.

The proposed method (i.e., *social-DRWR-P2P*[2]) is evaluated on Facebook data. It achieves a higher success rate and lower cost than *social-P2P*[3] and *traditional-P2P*[4]. Especially, *social-DRWR-P2P* could reach 100% of Search Success Rate (SSR) by selecting top 20 friends within two-hop for personal interests searching. Under the same condition, the compared methods achieve 90.5% and 61.4% of SSR respectively. In addition, *social-DRWR-P2P*

[2] *social-DRWR-P2P* selects receivers by the proposed algorithm over the social P2P network model

[3] *social-P2P* selects receivers randomly among the sender's friends over the social P2P network model

[4] In *traditional-P2P*, receivers are randomly selected among all the other nodes

achieves 6.5 Hits on average, which is more than 8 times superior to the compared methods.

We conclude the contributions in this paper:

(1). We collect social information of 384, 494 user profiles from Facebook. We also carry out extensive studies on these data and extract useful characteristics which inspire the design of the content discovery mechanism.

(2). We propose a social P2P network model and associate the nodes in P2P networks with social information reasonably. The model exhibits two advantages for content discovery: first, the model links nodes with their friends who can discover users' interests with higher probabilities compared to the randomly selected nodes; second, the node in this model estimates its friends' content discovery weights by integrating social features of *Knowledge* and *Similarity*.

(3). Based on the social P2P network model, we extract latent friendships among a node's friends and further propose a Top K *social-DRWR-P2P* algorithm to select a subset of optimal friends. In addition, we exploit a parameter optimization approach to adjusting social feature parameters in the algorithm. The extensive evaluations reveal the efficiency of the proposed method, especially for users' personal interests search.

(4). We discuss reasonability of the social P2P network model in Section 3.1.2 and discuss practicality of the proposed mechanism in Section 7. We give suggestions about how to apply the proposed mechanism to unstructured P2P applications.

The rest of this paper is organized as follows. Section 2 describes and analyzes Facebook dataset. We discuss the proposed mechanism in Section 3. In Section 4 we elaborate experimental methodology and parameters setup. We evaluate the proposed mechanism in Section 5. Section 6 introduces some related work. Section 7 discusses the practicality of the proposed mechanism and Section 8 concludes the paper.

2. Data Description and Analysis

Facebook is one of the most popular social networks and attracts a great deal of attention from all over the world including celebrities, merchants, and politicians. In this section, we

introduce the data captured from Facebook, and report the analytical results based on these data.

2.1. Data Description

We crawl Facebook by two methods - Breadth First Search (BFS) [17] and random methods. Using BFS, we construct a *Friends Group* dataset; and set up a *Random Group* dataset by a random method. To construct the *Friends Group*, we randomly select some users as roots and then collect social information from roots, roots' friends (one-hop) and the friends of roots' friends (two-hop) subsequently. In addition, we choose a group of Facebook users at random(without structure) and capture their profiles to build up the *Random Group*.

We classify the collected social information into three categories: basic profile, social relationship and user interest. Basic profile is comprised of the attributes of a user's *ID, Name, Age, Gender, Hometown, Education*, and *Profession*. Social relationship indicates the *Friend List* of a user; while we refer the users who are not on the friend list as strangers to the user. Additionally, user interest contains 5 particular attributes- *Music, Movie, Book, Game* and *Television* - which are present on users' 'Favorite' pages in Facebook. These five particular attributes are merged as Interest in the following data analysis and experiments.

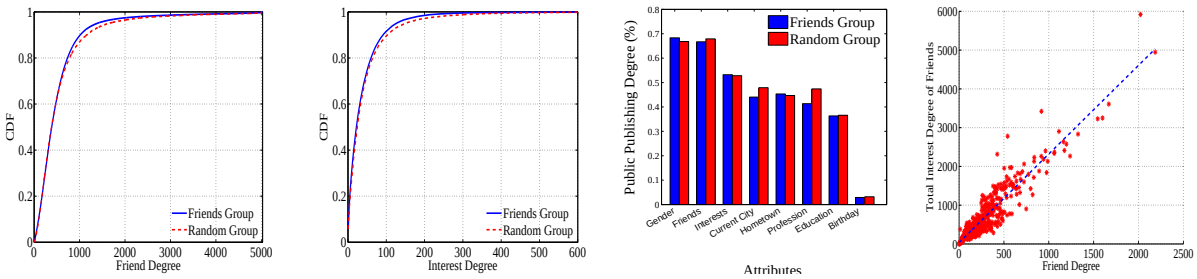
Note that users on Facebook are not requested to complete all social attributes in their profiles. Facebook also allows users to configure privacy according to each single attribtue. Thus we call users who present any of their social information to strangers as Public Users, and those who totally hide their information from strangers as Private Users. We merely extract Public Users' public social information as our dataset resources.

We collected $363,534$ users for *Friends Group* and $20,960$ users for *Random Group* from March to June in 2012.

2.2. Dataset Analysis

In this section, we study and compare users' characteristics in both groups. First, we conduct a series of preliminary statistics studies. We also compare the users' similarity

between two groups. Finally, we study the distributions of interests' popularity in both Groups.



(a) CDF of Friend Degree (b) CDF of Interest Degree (c) Attribute Public Degree (d) Total Interest Degree of Friends by Friend Degree

Figure 1: Preliminary Statistics Studies.

2.2.1. Preliminary Statistics Studies

First we review several statistics of public social information drawn from aggregations of users. We have two goals here: (1) reveal the representativeness of the datasets; (2) explore social features which might be useful for content discovery.

On one hand, the representativeness of the information in the two groups guarantees the reliability of the following data studies, comparisons and data-based experiments. BFS is not an absolutely unbiased sampling method. However, as we select the roots at random, we believe that the information in *Friend Group* is representative. Besides, this study demonstrates that the bias caused by the different amount of samples in two groups is negligible.

On the other hand, since we focus on the design of content discovery mechanism in this work, the inspection regarding the useful social information is prerequisite. In particular, concerning an interest catalogue with M interests in total and a user associating with m interests, the possibility of discovering any interest in the catalogue from the user equals m/M. It is an increasing function of m, which implies that the users who associate with more interests can provide larger probability to discover any interests for others. Therefore, we expect to reveal the users who associate with more interests.

Method(M)1: We compare Friend Degree, Interest Degree and Attribute Public Degree between *Friends Group* and *Random Group*. We also look into the relation between a user's Friend Degree and the total Interest Degree of all his/her friends.

M1.1: A user's Friend Degree is defined as the number of his/her friends. We plot Cumulative Distribution Function (CDF) of users' Friend Degree for both *Friends Group* (the blue solid line) and *Random Group* (the red dotted line) in figure 1(a). Note that the data shown in the figure has been excluded the users who have no friends.

M1.2: Similarly, a user's Interest Degree is defined as the total number of his/her interest (the sum of the number of *Music, Movie, Book, Game* and *Television*). Figure 1(b) draws CDF of users' Interest Degree. Similar to figure 1(a), the blue solid line stands for *Friends Group* and the red dotted line for *Random Group*.

M1.3: If a user has one public attribute, we consider this user as a Public User regarding this attribute. For instance, if a user U has two public attributes, named *Age* and *Gender*, we call U is a Public User both regarding *Age* and *Gender*. Accordingly, we define Attribute Public Degree of one attribute in a group as the number of Public User regarding the attribute divided by the total number of users in the group. We use Attribute Public Degree to further reveal the representativeness of the data in the two groups. We compare eight attributes: *Gender, Friends, Interest, Current City, Hometown, Profession, Education* and *Birthday*.

M1.4: Given a user with Friend Degree of n, we compute its Total Interest Degree of Friends by the overall number of interests that all his/her friends present. We plot users' total Interest Degree of Friends by their Friend Degree in figure 1(d).

Observation(O)1: The CDF of Friend Degree of the two groups match well with each other in figure 1(a); and so do the CDF of Interest Degree in figure 1(b). Figure 1(c) shows that the Attribute Public Degrees of the eight attributes in *Friends Group* are all very similar to those in *Random Group*. And figure 1(d) reveals that the Friend Degree strongly correlates to the total Interest Degree of Friends. Some specific observations in each subfigure are as follows:

O1.1: Most of users maintain a number of friends. 95.5% and 96.5% of users have a

Friend Degree higher than 50 in *Friend Group* and *Random Group*, respectively. And the Friend Degree of around 1% of users even exceeds 4000 in both groups. The median Friend Degree is 387 in *Friends Group* and 384 in *Random Group*, which are very similar.

O1.2: The users in *Random Group* show slightly higher Interest Degree than do the users in *Friends Group*. The median Interest Degrees is around 24 and 22 respectively in the two groups.

O1.3: Figure 1(c) shows that the largest difference of Attribute Public Degree between the two groups is approximately 6%. The average Attribute Public Degree difference between the two groups is only about 1.1%.

O1.4: It is observed that the Total Interest Degree of Friends goes up with the increasing of Friend Degree. This indicates that a user can associate (access from his/her friends) with more interests if he/she has more friends. The correlation between the total Interest Degree of Friends and the Friend Degree can be modeled linearly.

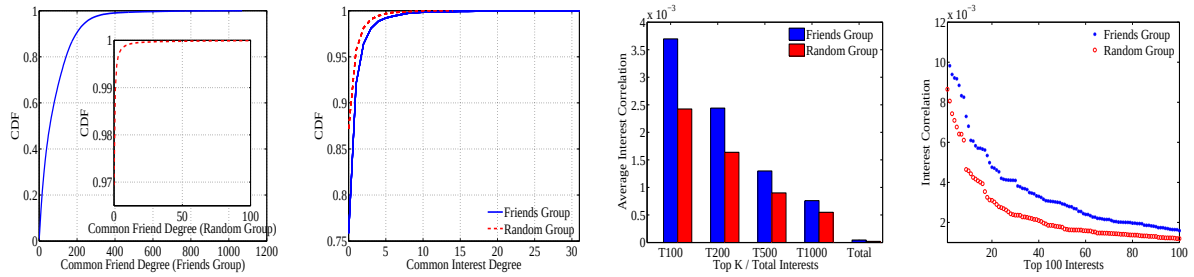
Inference(I)1: We obtain two inferences from the above comparisons:

I1.1: We assume that the social information in the two groups are representative if the statistical characteristics in the *Friends Group* approach to the corresponding ones in the *Random Group*. Therefore, we speculate that the social information in the two groups are representative and applicable for the following studies, comparisons and data-based experiments, grounded on the observations of **O1.1**, **O1.2** and **O1.3**.

I1.2: **O1.4** demonstrates that the users with higher Friend Degree can access more interests from their friends; while the users with higher Interest Degree have more interests by themselves according to the definition of Interest Degree. Hence, we infer that a user is more likely to find content from other users with higher Friend Degree and Interest Degree, since users associating with more interests can provide more probability to achieve content discovery.

2.2.2. Users Similarity

We suppose that if a user U shares more common interests with user A than with user B, it is easier for U to find his/her interests from A than from B. Similarly, if U shares more



(a) CDF of Common Friend Degree (b) CDF of Common Interest Degree (c) Average Interest Correlation of TopN Interests (d) Interest Correlation of Top 100 Interests

Figure 2: Users Similarity

common friends with A than with B, we assume that U has a stronger relationship with A than with B. Intuitively, the stronger relationships imply the more latent connections, common activities and common interests which might be beneficial to content discovery. In this section, we conduct studies on users' similarity in this subsection. We expect that a user present more similarity with his/her friends than with strangers.

M2: We learn similarity between two users by Common Friend Degree and Common Interest Degree. We further define Interest Correlation to compare interest similarity inside the two groups.

M2.1: We calculate the Common Friend/Interest Degree in *Friends Group* by counting the number of common friends/interests between users and their friends. For *Random Group*, we select two users at random and compute the Common Friend/Interest Degree by counting the number of common friends/interests between them. The more common friends/interests two users share, the higher similarity they have. Figure 2(a) shows the CDF of Common Friend Degree. The inside figure plots the CDF of the Common Friend/Interest Degree between strangers in *Random Group* and the outside figure shows the CDF of the Common Friend/Interest Degree between friends in *Friends Group*. Figure 2(b) presents Common Interest Degrees of the two groups.

M2.2: If a user claims a certain interest as one of his/her own interests, we call the user as a *fan* of this interest. The Interest Correlation of a certain interest is defined as the fraction of the fan number of the interest to the total fan number of all the interests in a

corresponding group as given below:

$$IC_{I_j} = \frac{\sum U_{I_j}}{\sum_{I_i \in I} \sum U_{I_i}}$$

where $\sum U_{I_j}$ is the fan number of interest I_j and I is the total number of interests in the group.

We rank all the interests in *Friend Group* and *Random Group* respectively by their Interest Correlation, and compute the Average Interest Correlation of the top K interests ($K = 100, 200, 500, 1000$, and the total number of interests) to compare the entire Interest Correlations inside the two groups. The larger the entire Interest Correlation within a group obtains, the higher is the interest similarity among users inside the group. In addition, we compare the individual interest correlation of the top 100 interests in the two groups.

O2: The investigations for similarity between users show that a user has higher Common Friend/Interest Degree with his/her friends than with strangers. We also note that different friends of a user share different Common Friend/Interest Degree with the user. In addition, it is observed that the Interest Correlations are higher among friends in *Friends Group* than those among strangers in *Random Group*. In particular, we observe:

O2.1: In figure 2(a), more than 99% of the randomly selected pairs of users have no common friends in the *Random Group*. In contrast, more than half of the friend pairs share 100 common friends in *Friends Group*. Although the common interests between two users are very sparse, the maximum Common Interest Degree of *Friends Group* reaches 31 which doubles that of 14 in *Random Group*. The average Common Interest Degree of *Friends Group* and *Random Group* are 0.42 and 0.21 respectively.

O2.2: The average interest correlation of the top K interests (shown in figure 2(c)) and individual interest correlation of the top 100 interests (shown in figure 2(d)) both are higher in *Friends Group* than in *Random Group*.

I2: We suppose that a user is more likely to find content for another user if they have higher similarity. Therefore, we obtain the following two inferences.

I2.1: As the Common Friend/Interest Degree and the Interest Correlations are higher in *Friends Group* than in *Random Group*, friends present a higher similarity than strangers.

Hence, we infer that it might be easier to discover content for a user via his/her friends than through strangers.

I2.2: We also conjecture that a user might be more likely to find a content from the friends with higher Common Friend/Interest Degree.

2.2.3. Interest Popularity Distribution

An interest is considered as a popular interest if many users state it as an interest on Facebook. In this section, we test how many percentages of interests are popular to most of the users in the two groups. For each user, we also study the percentage of unpopular interests that he/she presents. This study would reveal how important it is to take into account content discovery regarding users' personal interests (both popular and unpopular ones).

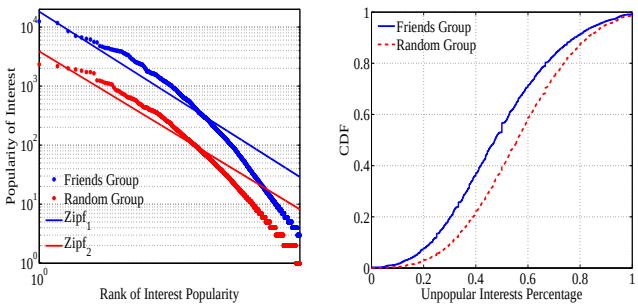
M3: we look into Interest Popularity Distribution and the percentage of unpopular interests among each users' personal interests. Interest Popularity Distribution is computed to estimate how popular the interests are. We also look into the Percentage of Users' Unpopular Interests.

M3.1: We define the popularity of an interest as the number of its fans. The interest is more popular if it attracts more fans. We rank all the interests based on their popularity. Figure 3(a) shows the interest popularity distribution in the log-log scale.

M3.2: We assume the top 500 interests are popular interests and the rests are unpopular ones. Figure 3(b) displays the CDF by the percentage of users' unpopular interests.

O3: We observe that the interest popularity distribution is very skewed - most of the interests are unpopular which only attracts quite few users. In addition, we also observe that almost 50% of a user's interests are not popular in both groups. Some details are reported as follows:

O3.1: Figure 3(a) shows that the Interest Popularity Distribution of both groups shapes in Zipf lines. In the *Friends Group*, only around 23.4% of users prefer the top One interest and the 500th interest attracts only 0.35% of users. While in the *Random Group*, the top One and the 500th interests are preferred only by 13.2% and 0.4% of users respectively.



(a) Interests Popularity Distribution

(b) CDF of percentage of users' unpopular interests

Figure 3: Interests Distribution

Generally speaking, most of interests are preferred by only a small number of users.

O3.2: Figure 3(b) reveals that, for more than 45% of users in the *Friends Group*, half of their interests are unpopular; while for nearly 75% of users when it comes to the *Random Group*.

I3: From the perspective of the interests in a group, most of interests are not popular; whereas from the perspective of users, unpopular interests account for around half of their interests on average. Therefore, we state that only improving discovery of popular content cannot satisfy users' requirements. We have to take into account users' unpopular interests meanwhile.

2.2.4. Analysis Summary

We briefly summarize the main inferences which might guide the design of content discovery mechanism as follows:

Summary(S)1: Concerning about content discovery for users' personal interests is very important for satisfying users' P2P experiences (see **I3**);

S2: A user discovers his/her personal interests more easily from his/her friends than from strangers (see **I2.1**);

S3: A user is more likely to find content from his/her friends with more friends and interests (see **I1.2**);

S4: The friends who share more common friends/interests would achieve content discovery with higher possibilities (see **I2.2**).

3. Social-Based Content Discovery Mechanism

The content discovery problem is normally approached by finding paths from a starting node to target nodes that store the queried content in a network. Our idea is to cast this problem as a task that a sender (starting node or any mediator node) ranks all candidate nodes and selects top-ranked ones as the next hop (i.e., receivers) on the paths. We aim to assign higher scores to the nodes that more likely reply to the sender's query.

Grounded on both the analytical results from the previous section and the idea of selecting receivers, we attempt to achieve content discovery with high performance for users' personal interests (see **S1**). First, we build up a social P2P network model which leads to the content discovery for a user via his/her friends (see **S2**). In this model, the nodes connect to their friends by using social relationships in social networks and weight their friends based on two types social attributes - **Knowledge** (refer to **S3**) and **Similarity** (refer to **S4**). On top of this model, we introduce a Top K *social-DRWR-P2P* search algorithm to select receivers for each sender. This algorithm chooses a user's friends that have more knowledge and share higher similarity with this user. The next two subsections explain the social P2P network model and search algorithm in details.

3.1. Social P2P Network Model

In order to construct the social P2P network model, shown in figure 4, we project users' social information on social networks into the corresponding nodes in a P2P network. The nodes thus inherit the users' basic profiles, friends' lists, and interests' lists. The nodes connect to each other if they are friends on social networks. Therefore, we define the social P2P network model as a weighted directed graph $G = \{V, S, E\}$, where V is the set of nodes in the network model; S is the set of nodes' social information inherited from social networks; and $E \subseteq V \times V$ is the set of weighted links which are determined by users' friendships. In this graph, each node estimates the weights of its links with respect to the corresponding

friends' probabilities of discovering content, namely friends' content discovery weights. In the following sections, we discuss the calculation of friends' content discovery weights and feasibility of the social P2P network model.

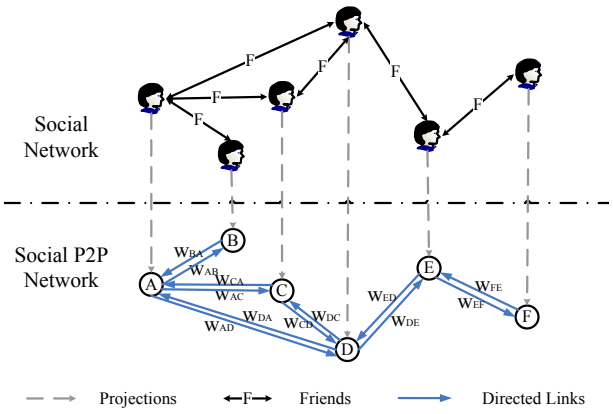


Figure 4: Social P2P Network Model

3.1.1. Friend's Content Discovery Weight

Referring to **S3** and **S4** in Section 2.2.4, we estimate friends' content discovery weight by two ways accordingly. Thus we obtain two types of social features, namely **Knowledge features** and **Similarity features**, which are detailed as follows.

Knowledge features: we define a node's knowledge features by the amount of resources (i.e., knowledge degree) with respect to various users' social attributes. In particular, each social attribute is associated with one knowledge feature. For example, regarding the social attribute of a user's friend (interest) list, we compute friend (interest) degree by counting the number of friends (interests). We expect that the friends with more knowledge would be more likely to reply the node's content query (refer to **S3**). Therefore, we assign higher weights to the friends with more knowledge.

Knowledge weight matrix: to explain how to weight friends by their knowledge, we consider a node i and its r friends. Specifically, assuming n types of knowledge features are employed, for one of its friend j, the node i denotes all the quantified knowledge degrees as $D_{ij}^{(K)} = (d_{ij}^{k1}, d_{ij}^{k2}, ..., d_{ij}^{kn})$. Similarly, for all of its friends, the node i generates a knowledge degree matrix $(D_i^{(K)})$. $D_i^{(K)}$ is a $r \times n$ matrix, in which each row stands for the knowledge

degrees of one friend over n knowledge features; and each column represents the knowledge degrees on one particular knowledge feature by different friends. Using the logistic way, we normalized the xth knowledge degree of friend j by: $norm_x(d_{ij}^{kx}) = \frac{1-exp(-d_{ij}^{kx}/\theta^x)}{1+exp(-d_{ij}^{kx}/\theta^x)}$, where θ^x is a regularization parameter by the xth knowledge degree. Eventually, the node i calculates knowledge weights for all its friends by normalizing the matrix of knowledge degree ($D_i^{(K)}$), denoted as:

$$W_i^{(K)} = norm(D^{(k)}) = \begin{bmatrix} norm_1(d_{i1}^{(k1)}) & norm_2(d_{i1}^{(k2)}) & \dots & norm_n(d_{i1}^{(kn)}) \\ norm_1(d_{i2}^{(k1)}) & norm_2(d_{i2}^{(k2)}) & \dots & norm_n(d_{i2}^{(kn)}) \\ \vdots & \vdots & \ddots & \vdots \\ norm_1(d_{ir}^{(k1)}) & norm_2(d_{ir}^{(k2)}) & \dots & norm_n(d_{ir}^{(kn)}) \end{bmatrix} \quad (1)$$

Similarity features: we compute similarity of two users with respect to their social attributes as similarity features. Such features measure how much two users are similar regarding the corresponding attributes. For example, we can derive friend (interest) similarities between a user and his/her friends by employing their social attributes of friend (interest) list. We conjecture that the friends who have higher similarity with the node would be more likely to reply a satisfactory content (refer to **S4**). Hence, such friends should be assigned with larger weights too.

Similarity weight matrix: suppose that we discuss m types of similarity features, then node i's similarity features are expressed by a vector, $F_i^{(S)} = (f_i^{s1}, f_i^{s2}, ..., f_i^{sm})$. Regarding each feature, node i computes the similarity with its friend by the cosine distance. In regard of the lth feature, the similarity weight between node i and its friend j can be calculated by:

$$w_{ij}^{(sl)} = \frac{f_i^{(sl)} \cdot f_j^{(sl)}}{\|f_i^{(sl)}\| \cdot \|f_j^{(sl)}\|}$$

For friend j, node i records their similarity weights over all the m features by a similarity weight vector, i.e., $w_{ij}^{(S)} = (w_{ij}^{s1}, w_{ij}^{s2}, ..., w_{ij}^{sm})$. Similarly, node i calculates the similarity weight vectors for all of its friends (r in total) and further integrates them into a similarity

weight matrix. Thus, the similarity weight matrix generated by node i equals:

$$W_i^{(S)} = \begin{bmatrix} w_{i1}^{(S)} \\ w_{i2}^{(S)} \\ \vdots \\ w_{ir}^{(S)} \end{bmatrix} = \begin{bmatrix} w_{i1}^{(s1)} & w_{i1}^{(s2)} & \dots & w_{i1}^{(sm)} \\ w_{i2}^{(s1)} & w_{i2}^{(s2)} & \dots & w_{i2}^{(sm)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{ir}^{(s1)} & w_{ir}^{(s2)} & \dots & w_{ir}^{(sm)} \end{bmatrix} \tag{2}$$

Integration of Knowledge and Similarity: at last, node i computes the integrative weights for its friends (i.e., friends' content discovery weights, $W_i^{(KS)}$) by incorporating their knowledge weights and similarity weights as follows:

$$W_i^{(KS)} = W_i^{(K)} \cdot \boldsymbol{\alpha} + W_i^{(S)} \cdot \boldsymbol{\beta} \tag{3}$$

where $\boldsymbol{\alpha} = [\alpha^{k1}\alpha^{k2}...\alpha^{kn}]^\top$ and $\boldsymbol{\beta} = [\beta^{s1}\beta^{s2}...\beta^{sm}]^\top$, are the parameters of the model, and $\alpha^{k1} + \alpha^{k2} + ... + \alpha^{kn} + \beta^{s1} + \beta^{s2} + ... + \beta^{sm} = 1$.

As different attributes might affect content discovery performance at varying degrees, we expect to find out a set of optimal feature parameters according to the feature's influence on performance of content discovery. The parameters optimization problem is discussed in Section 3.2.3.

3.1.2. Discussions of Social P2P Network Model

It is reasonable to map users' social information from social networks onto the nodes in a P2P network. Nowadays, a huge number of Internet users apply P2P platforms to share files, and meanwhile communicate on various social networks. For example, Bob often watches movies on PPStream, while he also claims his favorite movies on Facebook. Although Bob's favorite movies are not explicitly claimed on PPStream, it is reasonable that PPStream uses these information to enhance Bob's experience. We further discuss the practicality of this model in Section 7.

In addition, there are two reasons that we set up a social P2P network model by linking nodes via friendships. First, we are inspired by the analytical result that a user is more likely to find his/her interests through friends than strangers. Second, considering the plenty of

nodes in a P2P network, it is resource-consuming and time-wasting to compute links' weights and rank them. The social P2P network model considerably scales down a sender's candidate nodes to its friends and makes it lightweight to run a ranking algorithm.

3.2. Top K social-DRWR-P2P algorithm

In this section, we propose a Top K *social-DRWR-P2P* algorithm to further select a subset of friends over the social P2P network model. First we introduce the basic algorithm of Random Walking with Restart (RWR). Then we present a modified version of RWR, namely Distributed RWR (DRWR), which could be applied distributedly in our social P2P network model. DRWR biases the friends who are more likely to reply to the queries with higher scores. In order to score friends properly, we discuss the model parameter optimization problem subsequently. We eventually present the Top K *social-DRWR-P2P* mechanism and give an example of receiver selection.

3.2.1. Random Walk with Restart

Given a weighted graph $G(V, E)$, RWR performs walks starting from a node s to other nodes by following the probabilities of the edges that are proportional to their weights at each step. We assume that each step of a random walker is independent of its previous moves, thus we could employ a Markov chain to describe the path that the random walker visited. We denote the state that a random walker is visiting node i at step t as $i = i(t)$. The transition probability of a random walker shifting from state $i = i(t)$ to the next state $j = j(t + 1)$ is:

$$p_{sj}(t) = p(j(t + 1)|s(t)) \tag{4}$$

$\boldsymbol{p}(t) = \{p_{sj}(t)\}$ is called the transition probability vector at step t for all nodes. In addition, at each step we also consider a probability, namely the self-transition probability δ, of making the random walker go back to the starter s. We calculate the shifting rate by using the following equation recursively:

$$\boldsymbol{p}(t + 1) = (1 - \delta)\boldsymbol{A}\boldsymbol{p}(t) + \delta\boldsymbol{q} \tag{5}$$

In this equation, $\boldsymbol{q}$ is a vector where the elements equal 0 except for the one that corresponds to the initial node being set to 1. $\boldsymbol{A}$ is a matrix in which the elements stand for the state transition probabilities between two nodes. If i and j are disconnected to each other, $a_{ij} = 0$; and otherwise $a_{ij} = w_{ij}/w_{(i\cdot)}$ where $w_{(i\cdot)} = \sum_{j=1}^{n} w_{ij}$. w_{ij} is the weight that node i assigns to its friend j, calculated by equation 3. Therefore, the matrix A is computed as:

$$A = W^{(K)} \boldsymbol{.\alpha} + W^{(S)} \boldsymbol{.\beta} \tag{6}$$

Since the random walker's visiting pattern is a Markov process, the transition probability vector can converge after a number of steps l. Finally we obtain $p(l)$ as a stationary measure of the shifting rate.

3.2.2. Distributed RWR (DRWR)

Each node in the social P2P network constructs a <FRIEND, WEIGHT> table (denoted as $Ti < F, W >$) by computing its friends' weights and exchanges it with their friends. Each node, from its friends' $Ti < F, W >$, picks out the entries that reflect the latent relationship among its friends. By merging the selected entries from all its friends' $Ti < F, W >$, the node builds up a mixed <FRIEND, WEIGHT> table called $MTn < F, W >$ and calculates transition matrix $\boldsymbol{A}$. Finally, the node conducts a local random walk over all its friends and computes a stable transition probability for each friend as its score, by using the Eq.(5). DRWR method could extract the latent friendship behind a node to bias its friends' scores.

To further explain the DRWR method, we illustrate how node 1 in figure 5 assigns scores to its friends as an instance. Node 1 has three friends of nodes 2, 3 and 4 where node 3 connects to nodes 2 and 4 as well. We depict the links between two nodes in solid lines if both of them are either node 1 or its friends; while use dashed lines to represent the other links. The numbers on the links represent the friend content discovery weight. After exchanging $Ti < F, W >$, node 1 filters the weights of node 5 from $T2 < F, W >$ and $T3 < F, W >$. It also removes the weight of node 6 from $T4 < F, W >$. Node 1 obtains $MT1 < F, W >$ by means of filtering and merging all collected $Ti < F, W >$, as shown in the middle of figure 5. At the right side of this figure, we depict the initial transition matrix $\boldsymbol{A}$ on node 1. Node

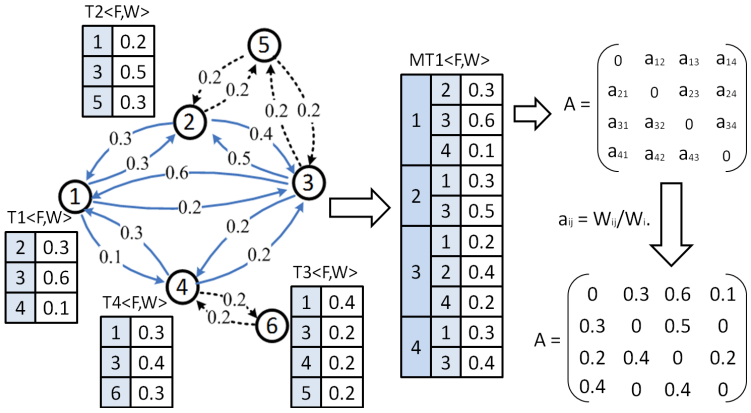


Figure 5: Distributed Random Walk

1 computes the scores by solving Eq.5.

3.2.3. Parameters Optimization

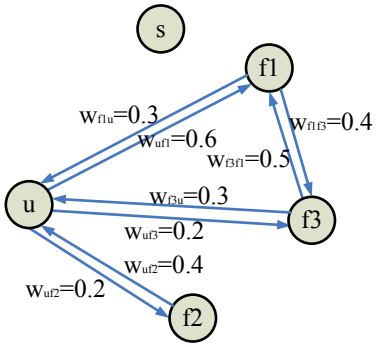
As we mentioned in Section 3.1, different information attributes affect content discovery performance at varying degrees. Hence we expect to find out a set of optimal feature parameters for the calculation of nodes' weight and finally to assign proper scores to friends by using DRWR. To address the problem, we begin with a sender s and divide all its friends into two subsets, denoted as F_k and F_r. We expect that the subset of F_k is comprised of the friends from which the sender could find the queried content with higher probabilities; while F_r consists of the friends of lower probabilities for content discovery. Therefore, we aim to find out an optimal parameter set for features that give the friends in F_k greater scores than those in F_r. We denote the parameter vector as $\boldsymbol{a}$ and define the optimization problem as:

$$\min_{\boldsymbol{a}} F(\boldsymbol{a}) = \| \boldsymbol{a} \|^2 + \lambda \sum_{k \in F_k, r \in F_r} h(p_r - p_k) \tag{7}$$

where λ is a regularization parameter and $h(.)$ generates a non-negative penalty which $h(.) = 0$ as $p_r < p_k$ while $h(.) > 0$ as $p_r > p_k$. To obtain the optimal parameters set, we exploit the gradient based optimization approach to minimizing the loss value [7] (Appendix A offers more details about parameter optimization.)

3.2.4. Top K social-DRWR-P2P Search Algorithm

In this section, we summarize the top K *social-DRWR-P2P* search algorithm: First, a node constructs connections based on its friendships presented by the corresponding user in social network. Then, the node leverages numerous features - namely friends' knowledge and similarity - to assign weights to its friends. By exploiting the DRWR algorithm, the node computes stable scores for its friends. Eventually, the node ranks all its friends based on their scores and selects the top K friends from the ranking list to forward queries.



R	Weights	DRWR Scores
f1	0.6	1
f3	0.2	0.669
f2	0.2	0.205
s	0	0

Figure 6: An Example of Top K *social-DRWR-P2P* Search Algorithm

An example of Top K *social-DRWR-P2P* Search is illustrated in Figure 6. First, every node connects with their friends and estimates its friends' weights based on their similarity and knowledge. In this example, user u connects to its friends $f1$, $f2$ and $f3$ and measures their weights (0.6, 0.2 and 0.2 respectively) based on friends' similarity and knowledge from u's own perspective. Similarly, $f1$, $f2$ and $f3$ also estimate weights for their friends. User u does not link to the stranger s and weights s as zero. Then, u runs DRWR algorithm to score each of its friends, shown in the right table.

Particularly, we note that the final scores assigned by DRWR are not as the same as the initial weights. For instance, from u's own perspective, $f2$ and $f3$ have the same weight. However, $f3$ should be assigned a higher score than $f2$ intuitively since $f3$ is also a friend of u's friend ($f1$). This relationship makes u, $f1$ and $f3$ much closer to each other and raises the content discovery probabilities of $f1$ and $f3$. It is the DRWR algorithm that explores the latent friendship between friends $f1$ and $f3$ to increase their scores. Finally, we select

top K friends from the friends ranking list.

Furthermore, the proposed mechanism is flexible with the changeable knowledge and similarity features. Different feature parameters are assigned according to specific applications. In addition, we notice that the complexity of the algorithm is $O(l)$ defined by the convergent steps.

Algorithm 1 Top K *social-DRWR-P2P* Search

Input: Friends' information from OSN:

The set of friends' list of a node;

The set of friends' feature;

The number of selected friends: K

1: Initialize unstructured social_P2P network;

2: **for** each $f^k \in KnowledgeFeatures(F^{(K)})$ **do**

3: Assign the weight of feature f^k to each friend of the node(Equation 1)

4: **end for**

5: **for** each $f^s \in SimilarityFeatures(F^{(S)})$ **do**

6: Assign weight of feature f^s to each friend of the node(Equation 2)

7: **end for**

8: Combine all factored features' weight (Equation 3)

Iteration: Run DRWR until probability vector $\boldsymbol{p}$ converges. $l = 0$;

9: **for** $\boldsymbol{p}$ is not convergent **do**

10: Calculate stable transition probability (i.e., score) for each friend (Equation 5)

11: $l++$;

12: **end for**

13: Order friends based on friends' scores

14: **return** Selected Top K friends of the node

4. Experiments Setup

We use the two Facebook datasets to evaluate the proposed the mechanism. The friendships are used to connect nodes in the social P2P network, and the information of a user's friends is applied to estimate the content discovery weights. In this section, we first introduce the experiment method and performance metrics. Then, we describe the parameter setup in the proposed social P2P network model.

4.1. Experiment Design

4.1.1. Assumption and Evaluation Strategies

Receivers (any mediator nodes or the target nodes) in the experiments store a set of content so as to reply to the queries from the starting node. Facebook supports user-generated interests explicitly. Here we assume the receivers store their favorite Movie, Music, Book, Game and TV series, or know how to find out their interests even if they do not store them on their disk. And then it is plausible to assume that a receiver's interest list on Facebook works as his/her content list.

From the perspective of a normal user (starting node), two kinds of interests are desirable: the user's personal interests and the most popular interests. Our evaluations are therefore composed of two parts: personal interests searching and popular interests searching. In personal interests searching, we assume that the starting node looks for all its interests from others. In popular interests searching, top 500 interests in each group are considered as its popular interests, and the starting node searches all the popular interests.

4.1.2. Comparison

We compare the newly proposed content discovery mechanisms (i.e., *social-DRWR-P2P*) to *social-P2P* and *traditional-P2P*.

social-DRWR-P2P: we first project the information of users in *Friends Group* to the nodes in P2P network one by one and generate the *social-P2P* network topology by following the social p2p network model introduced in Section 3.1. We run Top K *social-DRWR-P2P* algorithm and launch queries to the selected top K nodes over this network topology.

social-P2P: we use the same *social-P2P* network topology as *social-DRWR-P2P* mechanism does. However, the content discovery queries are forwarded to K randomly selected friends, instead of the top K friends selected by *social-DRWR-P2P*.

traditional-P2P: we map the users' information in the *Random Group* to the nodes in the P2P network and then a node selects K users at random to send queries.

We launch one-hop and two-hop searching by forwarding K queries at each sender.

4.2. Performance Metrics

In each content discovery procedure, a node (i.e., sender) sends the content discovery queries to K selected nodes (i.e., receiver[5]), and in turn H nodes (i.e., replier) among them reply. In addition, we refer the node that stores the queried content as a storer and denote the total number of storers as C. Then we define the following four metrics to evaluate our proposed method:

Hits: Hits is defined as the average number of replies during content discovery procedures (i.e. H). Intuitively, it relates to the selected number (K) of receivers: a sender might get more replies while it sends queries to more receivers.

Query Success Rate (QSR): QSR equals the fraction of the number of replies to the number of receivers (i.e., $QSR = H/K$). Although increasing the number of receivers might lead to more Hits, it costs more network resources (e.g., bandwidth). To some extent, over-query could even lead to network congestion and lower network performance. Hence, Hits alone is not enough for performance evaluation. Given two mechanisms which achieve the same Hits, the one with a higher QSR performs more efficiently.

Search Success Rate (SSR): We consider a content discovery procedure to be successful as long as the sender receives a reply at least from the receivers. SSR is a metric for estimating the success rate of procedures. We run M procedures in total and S of them are successful. Thus, we calculate SSR by dividing the number of successful procedures by the total number of procedures (i.e., $SSR = S/M$). Note that different P2P applications

[5]In *social-DRWR-P2P*, the receivers are the top K friends; in *social-P2P*, the receivers stand for the random selected friends; in *traditional P2P*, the receivers represent for the totally random selected users

have different requirements in content discovery: some of them are only interested in finding one single copy of content, while others look for as many copies as possible. Therefore, the former applications probably do not concern about QSR, since SSR is a very important metric for them. In contrast, QSR is meaningful for the latter applications.

Recall: Recall is computed as the number of repliers divided by the total number of storers (i.e., $Recall = H/C$). Recall reflects the capacity of a mechanism in terms of completely retrieving. If two mechanisms achieve the same Hits and QSR / SSR, the one that reaches higher Recall presents better performance.

4.3. Parameters Setup

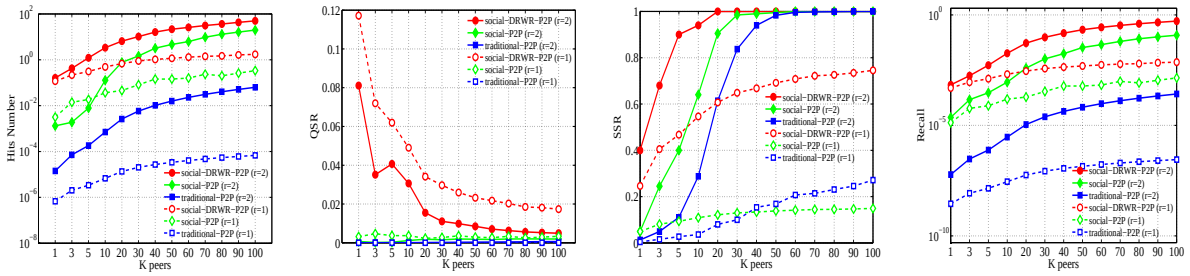
As described in Section 3, the proposed *social-DRWR-P2P* algorithm applies two ways to quantify users' social attributes, which respectively produce knowledge features and similarity features. In our experiments, we employ two social attributes which are users' friends list and interests list. Drawing on equations 1 and 2, we obtain the normalized friend degree and interest degree as knowledge features; and we compute friend similarity and interest similarity as similarity features.

5. Performance Evaluation

In this section, we compare the performance of *traditional-P2P*, *social-P2P* and *social-DRWR-P2P* with respect to personal interests searching and popular interests searching respectively. The results indicate that *social-DRWR-P2P* is superior to the other algorithms not only for discovering popular interests but also for nodes' personal interests.

5.1. Personal Interests Searching

To evaluate the discovery of users' personal interests, a starting node generates queries for all its personal interests and a receiver replies as long as it stores the queried interests. Figure 7 sequentially plots the personal interests search results of the Hits, QSR, SSR and Recall achieved by the three compared algorithms. The vertical axes are the values of the aforementioned four metrics and the horizontal axes represent the number of receivers. In



(a) Hits (b) Query Success Rate (c) Search Success Rate (d) Recall

Figure 7: Performance of Personal Interests Searching.

the figure, K only represents the number of receivers to which each sender forward queries. Therefore, the total number of receivers for two-hop search is $K + K^2$ corresponding to K at the horizontal axes in figure 7 and 8. We perform the experiments with K being $[1, 3, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$ respectively.

Figure 7(a) shows the average Hits. It is obvious that the values of Hits are getting higher as the number of receivers (K) increases. In cases of both one-hop and two-hop, *social-P2P* gains higher Hits than *traditional-P2P*. This implies that friends perform better than randomly selected nodes for personal interests searching. Compared with *social-P2P*, *social-DRWR-P2P* achieves even higher Hits. This observation indicates that friends with a higher similarity and more knowledge are more likely to find personal interests. Furthermore, in the one-hop experiments, the Hits of *social-DRWR-P2P* exceeds 1 when the receivers are more than 40; while the Hits of the other two mechanisms only reach 0.14 and 0.00003. In the two-hop estimations, *social-DRWR-P2P* can obtain 1.22 replies on average by sending queries within 5 receivers at each sender; however, *social-P2P* and *traditional-P2P* receive only 0.008 and 0.0002 replies respectively under the same condition. The results indicate that two-hop search costs fewer queries than one-hop search to achieve the same performance of Hits. For instance, to guarantee one Hits, a starting node, forwarding 5 queries at each sender in two-hop search, sends 30 queries in total; compared with 40 queries in one-hop search.

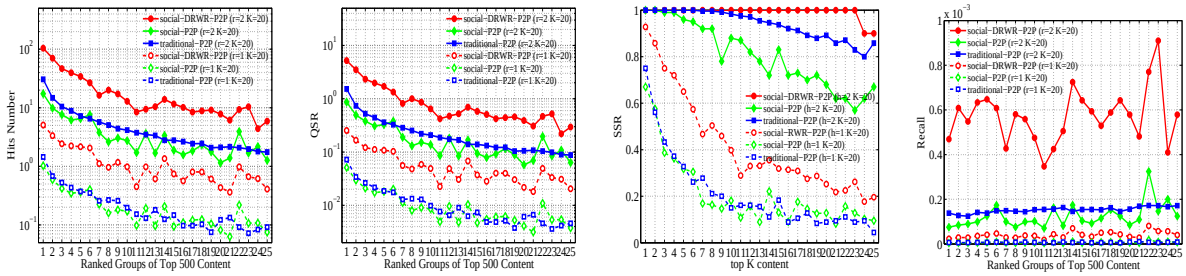
Figure 7(b) reveals that *social-DRWR-P2P* gains much higher QSR than *social-P2P* and

traditional-P2P. Additionally, we observe that, for *social-P2P* and *traditional-P2P*, the QSR changes little in a broad range of K values, especially in one-hop searching; however, the QSR of *social-DRWR-P2P* decreases obviously as K increases. In other words, the efficiency of *social-DRWR-P2P* drops while more friends with lower weight (i.e., K increases) are requested to. These observations reflect that the friends of more knowledge and similarity benefit more for content discovery. Combining the results from both Hits (figure 7(a)) and QSR (figure 7(b)), we note that when K is between 5 and 20, *social-DRWR-P2P* can obtain a good Hits (between 1.22 to 6.5)and a good QSR (between 0.041 to 0.015) within two-hop search.

From figure 7(c), we can see that the proposed mechanism also outperforms others in terms of the SSR. The SSR of *social-DRWR-P2P* achieves 100% with selecting 20 receivers at each sender in two-hop search. This means that *social-DRWR-P2P* can guarantee its success by two-hop search with sending 420 queries (*i.e.*,K=20) in total. To accomplish the same performance, *social-P2P* needs to query 3660 receivers (i.e., $K = 60$) and *traditional P2P* queries to 8190 receivers (i.e., $K = 90$) at least. In other words, in order to guarantee a successful search, *social-DRWR-P2P* saves almost 8 and 18 times queries compared to *social-P2P* and *traditional P2P*.

Figure 7(d) compares the completely retrieving capacity of the three mechanisms. In the best cases, *social-DRWR-P2P* can find out 0.73% and 52.12% of storers to reply queries in one-hop search and two-hop search respectively. Meanwhile, the *social-P2P* only locates 0.14% and 12.05% of storers, and the *traditional-P2P* explores about 3×10^{-5}% and 0.26% comparatively. On average, *social-DRWR-P2P* improves the percentage of retrieved storers by nearly 11 times in one-hop and 19 times in two-hop compared to *social-P2P*.

To summarize content discovery for personal interests, we suggest applying two-hop *social-DRWR-P2P* with selecting top 20 receivers at each sender. In this case, *social-DRWR-P2P* could guarantee a 100% successful content discovery. Also it achieves suitable Hits (6.5) with acceptable QSR (0.015).



(a) Hits (b) Query Success Rate (c) Search Success Rate (d) Recall

Figure 8: Performance of Popular Interests Searching. We group the top content by size of 20, thus the buckets are:[T1-T20],[T21-T40],...,[T481-T500].

5.2. Popular Interests Searching

In order to validate the performance of the three algorithms in terms of popular interests searching, we first rank all the interests by their popularity. Then we group the successive 20 interests from high to low in the ranking list into a bucket and calculate the average value for all the metrics. We consider the top 500 interests as the popular interests and generate 25 buckets. Figure 8 shows the one-hop and two-hop evaluations with $K = 20$ for *social-DRWR-P2P*, *social-P2P* and *traditional-P2P*.

We can see that, in the case of popular interest discovery, *social-DRWR-P2P* also outperforms *social-P2P* and *traditional-P2P* and achieves better performance of Hits, QSR, SSR and Recall under the same conditions. We account for friends' knowledge amount as a factor when we rank friends in *social-DRWR-P2P*. Therefore, the observations may respond to the fact that the selected receivers with higher scores can provide more content which also contain many popular content. However, *social-P2P* does not perform better than *traditional-P2P* method for popular interests searching, which implies that the algorithm merely involving the friendship does not benefit popular interests searching obviously.

We also observe that, in general, the content discovery queries for the interests in the higher position in a ranking list receive replies with higher probabilities as well as higher efficiency. This might suggest that for popular interests searching we could downsize the number of receivers to some extent in order to obtain enough Hits and reduce the cost

at the same time; and contrarily, we would have to send more queries to achieve similar performance for searching unpopular interests. In addition, we notice that the value of recall does not decrease as the ranking goes down. That is to say, the capacity for retrieving interests can maintain a certain level no matter how popular the interests are.

5.3. Result Discussions

The results obtained in this section provide a number of interesting insights that we summarize as follows:

(1) Due to the large number of available resources for the popular interests, retrieving such interests is a relatively easier task. Additionally users present many unpopular interests in general (see Section 2). Therefore, a good content discovery solution should be characterized by its twofold abilities of finding popular interests and personal interests. The experiments reveal that our proposed *social-DRWR-P2P* significantly improves the performance of content discovery not only for popular interests but also for personal interests.

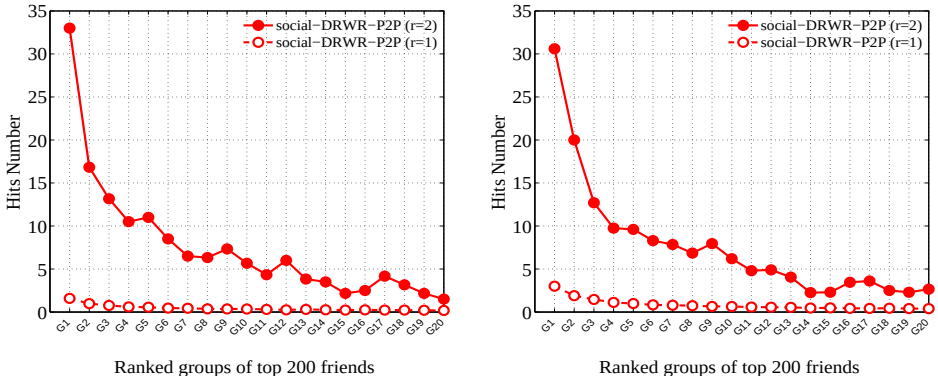
(2) We also notice that, for popular interests searching, *social-P2P* which merely considers the friendship among nodes does not show any advantage over *traditional-P2P*. This just indicates that the two aspects of our proposed mechanism - the social P2P network model and the Top K *social-DRWR-P2P* Search Algorithm - are both necessary in order to improve content discovery.

(3) It has been demonstrated that, for a certain number of queries, the proposed *social-DRWR-P2P* might perform better within two-hop search than one-hop search. For instance, if we query 110 friends within one-hop, the sender selects receivers including the relatively low ranking ones among all its friends. However, if the same amount of queries are issued within two-hop, the queries are sent to the 10 highest ranked friends and sequentially forwarded to 10 highest ranked friends of them.

(4) We can state that the friends with a higher similarity and more knowledge are more likely to reply the content from two perspectives: (i) *social-DRWR-P2P*, which selects the receivers with friends of higher weight, performs better than *social-P2P* (i.e., randomly select

friends); (*ii*) the QSR of *social-DRWR-P2P* decreases with involving more friends of lower weights (see figure 7(b)). Furthermore, we devise an experiment to verify this statement:

> With the ranked friends list generated by *social-DRWR-P2P*, we cluster each 10 successive friends from top to bottom into a group and compare the average Hits of each group. That is to say, the top 10 friends are clustered into group 1, and the next successive 10 friends (i.e., top 11th to top 20th) into group 2. In this way, we generate 20 ordered groups by the top 200 friends. Note that the friends in nth group have higher scores than friends in $n+1$th group. Figure 9 shows that, both for personal interests searching and popular interests searching, the friends with more knowledge and higher similarity can achieve better performance.



(a) Personal Interests Searching (b) Popular Interests Searching

Figure 9: Hits Number of *social-DRWR-P2P*.

6. Related Work

A lot of work has been devoted to address the problem of content discovery in P2P networks [16] [62] [35], since the very beginning of success in Gnutella [18]. As the explosion of OSN in recent years, much recent work tends to improve content discovery by involving social features or methods. Here, we briefly review social-related content discovery methods in P2P networks.

6.1. Social-Related P2P Networks

Social networks contain many inherited features which can be applied to enhance P2P systems. [47] leverages the implicit trust in social networks to address the churn problem in P2P systems. In [60], the authors reduce startup delays in P2P video sharing networks through a pre-fetching approach based on users' preferences. We try to improve content discovery techniques in unstructured P2P by exploiting social information. [31] maps Facebook users' information into P2P networks, which is related to our approach mostly. However, there are two main differences: in [31], users are clustered by their common interests, which would consume extra resources; and they organize nodes into a structured graph and perform searching by DHT. Structured P2P fails to implement keyword-matching, and it is not flexible under dynamic circumstances. There are numerous other related works which we classify them into three categories.

A classical category of social P2P searching approach forms social-like relations by learning strategies. [52] tries to infer relationships based on the historical behaviors, and identifies the powerful principle of interest-based locality: it is more likely to find content on a particular node if it occurred on the node in the past. Exploiting neighborhoods' historical queries, [33] sets up a social P2P network; the social P2P is further enhanced by introducing an active query mechanism - nodes are allowed to actively request interests from the new acquaintances [34]. In [63], the authors also look into users' friends circles and exploit the link prediction method to extract nodes' proximities, consequently enhancing the capacity for resource discovery in P2P circumstances. These approaches improve P2P searching based only on behavior observation. Yet other mechanisms build up social connections grounded on learning models which take into account nodes' attributes, preferences and any other possible social elements. In [15], the authors create an efficient overlay for a P2P file sharing system by learning users' preferences and the musical styles of users' libraries, and then connecting users who prefer the same styles of music. The authors also use real social data from a campus network. However, to identify and classify music styles, they should be aware of all the files on the network. This approach is not appropriate for a large network. Considered from the nodes' angle, it is difficult and even impossible to master the full image of the

network. Using an existing co-authorship graph, [10] generates a large P2P collaboration network, investigating diverse search mechanisms and indicating its quality.

Another group of methods applies various strategies to cluster nodes, namely community-based solutions. [32] introduces a small world architecture for P2P networks and proposes a semi-structured algorithm to achieve content discovery in multi-group P2P systems. [15] improves P2P performance by means of clustering users and creating a social network akin to the one based on users' music preference, with the Hierarchical Dirichleet Processes. In [46], the authors present the self-organized interest-based clusters in affinity networks which are further exploited to devise a proactive P2P recommendation system. [13] proposes an approach to grouping similar nodes and producing a super-peer for constructing Semantic Overlay Networks (SONs). It can achieve high-quality searching by posing similar queries to the N most-similar SONs. Generally, peers in the same community share more attributes and content, and consequently, organizing a community is a way to accelerate the search process. Meanwhile, detecting and establishing a useful community is not an easy task.

An approach that is distinct from the above-outlined classes uses the user-generated social relationship directly to improve system performance. [59] accelerates the performance of BT file sharing with the Twitter social network. The authors find that the nodes in Twitter communities are likely to meet each other again, which just restates the suggestion that long-term relationships among peers can achieve better sharing performance. Our proposal directly maps OSNs friendship graphs into an unstructured P2P network. However, in contrast to taking advantage of the inherit stable connections among acquaintances; we tend to make use of the attributes related to friends' knowledge and similarities.

6.2. Applications of Random Walks

Many other research areas have used Random Walks (RW) algorithms for different purposes. The authors in [7] propose a novel algorithm based on supervised random walks to predict the interactions that are likely to occur between users. In [27], the authors estimate an RW model of a dataset from the last.fm and show how it can ameliorate the item recommendation systems by integrating friendship and social tagging. In [2], the authors exploit

RWs to rank nodes in a graph. In the area of information retrieval, [11] applies an RW model to a large number of search engine's logs and produces a probabilistic ranking of documents for a specific query. In this paper, we propose to run a distributed RWR algorithm in P2P networks.

7. Discussion

In this section, we further discuss and explain three practical issues of the proposed mechanism.

7.1. Feasibility of Social P2P Model

In this paper, we project user social information into a P2P network to build up the social P2P network model. This model is the basis for the proposed content discovery algorithm. Therefore the feasibility of the model determines the practicability of the proposed systems. The core issue here is whether or not it is possible to set up a real social P2P sharing platform (i.e. a P2P network with users' social information).

For enlarging the influence and user volume, many existing P2P applications recommend users to combine their P2P accounts with their social network accounts. For instance, when logging on PPstream, a user would receive a message of "Login with your Weibo (Chinese Twitter) account; Login with your Renren (Chinese Facebook) account; Login with your QQ (Chinese MSN) account". Even though we have no idea of what percentage of users would meet these requirements, we believe users would accept this recommendation if they could obtain better performance for their P2P experience. The existing unstructured P2P applications could directly follow the proposed mechanism to improve content discovery.

Existing P2P applications (e.g. PPstream, PPlive) encourage users to register on their platforms. They record users' basic profile like age, gender, education, etc., and track users' history of uploading, watching and storing. As long as allowing users to make friends and encourage them to communicate with each other on their platforms, these P2P applications could easily construct their own social P2P networks. The proposed mechanism can be integrated into exiting platforms.

As Facebook is currently the biggest OSN in the world, we leverage Facebook to extract users' social information, setup the social P2P model and conduct the experiments in this research. However, for the two above-mentioned reasons, our proposed method is practical not only for P2P platforms with user accounts associated with Facebook, but also for others with their own user social networks.

7.2. Effectiveness of Facebook Dataset

Because of Facebook's privacy policy, we only crawl the public information from the public users from Facebook. Thus, one might doubt that the results of studies on Facebook and the data-based experiments are biased by the incomplete datasets. However, as shown in figure 1(c), 65% of users in the datasets present their friends and 53% of users show their interests to the public. From the point of studies, we have a considerable number of samples. In addition, we use two ways to collect information and present their generality in Section 2.2.1. From the perspective of data-based experiments, a node could probably achieve a better performance if it has more social information .

7.3. Selection of Social Features

To estimate friends' content discovery weights with respect to social attributes (in Section 3.1.1), we provide two ways by which we obtain knowledge features and similarity features respectively. We refer knowledge features as the quantifiable resources of a node; and regard similarity features as the metrics, which measure how much users are alike with respect to diverse attributes. In our opinion, this model can be flexibly extended to contain more relative social features regarding the available social information. For instance, age similarity might be a practical similarity feature, as in general younger generation of 1990s might present different tastes in movies or music, compared with middle-age people who were born in 1970s. In addition, the proposed algorithm, summing up the products of the features' values and the biased parameters (see Equation 3 in Section 3.1.1), seeks to achieve the best performance by taking overall advantage of the considered features.

8. Conclusions and Future Work

In this paper, we present a social P2P mechanism grounded on the real social network information. By linking nodes through their social friendship, we build up a social P2P network model; we weight the friendship regarding of knowledge features and similarity features. Based on this model, we further propose a content discover algorithm which selects a subset of friends by the modified version of the RWR algorithm (i.e., DRWR). This algorithm is able to explore the latent friendships among a node's friends. Although online social networks are mainly centralized nowadays, the social information that users generate and maintain can be exploited into a P2P environment. Besides, we capture real social information from $384,494$ Facebook users. Relying on this large dataset, we conduct comprehensive experiments to evaluate our proposed method. The experiment results have demonstrated that our proposed approach is capable of improving content discovery in P2P not only for popular content but also for users' personal interests. In the future, we plan to extend the current solution by selecting friends regarding their social features as well as the features of the requested content, so as to make the mechanism more effective and intelligent. Besides, we will take into consideration more specific social features.

Appendix A. Parameter Optimization

To optimize the parameters of the social P2P network model, we minimize equation 7 with respect to the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The parameters are represented uniformly as $\boldsymbol{a}$ instead in this section. Therefore we calculate the derivative of equation 7 as:

$$\frac{\partial F(\boldsymbol{a})}{\partial \boldsymbol{a}} = 2\boldsymbol{a} + \sum_{k,r} \frac{\partial h(p_r - p_k)}{\partial \boldsymbol{a}} = 2\boldsymbol{a} + \sum_{k,r} \frac{\partial h(p_r - p_k)}{\partial (p_r - p_k)} (\frac{\partial p_r}{\partial \boldsymbol{a}} - \frac{\partial p_k}{\partial \boldsymbol{a}}) \tag{A.1}$$

Applying the commonly used hinge-loss function, i.e., $h(p_r - p_k) = [1 - (p_r - p_k)(\boldsymbol{a}^T(\boldsymbol{w}_{rv} - \boldsymbol{w}_{kv}))]_+$; thus we have $\frac{\partial h(p_r - p_k)}{\partial (p_r - p_k)} = [\boldsymbol{a}^T(\boldsymbol{w}_{rv} - \boldsymbol{w}_{kv})]_+$. To calculate $\frac{\partial p_u}{\partial \boldsymbol{a}}$, we obtain the initial probability vector at step 0 by sending queries to all the friends of the starting node and calculating the success rate. We denote the initial probability vector as $\boldsymbol{p^{(0)}}$. According to equation 5, we can iteratively compute the final probability given by:

$$\boldsymbol{p} = (1-\delta)\boldsymbol{A'}\boldsymbol{p^{(0)}} \tag{A.2}$$

where $\boldsymbol{A'}$ is the final random walk transition probability matrix. Note that $\boldsymbol{p}$ is the principal eigenvector of matrix $\boldsymbol{A'}$. A.1 can be rewritten as $p_u = \sum_i p_i \boldsymbol{A'}_{iu}$. Therefore the derivative of p_u with respect to $\boldsymbol{a}$ equals:

$$\frac{\partial p_u}{\partial \boldsymbol{a}} = \sum_j A'_{iu} \frac{\partial p_i}{\partial \boldsymbol{a}} + p_i \frac{\partial A'_{iu}}{\partial \boldsymbol{a}} \tag{A.3}$$

By recursively employing the chain rule to A.3, we can compute the derivative of p_u iteratively [7] [4] [23] [3].

Eventually, we apply the gradient descent method to minimize $F(a)$ directly:

$$\boldsymbol{a} := \boldsymbol{a} - \mu \frac{\partial F(\boldsymbol{a})}{\partial \boldsymbol{a}}$$

[1] Adamic, L., Adar, E., 2005. How to search a social network. Social Networks 27 (3), 187–203.

[2] Agarwal, A., Chakrabarti, S., 2007. Learning random walks to rank nodes in graphs. In: Proceedings of the 24th international conference on Machine learning. ICML '07. ACM, New York, NY, USA, pp. 9–16.

[3] Andrew, A. L., Jan. 1978. Convergence of an iterative method for derivatives of eigensystems. Journal of Computational Physics 26, 107–112.

[4] Andrew, A. L., 1979. Iterative computation of derivatives of eigenvalues and eigenvectors. IMA Journal of Applied Mathematics 288, 209–218.
URL http://dx.doi.org/10.1007/978-3-642-13422-7_7

[5] Asthana, H., Cox, I. J., 2012. Pac'npost: a framework for a micro-blogging social network in an unstructured p2p network. In: Proceedings of the 21st international conference companion on World Wide Web. WWW '12 Companion. ACM, New York, NY, USA, pp. 455–456.

[6] Backstrom, L., Boldi, P., Rosa, M., Ugander, J., Vigna, S., 2011. Four degrees of separation. CoRR abs/1111.4570.

[7] Backstrom, L., Leskovec, J., 2011. Supervised random walks: predicting and recommending links in social networks. In: Proceedings of the fourth ACM international conference on Web search and data mining. WSDM '11. ACM, New York, NY, USA, pp. 635–644.

[8] Bao, H., Chang, E. Y., 2010. Adheat: an influence-based diffusion model for propagating hints to match

ads. In: Proceedings of the 19th international conference on World wide web. WWW '10. ACM, New York, NY, USA, pp. 71–80.

[9] BitTorrent, ????
URL http://www.bittorrent.com/

[10] Chirita, P. A., Damian, A., Nejdl, W., Siberski, W., 2005. Search strategies for scientific collaboration networks. In: Proceedings of the 2005 ACM workshop on Information retrieval in peer-to-peer networks. P2PIR '05. ACM, New York, NY, USA, pp. 33–40.

[11] Craswell, N., Szummer, M., 2007. Random walks on the click graph. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '07. ACM, New York, NY, USA, pp. 239–246.

[12] Dan, G., Carlsson, N., Chatzidrossos, I., 31 2011-sept. 2 2011. Efficient and highly available peer discovery: A case for independent trackers and gossiping. In: Proceedings of 2011 IEEE International Conference on Peer-to-Peer Computing (P2P). pp. 290 –299.

[13] Doulkeridis, C., Nørvåg, K., Vazirgiannis, M., 2008. Peer-to-peer similarity search over widely distributed document collections. In: Proceedings of the 2008 ACM workshop on Large-Scale distributed systems for information retrieval. LSDS-IR '08. ACM, New York, NY, USA, pp. 35–42.

[14] Facebook, ????
URL http://www.facebook.com/

[15] Fast, A., Jensen, D., Levine, B. N., 2005. Creating social networks to improve peer-to-peer networking. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. KDD '05. ACM, New York, NY, USA, pp. 568–573.

[16] Ferretti, S., 2013. Gossiping for resource discovering: An analysis based on complex network theory. Future Generation Computer Systems 29 (6), 1631–1644.

[17] Gjoka, M., Kurant, M., Butts, C., Markopoulou, A., october 2011. Practical recommendations on crawling online social networks. Selected Areas in Communications, IEEE Journal on 29 (9), 1872 –1892.

[18] Gnutella, 2003. Gnutella protocol development.
URL http://rfc-gnutella.sourceforge.net/

[19] Gou, L., Chen, H.-H., Kim, J.-H., Zhang, X. L., Giles, C. L., 2010. Sndocrank: a social network-based video search ranking framework. In: Proceedings of the international conference on Multimedia information retrieval. MIR '10. ACM, New York, NY, USA, pp. 367–376.

[20] Haveliwala, T. H., 2002. Topic-sensitive pagerank. In: Proceedings of the 11th international conference on World Wide Web. WWW '02. ACM, New York, NY, USA, pp. 517–526.

[21] Jawhar, I., Wu, J., 2004. A two-level random walk search protocol for peer-to-peer networks. In: Proc.

of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics.

[22] Jeh, G., Widom, J., 2003. Scaling personalized web search. In: Proceedings of the 12th international conference on World Wide Web. WWW '03. ACM, New York, NY, USA, pp. 271–279.

[23] Jin, Y., Matsuo, Y., Ishizuka, M., 2010. Ranking learning entities on the web by integrating network-based features. In: Ting, I.-H., Wu, H.-J., Ho, T.-H. (Eds.), Mining and Analyzing Social Networks. Vol. 24(2) of Studies in Computational Intelligence. Springer Berlin Heidelberg, pp. 107–123.
URL http://dx.doi.org/10.1007/978-3-642-13422-7_7

[24] Karger, D., Lehman, E., Leighton, T., Panigrahy, R., Levine, M., Lewin, D., 1997. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web. In: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing. STOC '97. ACM, New York, NY, USA, pp. 654–663.

[25] KaZaa, ????
URL http://www.kazaa.com/

[26] Kleinberg, J., 2006. Complex networks and decentralized search algorithms.

[27] Konstas, I., Stathopoulos, V., Jose, J. M., 2009. On social networks and collaborative recommendation. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. SIGIR '09. ACM, New York, NY, USA, pp. 195–202.

[28] Kourtellis, N., Finnis, J., Anderson, P., Blackburn, J., Borcea, C., Iamnitchi, A., 2010. Prometheus: user-controlled p2p social data management for socially-aware applications. In: Proceedings of the ACM/IFIP/USENIX 11th International Conference on Middleware. Middleware '10. Springer-Verlag, Berlin, Heidelberg, pp. 212–231.

[29] Kourtellis, N., Iamnitchi, A., 31 2011-sept. 2 2011. Inferring peer centrality in socially-informed peer-to-peer systems. In: Proceedings of 2011 IEEE International Conference on Peer-to-Peer Computing (P2P). pp. 318 –327.

[30] Leskovec, J., Horvitz, E., 2008. Planetary-scale views on a large instant-messaging network. In: Proceedings of the 17th international conference on World Wide Web. WWW '08. ACM, New York, NY, USA, pp. 915–924.

[31] Li, Z., Shen, H., 10 2012. Social-p2p: An online social network based p2p file sharing system. In: Proceedings of the 20th IEEE International Conference on Network Protocols, (ICNP). pp. 1–10.

[32] Liu, L., Antonopoulos, N., Mackin, S., 2007. Fault-tolerant peer-to-peer search on small-world networks. Future Generation Computer Systems 23 (8), 921–931.

[33] Liu, L., Antonopoulos, N., Mackin, S., feb. 2007. Social peer-to-peer for resource discovery. In: Proceedings of the 15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing, 2007 (PDP '07). pp. 459 –466.

[34] Liu, L., Antonopoulos, N., Mackin, S., 6 2008. Managing peer-to-peer networks with human tactics in social interactions. Journal of Supercomputing 44 (3), 217–236.

[35] Liu, L., Xu, J., Russell, D., Townend, P., Webster, D., 2009. Efficient and scalable search on scale-free p2p networks. Peer-to-Peer Networking and Applications 2 (2), 98–108.

[36] Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., Lim, S., quarter 2005. A survey and comparison of peer-to-peer overlay network schemes. Communications Surveys Tutorials, IEEE 7 (2), 72–93.

[37] Maymounkov, P., Mazires, D., 2002. Kademlia: A peer-to-peer information system based on the xor metric. In: Druschel, P., Kaashoek, F., Rowstron, A. (Eds.), Peer-to-Peer Systems. Vol. 2429 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 53–65.

[38] Milgram, S., 1967. The small world problem. Psychology Today 1, 61–67.

[39] Mishra, A., Bhattacharya, A., 2011. Finding the bias and prestige of nodes in networks based on trust scores. In: Proceedings of the 20th international conference on World wide web. WWW '11. ACM, New York, NY, USA, pp. 567–576.

[40] Networks, P. A., 2012. The application usage and risk report.
URL `http://media.paloaltonetworks.com/documents/`

[41] Noel, J., Sanner, S., Tran, K.-N., Christen, P., Xie, L., Bonilla, E. V., Abbasnejad, E., Della Penna, N., 2012. New objective functions for social collaborative filtering. In: Proceedings of the 21st international conference on World Wide Web. WWW '12. ACM, New York, NY, USA, pp. 859–868.

[42] Page, L., Brin, S., Motwani, R., Winograd, T., Nov 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab.

[43] Pan, J.-Y., Yang, H.-J., Faloutsos, C., Duygulu, P., 2004. Automatic multimedia cross-modal correlation discovery. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '04. ACM, New York, NY, USA, pp. 653–658.

[44] PPLive, ????
URL `http://www.pplive.com/`

[45] PPStream, ????
URL `http://www.pps.tv/`

[46] Ruffo, G., Schifanella, R., Feb 2009. A peer-to-peer recommender system based on spontaneous affinities. ACM Transaction on Internet Technology 9 (1), 4:1–4:34.

[47] Sanchez-Artigas, M., Herrera, B., Sept 2011. Socialhelpers: Introducing social trust to ameliorate churn in p2p reputation systems. In: Proceedings of 2011 IEEE International Conference on Peer-to-Peer Computing (P2P). pp. 328–337.

[48] Shakimov, A., Varshavsky, A., Cox, L. P., Cáceres, R., 2009. Privacy, cost, and availability tradeoffs in decentralized osns. In: Proceedings of the 2nd ACM workshop on Online social networks. WOSN '09.

ACM, New York, NY, USA, pp. 13–18.

[49] Sihem Amer-Yahia, L. V. S. L., Yu, C., 2009. Socialscope: Enabling information discovery on social content sites. Computing Research Repository, CoRR abs/0909.2058.

[50] Skype, ????
URL http://www.superb.net/

[51] Smirnov, N. V., 1948. Table for estimating the goodness of fit of empirical distributions. The Annals of Mathematical Statistics 19, 279–281.

[52] Sripanidkulchai, K., Maggs, B., Zhang, H., march-3 april 2003. Efficient content location using interest-based locality in peer-to-peer systems. In: Proceedings of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2003). Vol. 3. pp. 2166–2176.

[53] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., Balakrishnan, H., 2001. Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications. SIGCOMM '01. ACM, pp. 149–160.

[54] Tigelaar, A. S., Hiemstra, D., Trieschnigg, D., May 2012. Peer-to-peer information retrieval: An overview. ACM Transactions on Information Systems 30 (2), 9:1–9:34.

[55] Tong, H., 2006. Fast random walk with restart and its applications.

[56] Traverso, S., Huguenin, K., Triestan, I., Erramilli, V., Laoutaris, N., Papagiannaki, K., 2012. Tailgate: handling long-tail content with a little help from friends. In: Proceedings of the 21st international conference on World Wide Web. WWW '12. ACM, New York, NY, USA, pp. 151–160.

[57] UUSee, ????
URL http://www.uusee.com/

[58] Wang, C., Xiao, L., feb. 2007. An effective p2p search scheme to exploit file sharing heterogeneity. IEEE Transactions on Parallel and Distributed Systems 18 (2), 145 –157.

[59] Wang, H., Wang, F., Liu, J., Lin, C., Xu, K., Wang, C., September 2013. Accelerating peer-to-peer file sharing with social relations. IEEE Journal on Selected Areas in Communications 31 (9), 66–74.

[60] Wang, Z., Sun, L., Yang, S., Zhu, W., 2011. Prefetching strategy in peer-assisted social video streaming. In: Proceedings of the 19th ACM international conference on Multimedia. MM '11. ACM, New York, NY, USA, pp. 1233–1236.

[61] Zhang, H., Zhang, L., Shan, X., Li, V., june 2007. Probabilistic search in p2p networks with high node degree variation. In: Communications, 2007. ICC '07. IEEE International Conference on. pp. 1710 –1715.

[62] Zhang, Y., Liu, L., 2013. Distance-aware bloom filters: Enabling collaborative search for efficient resource discovery. Future Generation Computer Systems 29 (6), 1621–1630.

[63] Zhang, Y., Shen, G., Yu, Y., june 2007. Lips: Efficient p2p search scheme with novel link prediction techniques. In: Communications, 2007. ICC '07. IEEE International Conference on. pp. 1875 –1880.