Multimed Tools Appl
DOI 10.1007/s11042-010-0612-1

An analysis of peer similarity for recommendations in P2P systems

Loubna Mekouar · Youssef Iraqi · Raouf Boutaba

Abstract In this paper, we propose a novel recommender framework for partially decentralized file sharing Peer-to-Peer systems. The proposed recommender system is based on *user-based collaborative filtering*. We take advantage from the partial search process used in partially decentralized systems to explore the relationships between peers. The proposed recommender system does not require any additional effort from the users since implicit rating is used. The recommender system also does not suffer from the problems that traditional collaborative filtering schemes suffer from like the *Cold start* and the *Data sparseness*. To measure the similarity between peers, we propose *Files' Popularity Based Recommendation (FP)* and *Asymmetric Peers' Similarity Based Recommendation with File Popularity (ASFP)*. We also investigate similarity metrics that were proposed in other fields and adapt them to file sharing P2P systems. We analyze the impact of each similarity metric on the accuracy of the recommendations. Both weighted and non weighted approaches were studied.

Keywords Recommender system · Similarity metrics · Personalized recommendations · Content adaptation · Multimedia files · Peer-to-Peer systems

Preliminary versions were accepted in the IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises, Collaborative Peer-to-Peer Systems Workshop (COPS), 2008 and in the tenth IEEE International Conference on Computer and Information Technology (CIT), 2010.

L. Mekouar (✉) · R. Boutaba
David R. Cheriton School of Computer Science, University of Waterloo,
200 University Avenue West, Waterloo, ON, Canada N2L 3G1
e-mail: lmekouar@bbcr.uwaterloo.ca

R. Boutaba
Division of IT Convergence Engineering, POSTECH, Pohang, 790-784, Korea
e-mail: rboutaba@bbcr.uwaterloo.ca

Y. Iraqi
Department of Computer Engineering, Khalifa University, Sharjah, UAE
e-mail: Youssef.Iraqi@kustar.ac.ae

Published online: 06 October 2010

1 Introduction

The most important challenges of online environments is to assure satisfactory transactions. People will usually back up from meeting new strangers and buying new items that they did not know or try before. Therefore, people minimize their interactions and tend to remain in their comfort zone. Positive interactions can be achieved through reputation and recommendation systems. Reputation and recommender systems are most commonly found in e-Commerce applications. A positive interaction between strangers can be achieved by relying on reputation. Reputation is used to rate people. It is a valuable information that helps reduce the offensive and deceptive behavior of online users. While reputation systems are used to enforce appropriate behavior, recommender systems are used to allow satisfactory transactions by rating the quality of items (e.g., products, services).

In P2P file sharing systems, users are overwhelmed by a large collection of multimedia files available for download. Unfortunately, finding files of interest is time consuming. Recommender systems suggest to users files based on their profile. These users 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 [15].

Recommender systems are widely used in e-Commerce applications (e.g., amazon.com, BizRate.com, Epinions.com, yahoo.com) [12, 21, 22]. Recommender systems take advantage of the collected data that represents customers' experiences to predict their future needs. These systems suggest products and services that most likely will be of interest to the customers. The collaborative filtering recommender techniques are achieving widespread success on the web [14, 21, 22].

Although, e-Commerce applications have been using recommender systems for at least a decade, this research field is still a fertile area in P2P systems. Only few research works have addressed recommender schemes in P2P systems [19, 25, 26].

In P2P file sharing systems, peers spend a significant amount of time looking for relevant and interesting files. However, the files available for download represent on one hand a rich collection for different needs and preferences and on the other hand a struggle for the peers to find files that they like. The search process in partially decentralized P2P file sharing systems can be divided into the following steps:

1. Providing keywords.
2. Sending the request to the supernode.
3. The supernode will search for the file by contacting local peers if the file is available locally or sending a request to other supernodes.

In this paper, we propose a novel recommender framework for partially decentralized file sharing P2P systems. This recommender framework will help peers in the first step by finding relevant filenames for files of interest based on peers' profile. The profile reflects their past choices, experiences and preferences. The proposed recommender system is based on *collaborative filtering*. Peers collaborate to filter out irrelevant files and find interesting ones. Relationships between peers are explored by taking advantage from the partial search process used in partially decentralized systems. In order to make personalized recommendations, the implicit rating approach is used and hence, no additional effort is required from the users. In addition, the implicit rating helps in overcoming the problems that traditional collaborative filtering schemes suffer from like the *Cold start*, the *Data sparseness* and the *Popularity effect*.

Similarity has been used in many fields like natural and social sciences as well as engineering and statistics. Several metrics have been proposed to compute similarity. In this paper, we investigate several similarity metrics in the context of P2P recommender systems. We adapt these metrics to the context of recommender systems and particularly to file sharing P2P systems. We also propose a new similarity metric. We investigate these similarity metrics in both the weighted and non weighted techniques. The impact of each similarity metric on the accuracy of the recommendations is analyzed.

The paper is organized as follows. Section 2 provides an overview of recommendations in e-Commerce. Section 3 highlights related works in P2P systems. Section 4 describes the proposed recommender framework. Section 5 describes the similarity metrics used in this work. Section 6 describes the performance evaluation conducted and presents an analysis of the results. Finally, Section 7 concludes the paper.

2 Recommender systems in e-commerce

2.1 Collaborative filtering

Collaborative filtering is the most widely used technique for recommender systems. 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. In this approach, it is required that a large number of users rate items to ensure recommendation accuracy [14]. This technique has proved to be one of the most successful techniques in recommender systems in recent years.

Collaborative filtering can be divided into two main categories:

- User-based collaborative filtering algorithms: relationships between users are explored first to find similar users to the active user. These users are like-minded as the active user and based on their ratings of the item in question, a rating value is predicted. This value represents an estimation of the likeliness of the item in question by the active user.
- Item-based collaborative filtering algorithms: relationships are explored between items first rather than users. Items that are similar to the item in question are identified. Based on this similarity, a predicted rating is provided. The item–to–item recommender scheme used in amazon.com is an example of this approach.

2.2 User-based collaborative filtering

User-based collaborative filtering has the following steps:

- Consider the user-item matrix where each row represents the profile of a user and each column represents the users that have the item (e.g., purchased, rented, ...etc).
- Compute the similarity metric: the similarity metric is computed for each pair of users. This is used to predict the ratings for the active user A. Similar peers have

almost similar tastes and preferences, and so, they will have similar ratings for the same items. For N users, a user similarity matrix ($N \times N$) is computed.

- Choose the k most similar users to the active user A. These users are called neighbors of user A.
- Compute the predicted rating for the active user A for items i. These items are not yet purchased by the active user A.
- Recommend the items that have a high predicted rating value to the active user A.

The similarity matrix is usually computed using the *Pearson correlation* or the *Cosine measure* [14, 21, 22].

2.3 Using the Pearson correlation

To compute the similarity of peers, the most used technique is Pearson correlation coefficient (PC) [14, 21, 22, 25]:

$$PC_{A,B} = \frac{\sum_{l=1}^{m}(R_{Al} - \bar{R}_A)(R_{Bl} - \bar{R}_B)}{\sqrt{\sum_{l=1}^{m}(R_{Al} - \bar{R}_A)^2 \sum_{l=1}^{m}(R_{Bl} - \bar{R}_B)^2}} \tag{1}$$

Where:

- A is the active user for whom a recommendation will be proposed.
- B is a user.
- $\bar{R}_B$ is the average rating by the user B.
- $\bar{R}_A$ is the average rating by the user A.
- m is the number of items that they both rated.
- R_{Al} is the rating given by the user A to the item l.
- R_{Bl} is the rating given by the user B to the item l.

The Pearson correlation coefficient is significant if users share two or more ratings. Positive correlation shows similarities between users, while a negative value shows that these users are not similar and their interests are different.

In the similarity matrix, each row represents the similarity between a user and other users in terms of ratings. This matrix is used to predict the ratings for the current user. This is based on the assumption that if two users have similar preferences and interests, they will have similar ratings.

2.4 Using the cosine measure

Another way to compute similarity between users is to use the cosine measure [21, 22]. The active user A and a user B are represented by two vectors and the similarity between them is measured by computing the cosine of the angle between the two vectors.

$$\cos(\overrightarrow{A}, \overrightarrow{B}) = \frac{\overrightarrow{A} \cdot \overrightarrow{B}}{\parallel \overrightarrow{A} \parallel_2 \times \parallel \overrightarrow{B} \parallel_2} = \frac{\sum_{l=1}^{n} R_{Al} R_{Bl}}{\sqrt{\sum_{l=1}^{n} R_{Al}^2 \sum_{l=1}^{n} R_{Bl}^2}} \tag{2}$$

$\overrightarrow{A} \cdot \overrightarrow{B}$ denotes the dot product between the vectors $\overrightarrow{A}$ and $\overrightarrow{B}$, $\| \overrightarrow{A} \|_2$, $\| \overrightarrow{B} \|_2$ represent the Euclidean norm for the two vectors and R_{Al}, R_{Bl} represent their respective ratings for the n items.

2.5 Computing the predicted rating

The predicted rating value measures the likeliness of the active user A for an item l. The predicted rating value is computed as follows [14, 21, 22]:

$$PR_{A,l} = \bar{R}_a + \frac{\sum_{j=1}^{k} PC_{A,j}(R_{jl} - \bar{R}_j)}{\sum_{j=1}^{k} PC_{A,j}} \tag{3}$$

k is the number of users that are the neighbors of the active user A. These users are the most similar to the active user A.

Another alternative for computing the predicted rating is to use the cosine measure instead of the Pearson correlation coefficient $PC_{A,B}$.

The items that have a high value of $PR_{A,l}$ are recommended to the active user A.

2.6 Challenges of collaborative filtering algorithms

The main known problems of collaborative filtering are the followings [14, 21, 22]:

- *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 based on users' similarities since no rating is provided yet or the user's profile is not known yet.
- *Popularity effect*: This problem occurs when popular items will become even more popular as they will be more recommended.
- *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. The system should be able to choose only highly reputable users while making recommendations. This will reduce the impact of untrustworthy users that influence badly the recommendation accuracy and hence, will increase the trust given by the peers to the recommender system.

3 Recommender schemes in P2P systems

3.1 Related works

In [19], the authors propose a decentralized recommendation system that takes advantage of the high clustering coefficient of Preference Networks. The nodes of these networks are users of a file sharing system and the links are connections between pairs of nodes that share one or more identical files. The authors experimentally prove that the preference networks are small worlds. They propose a recommendation scheme based on the fact that nodes can be naturally gathered together on the basis of common interests. The top-N ranked items are recommended to the user and the location information in the buddy tables can be used to locate the recommended items.

In [25], the authors propose a distributed collaborative filtering method that is self-organizing and operates in a distributed way. Similarity ranks between items are computed and are stored locally in buddy tables. The buddy tables store the information about the top-N relevant items. This information can be used to locate the recommended files. In addition, the buddy tables automatically create a self-organizing semantic overlay that cluster similar multimedia files. To perform a recommendation for a given user, the buddy tables for all the items in user's profile are downloaded and the relevance ranks are computed based on a user-content relevance model. Based on this work, the authors in [26], introduce personalization on Tribbler, a P2P television system. In this work, buddyCast which is a distributed profile exchanger, generates a semantic overlay by clustering peers into social networks according to their profile. Periodically, a user connects either to one of his buddies to exchange social networks and current profile list (exploitation) or to a new randomly chosen user from the random cache to exchange this information (exploration). A ranked list is created based on the similarity of their profile with the profile of the active user. The buddy list of the selected user is merged and the top-N best ranked users are kept.

These recommender schemes are suitable for decentralized P2P systems but not for partially decentralized systems. In addition, theses schemes generate a significant amount of overhead to make files' recommendations. As an example, in [26], it is required to maintain the following lists by each peer in the system: the top-N most similar users, the top-N most fresh random IP addresses and the K most recently visited users. The periodic exchange and update of information between peers is costly.

3.2 Advantages of recommender systems

In P2P file sharing systems, the goal from using a recommender system is to achieve the following advantages:

- Attract more users by making the search process easier, and more efficient.
- Increase peers' satisfaction by informing them about files of interest.
- Increase peers' contribution to the system since peers will be motivated to stay connected to download the recommended files and upload files to other peers. Free riders may be motivated to share their files to get a profile that reflects their preferences in order to receive accurate recommendations.
- Preserve network resources since peers will not have to download a large number of files that they do not like and will just discard.

3.3 Evaluation of recommender systems

Several features can be taken into consideration for recommender schemes evaluation [21, 22]:

- The additional effort that users are required to make: Explicit rating requires users to participate by providing ratings. However, there is no guarantee that users will make such commitment. Therefore, taking information implicitly from users' behavior is more preferable.

- Ease of understanding by users: when the recommender scheme is not understood by the users, they will not trust the recommendations. Users need to understand how recommendations are made. The simplicity of the recommender scheme plays an important role for its success.
- The accuracy of the recommendations: providing accurate recommendations will increase peers' satisfaction.
- Ease of designing and maintaining the proposed system.
- Performance issues: the scheme should not suffer from scalability, *Cold start* and *Data sparseness* as it is the case in traditional collaborative filtering schemes.

4 The proposed recommender framework

In e-Commerce applications, the collaborative filtering technique is based on the ratings of the products provided by the customers. In P2P file sharing systems, the collaborative filtering technique can be used based on the ratings of the files provided by the users.

4.1 Implicit rating versus explicit rating

After downloading a file, two rating approaches can be considered: explicit rating and implicit rating. In the *explicit rating* approach, the user has to explicitly provide a rating for each file she/he downloads according to its content (i.e., matches the user's preferences or not). This approach necessitates an 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% of peers are free riders [1]. This approach will likely suffer from the *Cold start* and *Data sparseness*. Also, *explicit rating* provides malicious peers with a way to influence the rating system which may lead to the *Trust* issue described in Section 2.6.

The *implicit rating* approach does not require the users to rate the files. It assigns ratings implicitly. The fact that ratings are generated automatically without involving users, alleviate them from the burden of explicitly providing ratings for each file they have downloaded. 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 in the proposed framework since it solves the problems of collaborative filtering in e-Commerce. The *implicit rating* approach has the following advantages:

- It solves the *Cold start* problem: Indeed, even at the start of the system or when a new user joins the P2P system, a ratings of 0 or 1 is always automatically available for every file.
- It avoids the *Data sparseness* problem as ratings are available.
- The subjective rating of files opens the door to malicious peers to manipulate files' recommendation. The *implicit rating* approach avoids tampering with

the ratings of the files by malicious peers which reduces their impact on the recommender system and thereby avoiding the *Trust* problem.

4.2 User-based versus item-based collaborative filtering

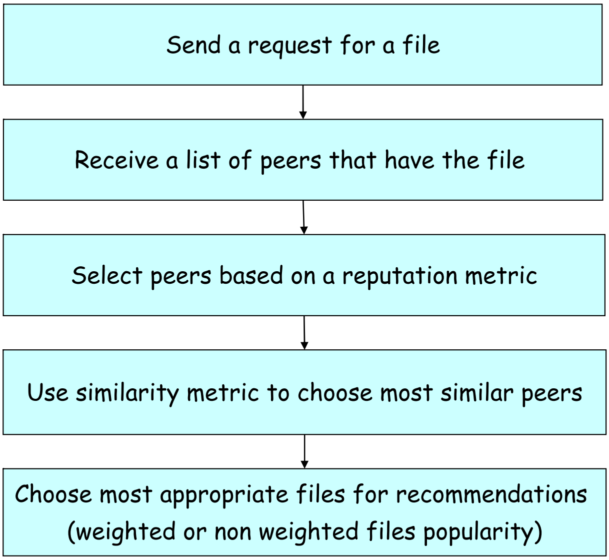
Figure 1 depicts the steps required in the proposed framework to make recommendations to the peers. During the life cycle of a transaction in a P2P system, the following steps are performed:

1. Send a file request
2. Receive a list of peers that have the requested file
3. Use similarity metric to choose most similar peers to the peer requesting the file (the active peer)
4. Use the weighted or non weighted files' popularity to choose most appropriate files for recommendations. The non weighted file popularity approach selects the most popular files among the selected similar peers independently from how similar the peers are to the active peer. The weighted approach uses the similarity metric to compute a weighted file popularity before suggesting files for recommendation.

The similarity metrics considered in this work are used during step 3, and the weighted and non weighted approaches have been enforced in step 4.

Figure 2 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.

Fig. 1 Recommendation transaction life cycle


Send a request for a file
Receive a list of peers that have the file
Select peers based on a reputation metric
Use similarity metric to choose most similar peers
Choose most appropriate files for recommendations (weighted or non weighted files popularity)

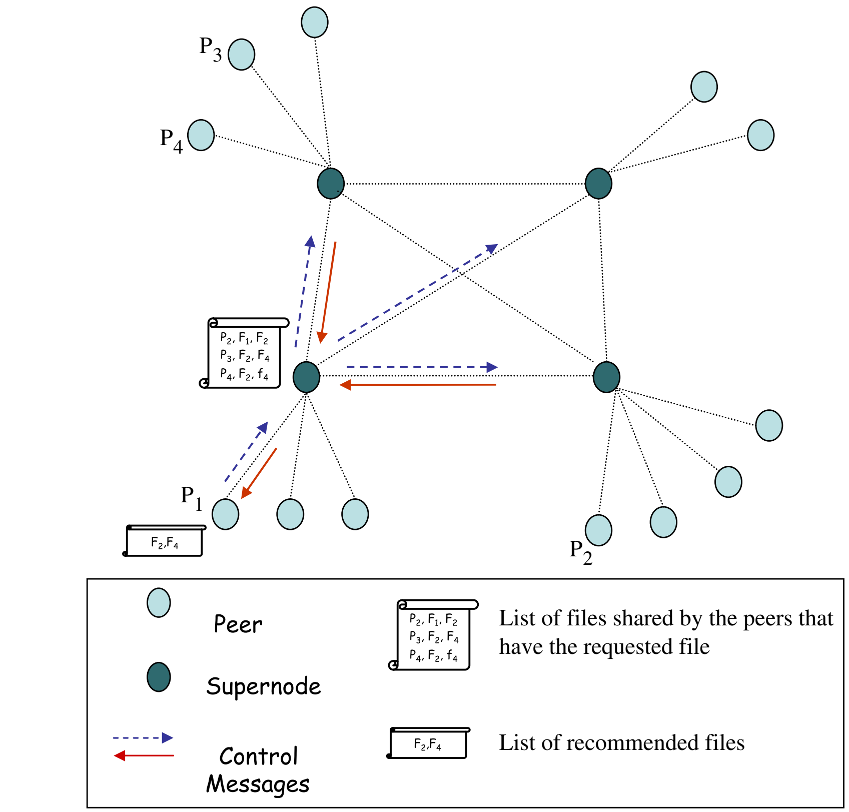

P3
P4
P2, F1, F2
P3, F2, F4
P4, F2, f4
P1
F2,F4
P2
Peer
Supernode
Control Messages
List of files shared by the peers that have the requested file
List of recommended files

Fig. 2 The proposed recommender framework

Since recommendations are given to peers in real time, it is preferable to explore relationships between peers rather than between files. We take advantage from the partial search process used in partially decentralized systems. 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. In addition, finding relationships between all files is time consuming and is usually done offline. For these reasons, adopting user-based collaborative filtering in P2P systems is more practical than using item-based collaborative filtering algorithms.

5 The similarity metrics

5.1 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 requester 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 $F_{P_{f_x}}$ 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:

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

5.2 Files' popularity based recommendation (FP)

This technique will allow a peer to discover the files that are more popular within the peers that have the requested file.

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:

$$\mathrm{Pop}(f_k) = \frac{|P_{f_k} \cap \ P_{f_x}|}{|P_{f_x}|} \tag{4}$$

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

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

In this technique, files f_k that are more popular will be recommended such that $\mathrm{Pop}(f_k) \geq t_1$, where t_1 is a threshold. This recommendation list is sorted according to the popularity of the files $\mathrm{Pop}(f_k)$ 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 which will increase peers' satisfaction.

5.3 Asymmetric peers' similarity based recommendation (AS)

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:

$$\mathrm{ASim}_{p_i}(p_j) = \frac{|f(p_i) \cap \ f(p_j)|}{|f(p_i)|} \tag{5}$$

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 $\mathrm{ASim}_{p_i}(p_j)$ 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., $\mathrm{ASim}_{p_i}(p_j)$ may not be equal to $\mathrm{ASim}_{p_j}(p_i)$.

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

Let $S^{t_2}_{p_i} = \{p_j, p_j \in P_{f_x} \textit{ and } \mathrm{ASim}_{p_i}(p_j) \geq t_2\}$

5.3.1 Asymmetric peers' similarity with file popularity (ASFP)

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

$$\mathrm{Pop}_{\mathrm{ASim}}(f_k) = \frac{|P_{f_k} \cap\ P_{f_x} \cap\ S^{t_2}_{p_i}|}{|P_{f_x} \cap\ S^{t_2}_{p_i}|} \tag{6}$$

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

This scheme will recommend only files f_k such that $\mathrm{Pop}_{\mathrm{ASim}}(f_k) \geq t_1$, where t_1 is a threshold. This recommendation list is sorted according to the popularity of the files $\mathrm{Pop}_{\mathrm{ASim}}(f_k)$ with the files that are most popular at the top of the list. Both t_1 and t_2 are application dependent values.

5.3.2 Asymmetric peers' similarity with weighted file popularity (ASWFP)

We apply the *Weighted File Popularity* technique within the set $S^{t_2}_{p_i}$ of peers most similar to peer p_i.

In this technique, we weight the files owned by the peers within the set $S^{t_2}_{p_i}$ of peers most similar to peer p_i according to peers' similarity. For every file, we add the similarity value for each peer P_j that owns this file and then we divide by the sum of all peers' similarities for peers that belong to the set $S^{t_2}_{p_i}$.

For every file, we compute:

$$\mathrm{WPop}_{\mathrm{ASim}}(f_k) = \frac{\sum_{P_j \in S^{t_2}_{p_i} \cap\ P_{f_k}} \mathrm{ASim}_{p_i}(p_j)}{\sum_{P_j \in S^{t_2}_{p_i}} \mathrm{ASim}_{p_i}(p_j)} \tag{7}$$

The recommendation list is sorted according to the weighted popularity of the files $\mathrm{WPop}_{\mathrm{ASim}}(f_k)$ with the files that have a higher weight at the top of the list.

5.4 Symmetric peers' similarity based recommendation (SS)

Here, we define another similarity metric. For every peer p_j in P_{f_x} we define the similarity relationship as:

$$\mathrm{SSim}_{p_i}(p_j) = \frac{|f(p_i) \cap f(p_j)|}{|f(p_i) \cup f(p_j)|} \tag{8}$$

Note that the denominator $|f(p_i) \cup f(p_j)|$ can not be null.

The value of $\mathrm{SSim}_{p_i}(p_j)$ is a numerical score that shows how similar the peer p_j is to the peer p_i. Note that this similarity relationship is symmetric, i.e., $\mathrm{SSim}_{p_i}(p_j) = \mathrm{SSim}_{p_j}(p_i)$

This scheme will choose only peers that have $\mathrm{SSim}_{p_i}(p_j) \geq t_3$. Where t_3 is a threshold.

Let $\mathrm{SS}^{t_3}_{p_i} = \{p_j, p_j \in P_{f_x} \textit{ and } \mathrm{SSim}_{p_i}(p_j) \geq t_3\}$.

5.4.1 Symmetric peers' similarity with file popularity (SSFP)

We apply the *FP* scheme within the set $SS_{p_i}^{t_3}$ of peers most similar to peer p_i. For every file, we compute:

$$\mathrm{Pop}_{\mathrm{SSim}}(f_k) = \frac{|P_{f_k} \cap P_{f_x} \cap \mathrm{SS}_{p_i}^{t_3}|}{|P_{f_x} \cap \mathrm{SS}_{p_i}^{t_3}|} \tag{9}$$

Note that if $t_3 = 0$ then $\mathrm{Pop}_{\mathrm{SSim}}(f_k) = \mathrm{Pop}(f_k)$.

This scheme will recommend only files f_k such that $\mathrm{Pop}_{\mathrm{SSim}}(f_k) \geq t_1$, where t_1 is a threshold. This recommendation list is sorted according to the popularity of the files $\mathrm{Pop}_{\mathrm{SSim}}(f_k)$ with the files that are most popular at the top of the list.

5.4.2 Symmetric peers' similarity with weighted file popularity (SSWFP)

We apply the *Weighted File Popularity* technique within the set $SS_{p_i}^{t_3}$ of peers most similar to peer p_i. For every file, we compute:

$$\mathrm{WPop}_{\mathrm{SSim}}(f_k) = \frac{\sum_{P_j \in SS_{p_i}^{t_3} \cap P_{f_k}} \mathrm{SSim}_{p_i}(p_j)}{\sum_{P_j \in SS_{p_i}^{t_3}} \mathrm{SSim}_{p_i}(p_j)} \tag{10}$$

The recommendation list is sorted according to the weighted popularity of the files $\mathrm{WPop}_{\mathrm{SSim}}(f_k)$ with the files that have a higher weight at the top of the list.

5.5 Similarity metrics and binary ratings

Similarity has been used in data mining, pattern recognition, information retrieval, information theory, data clustering and artificial intelligence.

The most used similarity techniques for recommender systems are the Pearson correlation and the Cosine measure [14, 21, 22]. However, a thorough investigation of similarity metrics based on binary ratings reveals the existence of a number of other potentially better similarity metrics.

Adopting an implicit rating approach, implicates a binary value (i.e., 1 if the peer has the file, 0 otherwise) and hence promotes the use of similarity measures for binary data.

Different similarity metrics have been used in exploratory data analysis [13], and in genetics and molecular biology [5].

We adopt the following notations:

Let p_i be the active peer (i.e., the peer requesting the file).

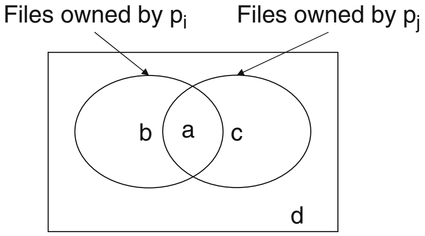
Let p_j be the peer for which we want to compute the similarity with the active peer p_i.

For a particular file f, let C be the observation that the active peer p_i has the file f. And let D be the observation that a peer p_j has this file.

Let a, b, c, and d as follows:

- a: number of times $C = 1$ and $D = 1$. This represents the number of files common to both p_i and p_j.
- b: number of times $C = 1$ and $D = 0$. This represents the number of files owned by p_i but not p_j.

Fig. 3 a, b, c and d



- c: number of times $C = 0$ and $D = 1$. This represents the number of files owned by p_j but not p_i.
- d: number of times $C = 0$ and $D = 0$. This represents the number of files neither owned by p_i nor p_j.

Figure 3 depicts a graphical representation of the considered notations.

The similarity metrics may be grouped into two classes according to how they deal with the negative co-occurrence (i.e., d value)[13]. These are the metrics that use the d value in their equation. Table 1 shows the similarity metrics that consider the negative co-occurrence, while Table 2 shows the similarity metrics that do not consider this co-occurrence. In [11], the similarity metrics in the former table are named type 2 similarity metrics, while those in the latter table are named type 1 similarity metrics.

Each similarity metric has its own characteristics and properties. In this paper, we explore all these similarity metrics by applying them to find the most similar peers in order to make appropriate recommendations. We also investigate both the weighted approach and the non-weighted approach in computing the recommendations. We want to analyze the impact of the similarity metrics on the recommender system. Furthermore, we study these similarity metrics under different scenarios to evaluate their performance and their ability to make accurate recommendations.

It is important to note that since implicit rating is used for files' recommendations, the use of Pearson correlation is not applicable since the average rating given by a peer p to its files is always 1. In this case, the Pearson correlation measure is not well defined.

Table 1 Similarity metrics with negative co-occurrence

Similarity metric	Equation	Scheme number
Rogers and Tanimoto [18]	$\frac{a+d}{a+d+2(b+c)}$	4
Simple Matching [23]	$\frac{a+d}{a+b+c+d}$	5
Ochiai II [16]	$\frac{ad}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$	6
Sokal and Sneath [24]	$\frac{2(a+d)}{2(a+d)+b+c}$	7
Russel and Rao [20]	$\frac{a}{a+b+c+d}$	11

Table 2 Similarity metrics without negative co-occurrence

Similarity metric	Equation	Scheme number
Ochiai I [16]	$\frac{a}{\sqrt{(a+b)(a+c)}}$	1
Jaccard [7]	$\frac{a}{a+b+c}$	2
Anderberg [2]	$\frac{a}{a+2(b+c)}$	8
Czekanowsky–Sorensen–Dice [4]	$\frac{2a}{2a+b+c}$	9
Kulczynski II [10]	$\frac{a}{2}\left(\frac{1}{a+b}+\frac{1}{a+c}\right)$	10

6 Performance evaluation

We have chosen a simulative approach as a first step to evaluate the considered similarity metrics using the proposed recommender framework. Based on the files shared by the peers that have the requested file, recommendations will be provided. If we choose to use real data, we will not be able to test the impact of the search process on the recommendations.

As a future work, we are investigating the considered similarity metrics using the real dataset from movielens (943 users, 1,682 items and 100,000 ratings) that corresponds to movie ratings and are obtained from the movielens research project. This dataset uses explicit rating from 1 to 5. The binary dataset from www.Audioscrobbler.com is no more available and we will not be able to use it.

6.1 Simulated schemes

In this paper, we simulate the following techniques using the non weighted and weighted rating approaches:

- Scheme 3: *Asymmetric Peers' Similarity with File Popularity (ASFP)*. As stated in Eq. 5, this metric uses the following equation: $\frac{a}{a+b}$
- and the following schemes: *Ochiai I* (OcI), *Jaccard* (Jac), *Simple Matching* (SM), *Rogers and Tanimoto* (RT), *Ochiai II* (OcII), *Sokal and Sneath* (SS), *Anderberg* (And), *Czekanowsky–Sorensen–Dice* (CSD), *Kulczynski II* (KII) and *Russel Rao* (RR) presented in Tables 1 and 2.

In the *Cosine measure* technique, the active peer and any other peer are represented by two vectors (generated from the list of files they own) and the similarity between them is measured by computing the cosine of the angle between the two vectors. In Binary rating, the *Cosine measure* and *Ochiai I* are equivalent. We simulate *Ochiai I*.

The *Jaccard* similarity metric is also equivalent to the previously proposed *Symmetric Peers' Similarity*. We simulate the *Jaccard* metric.

The goal from these simulations is to compare the performance of the presented schemes in terms of providing accurate files' recommendations.

6.2 Simulation parameters

The simulation parameters are the following:

- We simulate a system with 1,000 peers and 1,000 files.
- At the beginning of the simulation, each peer has several files and each file has at least one owner.
- 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.
- The percentage of peers in each category is 25% and the percentage of files in each category is 25%.
- Each peer belongs to one category. Peers prefer to have most of the files from their category and only few files from other categories. We investigate the different schemes using different probabilities termed *Initial Profile* (0.5, 0.6, 0.7, 0.8, 0.9, and 1) leading to 6 scenarios. In the case of 0.9 for example, initially, each peer will have 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.
- If no file is recommended, file requests follow the real life distribution observed in [6].
- The threshold for each similarity metric is set to 0.1. This means that the similarity of a peer should be greater than 10% for the peer to be considered.
- We simulate 50,000 requests for each simulation.

Our simulations were implemented using the Peer-to-Peer simulator PeerSim [17]. The simulations were repeated several times for each scheme and for each *Initial Profile* probability. The results presented are the average values. Each scheme has been simulated using the weighted and non weighted rating techniques.

The performance metrics used in the literature are called *Recall* and *Precision* [8, 9]. While *Precision* represents the probability that a recommended item is relevant, *Recall* represents the probability that a relevant item will be recommended. In [9], Recall is measured by taking into account the number of hits. A hit is considered when an item from the top N recommended files is in the test set. Usually, $N = 10$ is the number of items returned to users. The greater is N, the greater is the value of *Recall*.

In some research works [3], both the Recall and the Average Reciprocal Hit-Rank (ARHR) are computed to assess the performance of the recommender systems. The Recall treats all the hits equally regardless of their position in the top N recommended items. In contrast, the ARHR takes into account the position of the hits by giving more weight to the hits that occur in the first positions.

To assess the performance of the considered similarity metrics, we opted to limit the value of N to 1. We consider the fact that the recommended file ranked at the top of the list, will be most probably selected by the active user for download. By increasing the value of N to 10 as it is in the literature, the greater is the chance that a hit will occur. Reducing this value to 1 will make it hard to get a hit. For each scheme, we compute the *Peer Satisfaction*. This value is computed for all peers' categories and it represents the average value of the ratio between the number of recommended files that match peer's category over all the files recommended to the peer. This value is close to *Precision*; however, we are recommending only 1 file which is different and more difficult compared to other performance metrics used in some research works.

The simulations show that even with such a harsh constraint, some of the considered similarity metrics were able to achieve good performance.

6.3 Simulation results

We simulated all the schemes under the same conditions and we compared the performance of these schemes. The simulations were conducted in six different scenarios based on the *Initial Profile* probability.

6.3.1 First scenario

At the beginning of the simulations, peers get files from the category that they prefer with a probability of 1 and no file from other categories is selected. Figure 4 depicts the peers' satisfaction for all the schemes with the non weighted and weighted rating approaches. By comparing the results obtained, there is no significant difference between these approaches. Peers' satisfaction almost reaches 100% for the following schemes: Ochiai I (1), Jaccard (2), ASFP (3), Ochiai II (6), Czekanowsky-Sorensen-Dice (9) and Kulczynski II (10). For the Anderberg (8) scheme, peers' satisfaction is relatively lower (95%). However, this value decreases significantly for the following schemes: Rogers and Tanimoto (4), Simple Matching (5), Sokal and Sneath (7). This peers' satisfaction is settling around 23%. In these schemes, 98% of files that have been downloaded by the peers were recommended to them. The bad performance of these schemes can be explained by the fact that their corresponding similarity metrics

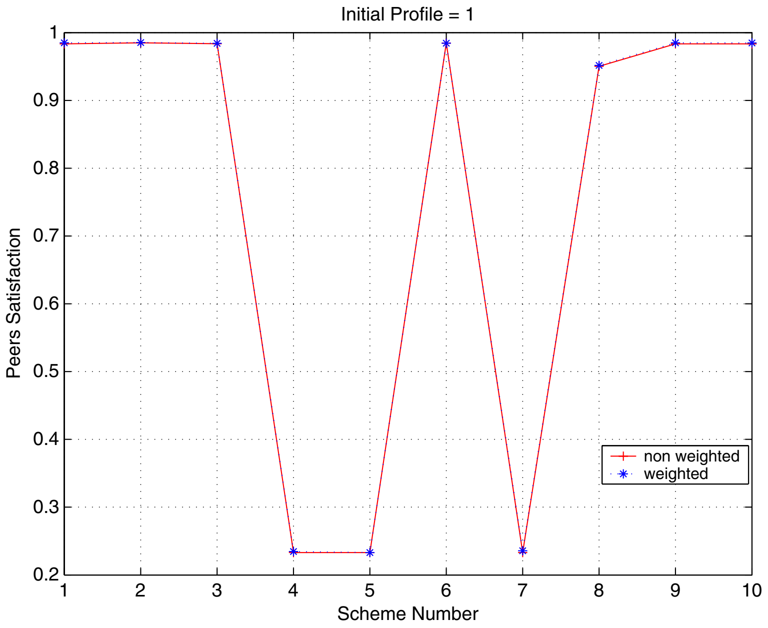


Fig. 4 Peers satisfaction (first scenario)

take into account the negative co-occurrence as explained in Table 1. However, the fact that two peers do not have a specific file, does not mean that they do not like it. Also, it does not mean that they have the same interests.

6.3.2 Second scenario

In this scenario, peers get files from the category that they prefer with a probability of 0.9 and only a probability of 0.1 for files from other categories. Figure 5 depicts the peers' satisfaction for all the schemes. The results are similar for both the weighted and non weighted approaches. The files that are recommended to peers match the peers' preferences for the following schemes: Ochiai I (1), Jaccard (2), ASFP (3), Ochiai II (6), Czekanowsky–Sorensen–Dice (9) and Kulczynski II (10). Peers' satisfaction reaches 98%. A slightly decrease in peers' satisfaction is noticed for the Anderberg (8) scheme (92%). In this scheme, an average of 84% of files downloaded by peers, were recommended to them.

Decreasing the value of *Initial Profile* probability will necessarily decrease peers' satisfaction. This can be explained by the fact that peers have files from several categories, recommender schemes can not easily identify peer's category and recommend files that match its interests.

6.3.3 Third scenario

Peers start with files that match their preferences with a probability of 0.8 and a probability of 0.2 for files from other categories. Figure 6 shows the results obtained

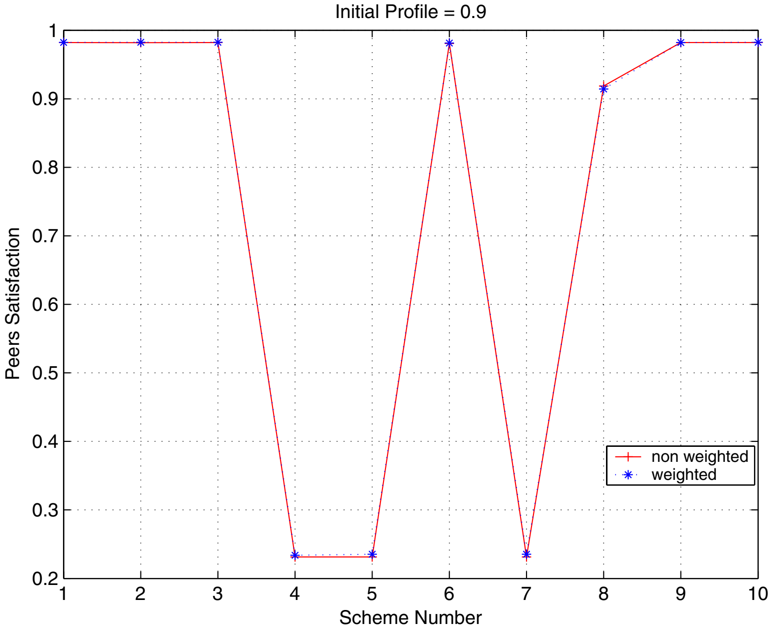


Fig. 5 Peers satisfaction (second scenario)

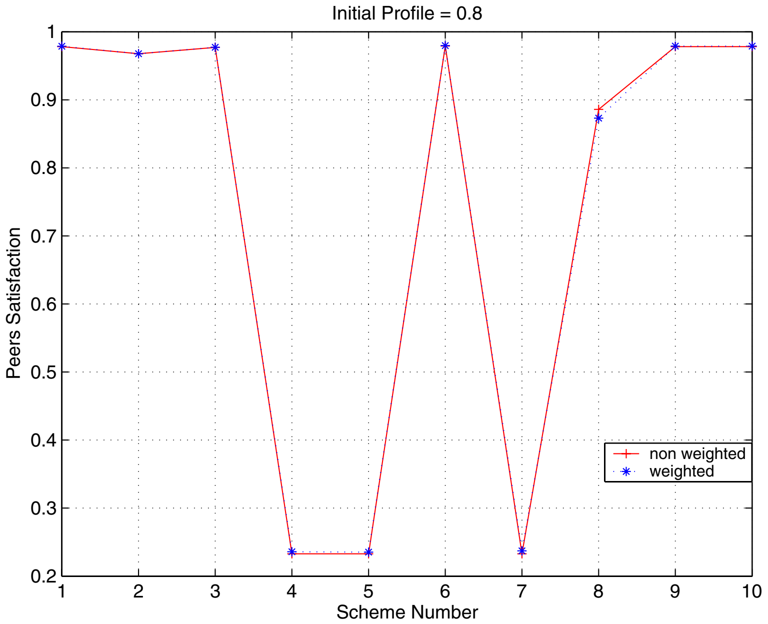


Fig. 6 Peers satisfaction (third scenario)

for all the schemes. In this scenario, peers' satisfaction is almost 98% for the following schemes: Ochiai I (1), Jaccard (2), ASFP (3), Ochiai II (6), Czekanowsky–Sorensen–Dice (9) and Kulczynski II (10). Most of the files that are recommended to peers are of interest to them. The Anderberg scheme (8) is less accurate in making recommendations. In this scheme, peers' satisfaction is 88%. The achieved performance of the other schemes is lower, settling around 23%.

6.3.4 Fourth scenario

To show the effectiveness of the proposed schemes, we performed another set of simulations. In this scenario, peers start with files that match their category with an *Initial Profile* probability equals to 0.7. Figure 7 presents the results. Peers satisfaction is still higher for the following schemes: Ochiai I (1), ASFP (3), Ochiai II (6), Czekanowsky–Sorensen–Dice (9) and Kulczynski II (10) compared to other schemes. Peers' satisfaction is decreased when using the Jaccard scheme (2) to achieve only 91% in the non weighted rating. A significant decrease in the performance of the Anderberg scheme (8) is also noticed in this scenario. The peers' satisfaction is only 82% in the non weighted rating approach which is slightly higher than the weighted rating approach. As mentioned in the previous scenarios, the following schemes: Rogers and Tanimoto (4), Simple Matching (5), Sokal and Sneath (7) do not provide good recommendations to the peers.

A decrease of the *Initial Profile* value to 0.7 will not lead to a significant decrease in *Peers Satisfaction* while using the weighted and non weighted rating approaches.

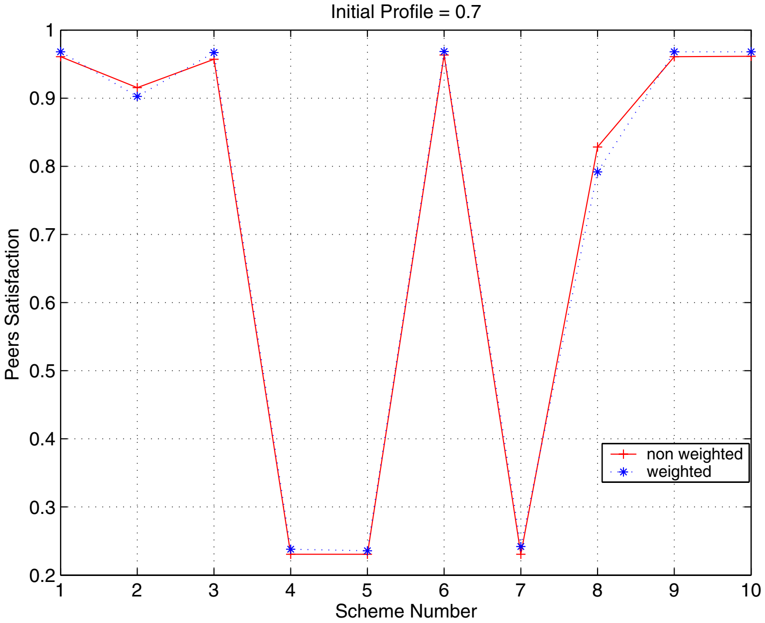


Fig. 7 Peers satisfaction (fourth scenario)

6.3.5 Fifth scenario

Decreasing the value of *Initial Profile* allows to distinguish among the schemes that provide better recommendations to the peers. Figure 8 depicts the peers' satisfaction for all the schemes. The results are not as good as in the previous set of simulations. Peers' satisfaction is approximatively 70% by using the non weighted rating for the following schemes: Ochiai I (1), Jaccard (2), ASFP (3), Czekanowsky–Sorensen–Dice (9) and Kulczynski II (10). However, despite of the low value of *Initial Profile* probability, the peers' satisfaction value is still acceptable. The recommender scheme Ochiai II (6) shows a significant increase in peers' satisfaction compared to the previously mentioned schemes. In the Ochiai II (6) scheme, peers satisfaction achieves a high score equals to 79%. The performance of this scheme in this scenario surpasses all other schemes. The Anderberg scheme (8) is less accurate in making recommendations. As discussed in the previous scenario, the following schemes: Rogers and Tanimoto (4), Simple Matching (5), Sokal and Sneath (7), are the worst schemes in making recommendations.

Figure 8 shows the good performance of the following recommender schemes using the weighted rating approach: Ochiai I (1), ASFP (3), Czekanowsky–Sorensen–Dice (9) and Kulczynski II (10). These schemes surpass the other schemes in providing appropriate and accurate recommendations. Although the value of *Initial Profile* probability is relatively lower, the use of the weighted rating technique allows these schemes to make a good distinction between files' categories and recommend the appropriate files based on peers' profiles. Peers' satisfaction reaches 87% in contrast to 79% in the non weighted approaches. In general, the weighted rating

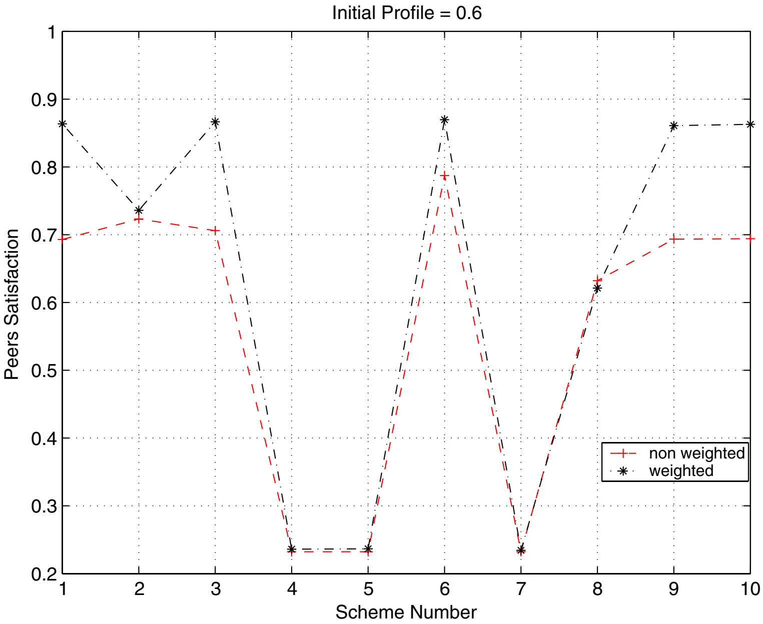


Fig. 8 Peers satisfaction (fifth scenario)

techniques provide better recommendations' accuracy compared to the non weighted rating techniques.

6.3.6 Sixth scenario

Figure 9 shows the results for the *Initial Profile* probability of 0.5. In this case, all schemes achieved lower results than in the other two scenarios. Again, the weighted approaches outperformed the non weighted ones.

6.3.7 The Russel Rao scenario

Using the *Russel Rao* metric under the same conditions does not provide any recommendations. This similarity metric is different from the other ones. The Russel and Rao similarity metric presents results different from the other similarity metrics because it excludes the negative co-occurrences in the numerator and includes it in the denominator. To show the performance of this metric, we conducted a new set of simulations using the same parameters as presented in Section 6.2 with the threshold $t_2 = 0$. This means that we consider all the peers that have the requested file as neighbors. However, the degree of similarity between the active peer and the peers from the neighborhood set can be different from one peer to another. We repeated the simulations 10 times for each of the following *Initial Profile* probability: 0.5, 0.6, 0.7, 0.8, 0.9, 1.

Figure 10 presents the obtained results. The *X* axis represents the *Initial Profile* probability while the *Y* axis represents the peers' satisfaction using the non weighted

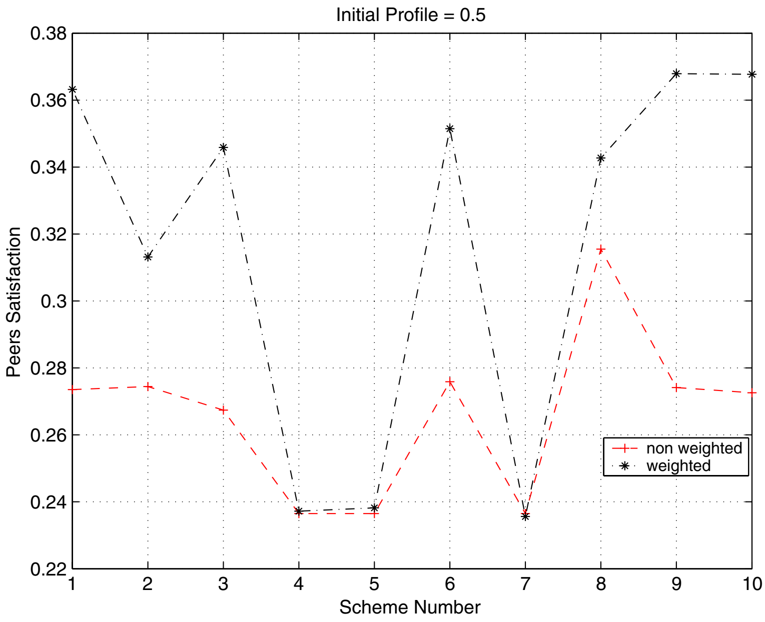


Fig. 9 Peers satisfaction (sixth scenario)

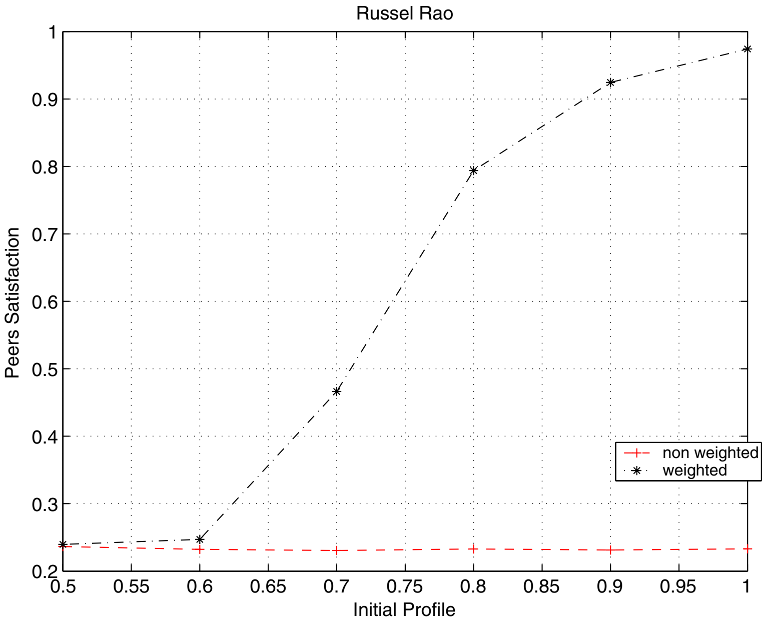


Fig. 10 Peers satisfaction of the Russel Rao

and the weighted approaches. In the former approach, peers satisfaction achieves a low score (23%). In the second approach, peers satisfaction increases according to the value of *Initial Profile*. For example, an *Initial Profile* value of 0.8 will lead to 79% of peers satisfaction. The more precise is the *Initial Profile*, the higher is the peers' satisfaction.

6.4 Analysis of recommender schemes

Similarity metrics aim at quantifying the extent to which objects resemble each other [11]. Similarity metrics make possible to determine if the compared peers can be assigned to the same class or not.

Similarity metrics express the proportion of matches between two peers in a different way. While *Anderberg* and *Rogers and Tanimoto* similarity metrics give twice weight to disagreement, the similarity metrics *Sokal and Sneath* and *Czekanowsky–Sorensen–Dice* give more weight to agreement. Also, the *Sokal and Sneath* metric is similar to the *Simple Matching* metric but gives double weight to matches. Similarly, *Czekanowsky–Sorensen–Dice* metric is similar to the *Jaccard* similarity metric but gives twice the weight to matches. In Table 2, most of the similarity metrics are increasing functions of a and decreasing functions of b and c. Similarity is higher when the compared peers share more common files and have few distinctive files. While, similarity metrics in Table 1 take into consideration the intersection, the differences and also the intersection of the complementary sets of the compared peers. For these metrics, the common files and the absence of same files have the same role. In addition to the common files, the absence of same files increases the similarity between the compared peers. However, the *Russel and Rao* metric is more severe in attesting the resemblance between peers, since the absence of same files is added only in the denominator. In this metric, similarity is based only on the common files owned by the compared peers over all the files.

In many applications such as image retrieval, the user is interested in the list of objects most similar to its request (ordered-based approach) rather than the values of the similarity scores (value-based approach) [11]. The similarity scores are not as important as the order of similar objects. However, in the performed simulations, we were interested to know the order of similar peers to the requester peer (i.e., to whom the recommendation is made) in addition to the similarity scores. The similarity of a peer should be greater than 10% to be considered. We considered both the ordered-

Table 3 Summary of peers' satisfaction

Probability	Oc I	Oc I W	Jaccard	Jaccard W	ASFP	ASWFP	RT	RT W	SM	SM W
1	98.34	98.50	98.49	98.53	98.37	98.42	23.30	23.46	23.30	23.30
0.9	98.22	98.22	98.18	98.25	98.21	98.25	23.13	23.38	23.13	23.53
0.8	97.81	97.88	96.77	96.77	97.69	97.72	23.28	23.60	23.28	23.53
0.7	96.08	96.81	91.54	90.25	95.72	96.69	23.05	23.80	23.05	23.56
0.6	69.30	86.36	72.33	73.59	70.61	86.66	23.22	23.60	23.22	23.66
0.5	27.35	36.32	27.45	31.31	26.74	34.58	23.65	23.72	23.65	23.82

Table 4 Summary of peers' satisfaction (cond.)

Probability	Oc II	Oc II W	SS	SS W	And	And W	CSD	CSD W	K II	K II W
1	98.46	98.44	23.30	23.58	95.03	95.16	98.34	98.50	98.34	98.48
0.9	98.08	98.13	23.13	23.54	91.87	91.44	98.22	98.21	98.22	98.24
0.8	97.90	97.96	23.28	23.71	88.60	87.31	97.81	97.89	97.81	97.90
0.7	96.33	96.84	23.05	24.19	82.84	79.19	96.09	96.82	96.15	96.81
0.6	78.72	86.97	23.22	23.44	63.22	62.13	69.35	86.08	69.42	86.28
0.5	27.59	35.15	23.65	23.56	31.55	34.27	27.41	36.79	27.26	36.77

based and the value-based comparisons in the simulations to assess the performance of the similarity metrics.

Tables 3 and 4 present a summary of the results of the simulations. By comparing all the schemes using the weighted and non weighted rating techniques in terms of peers' satisfaction, we found that the following schemes: Ochiai I (1), ASFP (3), Ochiai II (6), Czekanowsky–Sorensen–Dice (9) and Kulczynski II (10), provide better performance in terms of recommendations' accuracy. The Jaccard (2) and the Anderberg (8) schemes are less accurate. However, a low performance in providing appropriate recommendations is observed for the following schemes: Rogers and Tanimoto (4), Simple Matching (5), and Sokal and Sneath (7). From the Tables 3 and 4, it is also clear that as the initial distribution of files becomes fuzzy, the schemes are not able to clearly find the exact peers' profile and hence will lead to poor peers' satisfaction. Moreover, the weighted rating technique improves the performance of the schemes since the weight of similarity measures is taken into account while computing peers recommendations compared to the non weighted approach. The low performance of the schemes: Rogers and Tanimoto (4), Simple Matching (5), Sokal and Sneath (7) can be explained by the fact that these schemes take into consideration negative co-occurrence as explained in Table 1. If two peers do not have a file, a rating of 0 is assigned to this file. Considering that these two peers are similar if there are files that they both do not have, does not make them necessarily similar. It does not mean that they have the same interests. Indeed, the negative co-occurrence does not mean necessarily any resemblance or similarity in our context. However, an acceptable performance of the Ochiai II (6) scheme has been shown in the simulations although this scheme belongs to this category. On the other hand, similarity coefficients with no negative co-occurrence as described in Table 2, lead to better recommendations' accuracy and higher peers' satisfaction.

Table 5 Summary of percentage of recommended files

Probability	Oc I	Oc I W	Jaccard	Jaccard W	ASFP	ASWFP	RT	RT W	SM	SM W
1	97.61	97.61	97.42	97.42	97.61	97.61	98	98	98	98
0.9	97.72	97.71	97.17	97.16	97.72	97.71	98	98	98	98
0.8	97.75	97.75	96.20	96.22	97.75	97.72	98	98	98	98
0.7	97.75	97.75	93.93	93.91	97.75	97.75	98	98	98	98
0.6	97.72	97.72	91.26	91.19	97.73	97.73	98	98	98	98
0.5	97.68	97.68	89.51	89.39	97.70	97.71	98	98	98	98

Table 6 Summary of percentage of recommended files (cond.)

Probability	Oc II	Oc II W	SS	SS W	And	And W	CSD	CSD W	K II	K II W
1	97.61	97.61	98	98	90.40	90.42	97.61	97.61	97.61	97.61
0.9	97.71	97.71	98	98	84.21	84.11	97.72	97.71	97.72	97.71
0.8	97.71	97.72	98	98	78.89	78.77	97.75	97.75	97.75	97.75
0.7	97.61	97.61	98	98	74.34	74.27	97.75	97.75	97.75	97.75
0.6	97.41	97.42	98	98	70.88	70.70	97.72	97.72	97.72	97.72
0.5	97.14	97.14	98	98	68.88	68.78	97.68	97.68	97.68	97.68

In the performed simulations, we want to investigate other important issues as the *Cold start* and the *Data sparseness* as discussed in Section 2.6. In the ASFP scheme with weighted rating approach and with *Initial Profile* probability equals to 0.8, 97.72% of files downloaded by the peers were recommended. This means that the ASFP scheme does not suffer from the *Cold start* and the *Data sparseness* as explained in Section 5 since recommendations are made to the users. Simulations results also show that the recommendations are provided to the peers as soon as the system starts. Although no explicit rating was provided, the scheme is able to make recommendations and more precisely accurate ones. Tables 5 and 6 present a summary of the results of the simulations. By comparing all the schemes using the weighted and non weighted rating techniques in terms of percentage of recommended files, we found that the following schemes: Ochiai I (1), ASFP (3), Rogers and Tanimoto (4), Simple Matching (5), Sokal and Sneath (7), Ochiai II (6), Czekanowsky–Sorensen–Dice (9) and Kulczynski II (10), provide recommendations with a higher percentage value compared to the Jaccard (2) and the Anderberg (8) schemes.

Another important issue to discuss is the *Trust* problem. Malicious peers that send inauthentic files may also share useless files or provide high ratings for irrelevant files to impact badly files' recommendations. To solve this problem, we suggest to make recommendations based on files shared by the reputable peers. The recommender system can choose files only from trustworthy provider peers that have the requested file. These peers may be selected based on a reputation value. This will reduce the impact of malicious peers on the recommendations.

7 Concluding remarks

In this paper, we proposed a novel recommender framework for partially decentralized file sharing P2P systems. We investigated similarity metrics, that were proposed in other fields, and adapted them to file sharing P2P systems. We analyzed the impact of each similarity metric on the accuracy of the recommendations. Both weighted and non weighted approaches were investigated. In general, the weighted approaches achieve higher recommendation accuracy. Within the weighted approaches, similarity metrics that do not consider negative co-occurrence lead to better recommendation performance. Files' recommendations will, on one hand, increase users' satisfaction since they will receive recommendations on files that they prefer. On the other hand, they will help peers stay connected to the system to serve other peers in addition to increasing the peers' loyalty to the system.

Acknowledgements This work was supported in part by the Natural Science and Engineering Council of Canada (NSERC) under its Discovery program, and the WCU (World Class University) program through the Korea Science and Engineering Foundation funded by the Ministry of Education, Science and Technology (Project No. R31-2008-000-10100-0).

References

1. Adar E, Huberman B (2000) Free riding on gnutella. Tech. rep., HP
2. Anderberg M (1973) Cluster analysis for applications. Academic Press
3. Deshpande M, Karypis G (2004) Item-based top-N recommendation algorithms. ACM Trans Inf Sys 22(1):143–177
4. Dice L (1945) Measures of the amount of ecological association between species. Ecology 26: 297–302
5. Duarte J, dos Santos J, Melo L (1999) Comparison of similarity coefficients based on RAPD markers in the common bean. Genet Mol Biol 22(3):427–432
6. Gummadi K, Dunn RJ, Saroiu S, Gribble SD, Levy HM, Zahorjan J (2003) Measurement, modeling, and analysis of a Peer-to-Peer file sharing workload. In: ACM symposium on operating systems principles, pp 314–329
7. Jaccard P (1901) Etude Comparative de la Distribuition Florale dans une Portion des Alpes et de Jura. Bull Soc Vaud Sci Nat 37:547–579
8. Jamali M, Ester M (2009) Using a trust network to improve top-N recommendation. In: ACM conference on recommender systems, pp 181–188
9. Karypis G (2001) Evaluation of item-based top-N recommendation algorithms. In: International conference on information and knowledge management, pp 247–254
10. Kulczynski S (1927) Classe des sciences mathématiques et naturelles. l'Acadamie Polonaise des Sciences et des Lettres II:57–203
11. Lesot M, Rifqi M, Benhadda H (2009) Similarity measures for binary numerical data: a survey. IJKESDP 1(1):63–84
12. Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. In: IEEE internet computing, pp 76–80
13. Lourenço F, Lobo V, Baçao F (2004) Binary-based similarity measures for categorical data and their application in self-organizing maps. In: JOCLAD
14. Massa P, Avesani P (2004) Trust-aware collaborative filtering for recommender systems. In: International conference on cooperative information systems
15. Mekouar L, Iraqi Y, Boutaba R (2009) A contribution-based service differentiation scheme for Peer-to-Peer systems. Int J Peer-to-Peer Netw Appl 2(2):146–163
16. Ochiai A (1957) Zoogeographic studies on the soleoid fishes. Japanese Society for Fish Science 22:526–530
17. PeerSim (2003). http://peersim.sourceforge.net/
18. Rogers J, Tanimoto T (1960) A computer program for classifying plants. Science 132:1115–1118
19. Ruffo G, Schifanella R, Ghiringhello E (2006) A decentralized recommendation system based on self-organizing partnerships. In: IFIP networking, pp 618–629
20. Russel P, Rao T (1940) On habitat and association of species of anopheline larvae in South-Eastern Madras, vol 3, pp 153–178
21. Sarwar B, Karypis G, Konstan J, Reidl J (2000) Analysis of recommendation algorithms for e-commerce. In: EC, pp 158–167
22. Sarwar B, Karypis G, Konstan J, Reidl J (2001) Item-based collaborative filtering recommendation algorithms. In: International conference on world wide web, pp 285–295
23. Sokal R, Michener C (1958) A statistical method for evaluating systematic relationships. Society of University of Kansas 38:1409–1438
24. Sokal R, Sneath P (1963) Principles of numeric taxonomy. W.H. Freeman

25. Wang J, Pouwelse J, Lagendijk RL, Reinders MJT (2006) Distributive collaborative filtering for Peer-to-Peer file sharing systems. In: ACM symposium on applied computing, pp 1026–1030
26. Wang J, Pouwelse JA, Fokker J, de Vries A, Reinders M (2008) Personalization on a Peer-to-Peer television system. Multimedia Tools Appl 36:89–113

Loubna Mekouar received her M.Sc. degree from the University of Montreal in 1999 and Ph.D. in computer science from the University of Waterloo, Canada in 2010. Her research interests include trust and reputation in Peer-to-Peer systems, Quality of Service in multimedia applications, network and distributed systems management, network virtualization and Web Services. Loubna Mekouar has received many scholarships of excellence during her studies.

Youssef Iraqi received his B.Sc. in Computer Engineering, with high honors, from Mohammed V University, Morocco, in 1995. He received his M.Sc. and Ph.D. degrees in Computer Science from the University of Montreal in 2000 and 2003 respectively. From 1996 to 1998, he was a research assistant at the Computer Science Research Institute of Montreal, Canada. From 2003 to 2005, he was a research assistant professor at the David R. Cheriton School of Computer Science at the University of Waterloo. He is currently an assistant professor at Khalifa University, Sharjah, UAE. His research interests include network and distributed systems management, resource management in multimedia wired and wireless networks, and Peer-to-Peer networking.

Raouf Boutaba received the M.Sc. and Ph.D. Degrees in Computer Science from the University Pierre & Marie Curie, Paris, in 1990 and 1994 respectively. He is currently a Professor of Computer Science at the University of Waterloo. His research interests include network, resource and service management in wired and wireless networks. Dr. Boutaba is the founder and Editor-in-Chief of the IEEE Transactions on Network and Service Management and on the editorial boards of several other journals. He is currently a distinguished lecturer of the IEEE Communications Society, the chairman of the IEEE Technical Committee on Information Infrastructure. He has received several best paper awards and other recognitions such as the premier's research excellence award.