A Recommender Scheme for Peer-to-Peer Systems

Loubna Mekouar
University of Waterloo
Waterloo, Canada
lmekouar@bbcr.uwaterloo.ca

Youssef Iraqi
Dhofar University
Salalah, Oman
y_iraqi@du.edu.om

Raouf Boutaba
University of Waterloo
Waterloo, Canada
rboutaba@bbcr.uwaterloo.ca

Abstract

In Peer-to-Peer (P2P) file sharing systems, peers spend a significant amount of time looking for relevant files. However, the files available for download represent on one hand a rich collection and on the other hand a struggle for the peers to find files that they like. In this paper, we propose "Asymmetric Peers' Similarity Based Recommendation with File Popularity" scheme that helps peers find and discover new and interesting files. To overcome the problems of traditional collaborative filtering recommender systems, an implicit rating approach is used. Simulation results confirm the effectiveness of the proposed scheme in providing accurate recommendations.

1 Introduction

Peer-to-Peer (P2P) file sharing systems have emerged as a new way of sharing files. However, users are overwhelmed by a huge selection of files available for download. Unfortunately, users have to struggle to choose the right items that are of interest to them. Using a recommender system will help users to search through this large selection of files. Based on peers' behavior, the recommender system will suggest items that they will most probably like. These peers will be motivated to download the recommended files and hence, will remain active members. While they are downloading the files, they will upload files to others increasing their contribution to the system [4].

Although, E-Commerce applications like amazon.com, BizRate.com, and Netflix.com [6, 2, 7] have been using and benefiting from recommender systems for a long time, recommender schemes are still a fertile area in P2P environments. Only few research works [8, 5] have been proposed. However, these are suitable for decentralized P2P systems but not for the partially decentralized systems (like KaZaA, Gnutella2) that are the most popular. In this paper, we propose a novel recommender scheme for these systems.

The paper is organized as follows. Section 2 highlights important recommender schemes used in E-Commerce. Section 3 describes the proposed recommender scheme. Section 4 presents the performance evaluation. Finally, Section 5 concludes the paper.

2 Recommender Systems in E-Commerce

Collaborative filtering [6, 3, 7, 8] is the most widely used technique for recommender systems in E-Commerce. This approach is based on collecting users' ratings. It suggests items based on similarities between the active user's profile and other users or similarities between items. The main known problems of collaborative filtering are:

- *Cold start*: This problem occurs for a new user or at the start of the system. It is difficult to make recommendations for a new user since no rating is provided.
- *Popularity effect*: This problem occurs when the given recommendations are obvious and evident from the user's point of view.
- *Data sparseness*: This problem occurs when only few users have rated few items. It is difficult to predict the user's interests and make accurate recommendations.
- *Trust*: This problem occurs when untrustworthy users provide false ratings. Only highly reputable users must be chosen while making recommendations.

3 The proposed Recommender Scheme

3.1 Explicit Rating versus Implicit Rating

In E-Commerce, the collaborative filtering technique is based on the ratings by the customers of the products and/or the purchases they made. In P2P file sharing systems, the collaborative filtering technique can be used based on the ratings by the users of the files they have. We can distinguish two approaches to rating: *explicit rating* and *implicit rating*.

978-0-7695-3297-4/08 $25.00 © 2008 IEEE
DOI 10.1109/SAINT.2008.99

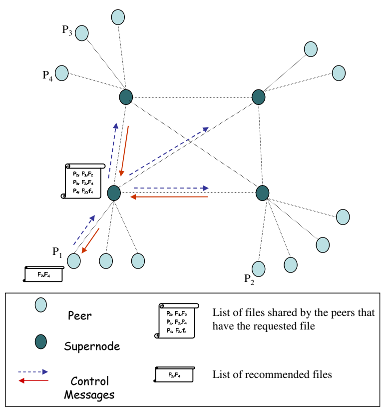

Figure 1. Example of the information flow

In the *explicit rating* approach, the user has to explicitly provide a rating for each file he/she downloads according to its content (i.e. matches the user's preferences or not). This approach necessitates additional effort from the users. A rating scheme from 1 (not interesting at all) to 5 (very interesting) can be useful to assure recommendation accuracy. Users have to provide their ratings for different files to enrich the system with different opinions and experiences. Since *explicit rating* solicits an additional effort from users, it is difficult to enforce especially in systems where 70% are free riders [1]. This approach will likely suffer from the *cold start* and *data sparseness* problems. Also, *explicit rating* provides malicious peers with a way to influence the rating system which may lead to the *trust* issue as described in section 2.

The *implicit rating* approach does not require the users to rate the files. It assigns ratings implicitly (i.e. automatically without involving the user) based on users' profile. We propose to assign a rating of 1 (*I like it*) to the files owned by the user. All other files are assigned a rating of 0 (*I do not know*). Note that a rating of 0 does not mean that the user does not like the file.

We adopt the *implicit rating* approach since it solves the problems of collaborative filtering in E-Commerce. A ratings of 0 or 1 is always automatically available for every file, solving both the *Cold start* and the *Data sparseness* problems and minimizing the impact of the *Trust* problem. The *popularity effect* problem is solved by not recommending files already owned by the user.

3.2 User-based Collaborative Filtering

In this paper, we consider partially decentralized P2P systems. In these systems, peers connect to their supernodes that index shared files and proxy search requests on behalf of these peers. Figure 1 depicts an example of the information flow between the peer P_1 requesting a file and its supernode. After receiving a request from peer P_1, and assuming the file is not found locally, its supernode sends a request to other supernodes. These supernodes will send back the search result which is a list of peers that have the requested file and the files that these peers are sharing. Based on this information, the supernode of P_1 will use the proposed recommender scheme to generate a list of recommended files.

Since recommendations are given to peers in real time, it is preferable to explore relationships between peers rather than between files. Indeed, the partial search performed by supernodes limits the number of peers in the search result. This number is much less than the number of files shared by all the peers in the system. For these reasons, using user-based collaborative filtering in P2P systems is more practical than using item-based collaborative filtering algorithms.

3.3 Formal Notations

In the remaining of the paper, we will use the following formal notations:

Let P be the set of all peers in the system.

Let F be the set of all files shared by the peers.

Let p_i be the requestor peer looking for a file f_x. p_i is the user to whom the recommendation will be made.

Let P_{f_x} be the set of peers that possess the file f_x.

Let $FP_{P_{f_x}}$ be the set of files that these peers possess in addition to f_x. This is the set of files that these peers are sharing.

Let $f : P \rightarrow \Omega(F)$, such that $f(p_j)$ is the set of files held by peer p_j for every j and $\Omega(F)$ is the power set of F. Then we have:

$$F_{P_{f_x}} = \bigcup_{p_k \in P_{f_x}} f(p_k)$$

3.4 Files' Popularity Based Recommendation (FP)

This technique will allow a peer to discover the files that are more popular.

Let $G_{p_i} = F_{P_{f_x}} - f(p_i) - \{f_x\}$ be the set of files that p_i does not have from the set $F_{P_{f_x}}$ not including the file f_x. These are all the files owned by the peers in P_{f_x} that the peer p_i does not have.

For every file $f_k \in G_{p_i}$, we define its popularity as:

$$Pop(f_k) = \frac{|P_{f_k} \cap P_{f_x}|}{|P_{f_x}|}$$

where $|P|$ is the cardinality of the set P.

The value of $Pop(f_k)$ is a numerical score that shows the popularity of the file f_k among the peers in P_{f_x}.

In this technique, files that are more popular will be recommended. This scheme will recommend only files f_k such that $Pop(f_k) \geq t_1$, where t_1 is a threshold. This recommendation list is sorted according to the popularity of the files $Pop(f_k)$ with the files that are most popular at the top of the list. The supernode of peer p_i may keep track of these files for future recommendations. This technique will accelerate significantly the spread of popular files.

3.5 Asymmetric Peers' Similarity Based Recommendation with File Popularity

Peers' similarity is an important factor in this technique. To be able to make accurate recommendations, we compare the active user's files against those of other users. The goal of this process is to find peers with similar preferences as the active peer p_i and make recommendations based on the files that they have. In fact, we apply the files' popularity approach within these peers.

For every p_j in P_{f_x} we define the similarity relationship as:

$$ASim_{p_i}(p_j) = \frac{|f(p_i) \cap f(p_j)|}{|f(p_i)|}$$

We assume that $|f(p_i)|$ is not null, which means that the peer p_i owns at least one file. If the peer does not own any file, the *FP* scheme is used.

The value of $ASim_{p_i}(p_j)$ is a numerical score that shows how similar the peer p_j is to the peer p_i. Note that this similarity relationship is not symmetric, i.e. $ASim_{p_i}(p_j)$ may not be equal to $ASim_{p_j}(p_i)$

This scheme will choose only peers that have $ASim_{p_i}(p_j) \geq t_2$. Where t_2 is a threshold.

Let $S_{p_i}^{t_2} = \{p_j, p_j \in P_{f_x} \; and \; ASim_{p_i}(p_j) \geq t_2\}$

We apply the *FP* within the set $S_{p_i}^{t_2}$ of peers most similar to peer p_i. For every file, we compute:

$$Pop_{ASim}(f_k) = \frac{|P_{f_k} \cap P_{f_x} \cap S_{p_i}^{t_2}|}{|P_{f_x} \cap S_{p_i}^{t_2}|}$$

Note that if $t_2 = 0$ then $Pop_{ASim}(f_k) = Pop(f_k)$.

This scheme will recommend only files f_k such that $Pop_{ASim}(f_k) \geq t_1$, where t_1 is a threshold. This recommendation list is sorted according to the popularity of the files $Pop_{ASim}(f_k)$ with the files that are most popular at the top of the list. Both t_1 and t_2 are application dependant values. The higher these values are, the more exigent we are in recommending files to the peers in the system.

4 Performance Evaluation

We have simulated the following techniques:

- *Random Based Recommendation (RBR)* where peers will choose files randomly. Although users usually tend to search and download files according to their choices and needs, we wanted to see the effect of recommender schemes on the P2P system.
- *Files' Popularity Based Recommendation (FP)*
- *Asymmetric Peers' Similarity Based Recommendation with File Popularity (ASFP)*

For each scheme, we compute the following metrics:

- For each peer's category, the ratio of the number of downloaded files that belong to this category over all the files downloaded by the peers from this category.
- For each peer's category, the ratio of the number of downloaded files that belong to other categories over the files downloaded by the peers from this category.

4.1 Simulation Parameters

The simulation parameters are as follows:

- We simulate a system with 1,000 peers and 1,000 files.
- Peers are divided into four interest categories (C1: Action, C2: Romance, C3: Drama, and C4: Comedy) and files are also divided into the same four categories. Each peer belongs to one category. For this reason, peers prefer to have most of the files from their category and only few files from other categories. At the beginning of the simulations, each peer will get files from the category that she/he prefers with a probability of 0.9 and files from other categories with a probability of 0.1. At the beginning, each peer has at most 50 files and each file has at least one owner. This allows for a maximum of 50,000 files.
- The percentage of peers in each category is 25%, and the percentage of files in each category is 25%.
- The maximum number of files proposed in each recommendation is set to 5. We also keep history of previously recommended files.
- We simulate 200,000 requests in each simulation. Simulations are repeated 10 times and the results presented are the average values.

Following the simulation parameters, peers with indices from 1 to 250 belong to category C1 (Action), peers with indices from 251 to 500 belong to category C2 (Romance). Peers with indices from 501 to 725 belong to C3 (Drama) and peers with indices from 726 to 1,000 are peers that belong to C4 (Comedy).

4.2 Simulation Results

In the random selection scheme, no recommendation is provided. At the beginning of the simulation, at most 50 files are distributed to each peer with 90% of them from the peer's category and 10% from other categories. During the simulation about 200 files are downloaded by each peer from random categories. This means that 25% of the 200 files will belong to the peer's category and 75% of them will belong to other categories. So in total, the peer will have about 95 ($50 \times 90\% + 200 \times 25\%$) files from her/his category (about 35%) and about 155 ($50{\times}10\%{+}200{\times}75\%$) from other categories (about 65%).

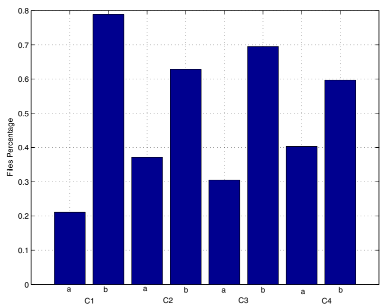

Figure 2. Results when using FP

Figure 2 depicts the percentage of files for each peer category when using the FP scheme. The X axis represents the four peers' categories while the Y axis represents the percentage of files for each peer category. In the X axis, the (a) bars show the percentage of files from the same category as the peers' category while the (b) bars show the percentage of files from other categories. In the first peer category (C1), peers possess 21% of files from the category that they prefer (a) and 79% from other categories (b). This technique shows that Files Popularity does not fulfill peers expectations and does not match their preferences. However, the benefit from this scheme is that peers are aware of the most popular files within the file sharing system, hence, this scheme is effective in spreading popular files.

Figure 3 depicts the percentage of files for each peer category based on the ASFP scheme. This figure shows clearly the effectiveness of this recommender scheme. In the first peer category (C1), peers have 89% from the first category of files that they like (a) and only 11% from all other files categories (b). From this figure, it is clear that peers have more files from the category that they prefer since (a) bars are higher than (b) bars for all categories.

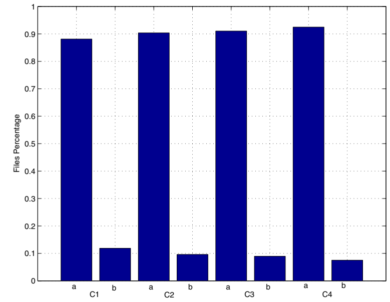

Figure 3. Results when using ASFP

5 Conclusion

In this paper, we propose a new recommender scheme for partially decentralized P2P file sharing systems. This scheme helps the peers discover relevant files, hence, increasing peers satisfaction. While the peers are downloading these files, they will upload files to other peers. The proposed scheme is proactive, easy to understand by users and easy to implement. As future work, we plan to include metrics to identify the degree of likeliness of a file by a peer. This way, providing more information about users' experiences on items. We also plan to take into consideration the reputation of peers when computing the similarity metric.

References

[1] E. Adar and B. A. Huberman. Free Riding on Gnutella. Technical report, 2000. http://www.hpl.hp.com/research/idl/papers/gnutella/.

[2] G. Linden, B. Smith, and J. York. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. In *IEEE Internet Computing*, pages 76–80, Jan/Feb 2003.

[3] P. Massa and P. avesani. Trust-aware Collaborative Filtering for Recommender Systems. In *International Conference on Cooperative Information Systems*, 2004.

[4] L. Mekouar, Y. Iraqi, and R. Boutaba. Free Riders under Control through Service Differentiation in Peer-to-Peer Systems. In *IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2005.

[5] G. Ruffo, R. Schifanella, and E. Ghiringhello. A Decentralized Recommendation System based on Self-Organizing Partnerships. In *IFIP-Networking*, 2006.

[6] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Analysis of Recommendation algorithms for E-commerce. In *EC*, 2000.

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-Based Collaborative Filtering Recommendation Algorithms. In *WWW*, 2001.

[8] J. Wang, J. Pouwelse, R. L. Lagendijk, and M. J. T. Reinders. Distributive Collaborative Filtering for Peer-to-Peer File Sharing Systems. In *SAC*, 2006.