

# P2PCF: A collaborative filtering based recommender system for peer to peer social networks

Lyes Badis <sup>a,\*</sup>, Mourad Amad <sup>a,b</sup>, Djamil Aïssani <sup>a</sup> and Sofiane Abbar <sup>c</sup>

<sup>a</sup> *LaMOS Research Unit, Computer Sciences Department, Faculty of Exact Sciences, Bejaia University, Algeria*  
E-mails: [badis.lyes@yahoo.fr](mailto:badis.lyes@yahoo.fr), [djamil\\_aissani@hotmail.com](mailto:djamil_aissani@hotmail.com)

<sup>b</sup> *Computer Sciences Department, Faculty of Sciences and Applied Sciences, Bouira University, Algeria*  
E-mail: [amad.mourad@gmail.com](mailto:amad.mourad@gmail.com)

<sup>c</sup> *Qatar Computing Research Institute (QCRI), Hamad bin Khalifa University, Doha, Qatar*  
E-mail: [sabbar@hbku.edu.qa](mailto:sabbar@hbku.edu.qa)

**Abstract.** The recent privacy incidents reported in major media about global social networks raised real public concerns about centralized architectures. P2P social networks constitute an interesting paradigm to give back users control over their data and relations. While basic social network functionalities such as commenting, following, sharing, and publishing content are widely available, more advanced features related to information retrieval and recommendation are still challenging. This is due to the absence of a central server that has a complete view of the network. In this paper, we propose a new recommender system called P2PCF. We use collaborative filtering approach to recommend content in P2P social networks. P2PCF enables privacy preserving and tackles the cold start problem for both users and content. Our proposed approach assumes that the rating matrix is distributed within peers, in such a way that each peer only sees interactions made by her friends on her timeline. Recommendations are then computed locally within each peer before they are sent back to the requester. Our evaluations prove the effectiveness of our proposal compared to a centralized scheme in terms of recall and coverage.

Keywords: Decentralized social network, peer to peer, recommender system, collaborative filtering, cold start

## 1. Introduction

Nowadays, recommender systems constitute an important part in different aspects of our lives. The news we read [67], the movies and TV shows we watch on Netflix [46], the groceries and shopping we request on Amazon [32], the music we listen on Spotify [39], the insurance and financial products we subscribe to [30], the house we rent [49], to name but few, all these choices we make are somewhat influenced by recommendations that are generated on our behalf using the precious data we voluntarily contribute and generate on different online services. However, recent scandals [50] that the world witnessed several times over the last decade regarding social media platforms such as Facebook [50], YouTube [1], and Twitter [2] have raised serious privacy concerns about the way public and private users' data are handled by few centralized systems [43]. While many of these companies strive to adapt to new challenges related to preserving users' data, their business models which rely mainly on advertisement, makes it difficult for them not to expose and exploit user generated content, which constitutes one of the most important back-doors for privacy violations.

Decentralized online social networks (DOSN) constitute an interesting paradigm to tackle the aforementioned privacy issues, by giving back users control over their data. Examples of such systems include Safebook [54], DECENT [33], Mastodon [66], and Confidant [37] which can be categorized into three families: (i) Pure P2P social

---

\*Corresponding author. E-mail: [badis.lyes@yahoo.fr](mailto:badis.lyes@yahoo.fr).

networks, (ii) Federated social networks, and (iii) Hybrid social networks [45]. Unlike the large social networks which log all user interactions into a large centralized matrix of ratings, in which rows are users, columns are content/items, and each cell reports the rating strength of a particular user on a particular item, DOSN forbid such centralization and pushes for the matrix to be distributed among user nodes, in a peer-to-peer (P2P) fashion. This is the main reason why DOSN systems lack several attractive functionalities such as reasonable search engines and content recommendations which are responsible for user acquisition and growth [45]. Indeed, these systems rarely go beyond the simple functionalities of posting, commenting, and sharing.

In this article, we aim at bridging this gap and propose a novel framework for decentralized recommender systems named P2PCF. Our objective is to generate reasonable recommendations for users without compromising their privacy, i.e., without having access to the entire interaction logs generated in the system by all users. P2PCF generates recommendations in three steps: First, a user selects a subset of relevant friends from which to request recommendations. Second, each friend would exploit her local knowledge about the requestor's preferences to produce a list of recommendations that are sent back to the requestor. Finally, the user who initialized the request, aggregates all recommendations based on different parameters related to the content received and the users who shared it.

The main challenge encountered in decentralized recommender systems is how to compute meaningful recommendations from partial data. This problem has limited the scope of possible recommendation techniques that one can use. Indeed, in the literature, recommender systems are categorized into three main classes [62]: content-based recommendation, collaborative filtering, and hybrid systems. However, collaborative filtering based recommender systems are the most widely used as they achieved greater success. We distinguish two types of collaborative filtering approaches: (i) memory-based techniques which keep user interactions with items in a rating matrix, then predict missing ratings using k-nearest neighbors approach. These techniques are quite efficient in terms of run time. (ii) model-based techniques which have been introduced more recently and mostly rely on machine learning techniques to predict user ratings. Examples of such techniques include matrix factorization [60], latent semantic analysis [31], and deep neural networks [22]. However, despite the recent success achieved by applying machine learning techniques to different areas of research such as transport [3], medicine [11,12], and cyber security [53], it is widely accepted that most of this success comes from supervised learning which took advantage of the recent availability of large amounts of labeled data. While this suits perfectly centralized recommender systems, it is very hard indeed to envision a similar level of success in the context of decentralized systems, where only small fractions of data are locally available to the machine learning. Thus, we decided to adapt memory-based collaborative filtering in our P2PCF framework, which is less data-intensive.

The research questions we aim at answering throughout this paper include the following: RQ1. Is it possible to generate recommendations in a decentralized system that are similar to those generated in a centralized system? If not, what is the impact of decentralization (i.e., privacy preserving) on the quality of recommendations users see? RQ2. Is it possible to select a limited set of friends from which to request recommendations? If so, how many and how to select them? This is particularly important to reduce network traffic in P2P systems. RQ3. How can we aggregate and sort sets of recommendations generated by different friends before displaying them on the user timeline?

Our evaluation is conducted on a real data set from Twitter, comprising 13,158 tweets (publications) authored by 5,714 unique users. These tweets received 60K retweets (reactions) from 40k users. Experiments have shown the effectiveness of our proposal in terms of recall (proportion of recovered true positives) and coverage (number of tweets that have received at least one reaction). Indeed, our results show an increase of 50% in the recall and 23% in the coverage level of P2PCF compared to the baseline centralized system, when all the friends contribute to the recommendation process. We also studied the effect of the number of interfering friends and the number of candidate publications per user on the quality of recommendations. We found involving 50% of friends only allows P2PCF overtake the centralized recommender system.

A summary of our main contributions is listed below:

- We introduce a new privacy preserving paradigm for collaborative filtering recommendation in P2P social networks.

- We show that relevant recommendations can be effectively compiled from peers, when the right similarity metrics are used.
- We propose a solution to deal with the inherent cold-start problems for both new users and publications.

The rest of this paper is structured as follows. Section 2 provides background and related works. Section 3 gives a detailed description of our proposal. Section 4 reports our evaluation results. Section 5 concludes the paper and outlines some future directions.

## 2. Background and related works

### 2.1. Recommender systems

Recommender systems are techniques used to propose and recommend relevant content to users. The recommended content can be a movie, a song, a book, a product on an e-commerce platform or a publication in a social network. The recommender system explores the previous opinions (*rating*) of users about a given content to better predict their future interactions (rating) to the content and suggest the most appropriate one. There are three types of recommender systems: (i) content based, (ii) collaborative filtering (CF), and (iii) hybrid approaches [57].

In Content based recommender systems, the content of rated items is analyzed to construct user profiles and preferences. Then, new items are compared to these preferences to recommend the most interesting ones. Content-based filtering systems usually generate recommendations based on the pre-constructed user profiles by measuring the similarity of content to these profiles using some vector based metrics, such as cosine similarity [62]. This approach was applied in several works especially those that aim to suggest text content [29,59].

In collaborative filtering, we use historical ratings to predict the user rating on a new item. The main idea of collaborative filtering based systems is that: if users have agreed with each other in the past (same items, same ratings) then they are most likely to agree on unseen items in the future [57]. Recently, collaborative filtering-based approaches became the most practical and successful methods in recommender systems, evidenced by a number of commercial products or systems [62], including Netflix, Spotify, and Youtube. This development, led to the distinction of two types of collaborative filtering systems: (i) memory based collaborative filtering and (ii) model based collaborative filtering. In memory based collaborative filtering, all the ratings are stored in a matrix (*Users / Items*). Each row corresponds to a user and each column corresponds to an item. Cells are user/item ratings. When we want to recommend a new item to a user, we predict her rating. This prediction is based on the ratings given by this user to similar items (*item based CF*) or on the rating given to the new item by similar users (*users based CF*). Figure 1 illustrates the memory based CF. In a model based collaborative filtering, the idea consists in deriving a model from the historical rating data. To derive the hidden model, a variety of statistical machine learning algorithms are employed on the training database, such as Bayesian networks, neural networks, clustering, and latent semantic analysis to name but a few [62].

The third class of recommender systems is hybrid, in which several techniques are combined [40,44,52,63].

More recently, some recommender systems started using social relations as an additional factor in the recommendation process. These are called social recommender systems. Social relations can be trust relations, friendships, memberships or following relations [57]. This definition was extended to include “*any recommender system that targets social media domains*” [26]. The new definition covers systems recommending any objects in social media domains such as items, tags, links, people, and communities [17,57]. It has been shown that social considerations improve recommender systems since there is a significant overlap between users’ interest and connections [48,56,61].

### 2.2. Content propagation in P2P social networks

In P2P social networks, data are stored on users’ devices [5,35] or replicated in trusted devices [54]. The main problem noticed in these schemes is data availability which strongly depend on devices connection status [4,14,51].

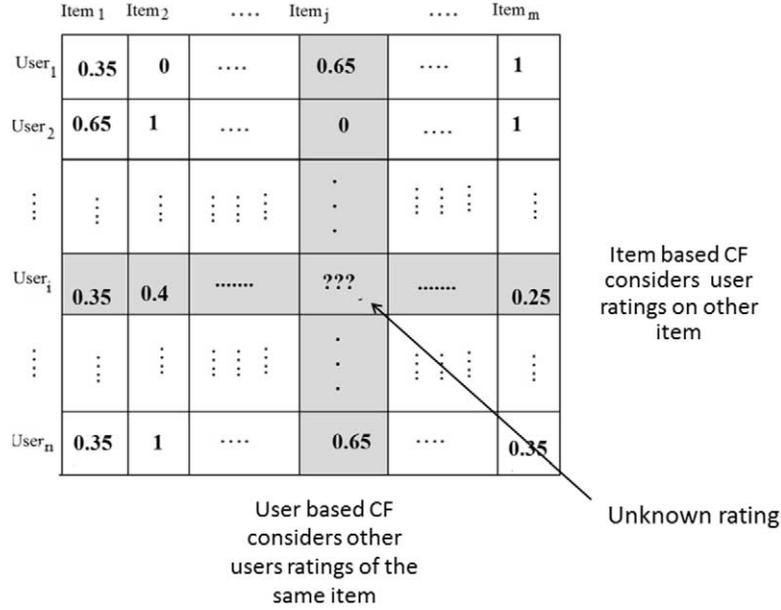


Fig. 1. Memory based collaborative filtering.

Another problem in P2P based systems is data discovery as it is very difficult to get an overall view of all resources in the P2P network [27]. The techniques of data propagation in P2P social networks are categorized into two main classes: (i) active dissemination [38], and (ii) request-reply [25].

Active dissemination means that a user who produces updates pushes them to the network. In [38], authors propose a rumor mongering protocol where users periodically send content to their neighbors. This protocol is improved by introducing heuristic techniques to select the target neighbors [16]. A Community based protocol is proposed in [8]. It relies on a central index to detect communities. Updates are sent to one neighbor from each community. Then, updates are spread within communities. GoDisco [13] exploits semantic context to select the friends concerned by each update message. This method requires that users inform their neighbors about their interests as well. Finally, in [55], users' interactions are used to model the strength of relationships in a given area of interest.

In Request-Reply diffusion, a user looking for updates should send requests to find her friends' profiles. The network overlay (*basically DHT overlay*) is responsible to lead this request to the node storing the profile (*user device or active replicas*). Several systems use this protocol, including: PeerSon [5], LifeSocial.KOM [24], DECENT [33], and SOUP [36]. The authors in [25] consider Cachet [42] as a hybrid technique which introduces the social caching to improve the performances of DECENT. In DECENT [33], a lookup algorithm is launched through the DHT overlay to collect all available updates. Then, a timeline is created and presented to the user. In Cachet, a gossip-based social caching algorithm is also exploited: when an available user is prompted for updates, she returns her proper updates and other cached updates from friends.

In our work, we adopt the request-reply mechanism for our P2P collaborative filtering recommender system. We model user clients in a way that they would reply with a specific content to each request they receive. Furthermore, users' interactions are locally stored and used to recommend relevant updates.

### 2.3. Recommendation in P2P networks

Authors in [20] propose P2PRec, a hybrid recommender algorithm to improve information retrieval in P2P file sharing system combining collaborative filtering and content based filtering. The proposed system extracts topics

from documents and use them to model users' expertise. When a request is made, its topics are first inferred, then the request is only routed to the users qualified as experts in those topics. In F2Frec [19], the social relations are considered to improve recall, trust, and the results confidence. The most relevant friends are selected to recommend documents. In this system, the distance between users is computed by combining the two elements: friendship networks and topics of interest. Both systems, F2Frec and P2PRec, are an enhancement of the classical flooding algorithms [19]. However, they both need additional computations to extract topics from requests and documents. This process is more complicated on social network posts (*short text and abbreviations*).

In [15], authors propose a decentralized recommender system for mobile devices. Here, users' rating is broadcasted to their neighbors' (*distance point of view*). Every user stores and manages the rating of other users on her device and explores her local database to predict ratings. Thus, no central server is needed. The system is less expensive because it is based on direct communication, but has no historical depth: only the ratings noted in the same period are caught. Besides, an implementation is applied to recommend items to see in a museum. In [34], multimedia file exchange is enhanced by introducing the concept of community of interest. Starting from a flooding routing, the peers are classified into communities, according to the semantic of their shared files; the requests are then routed to relevant communities only. The challenge of this system is the decentralized composition of communities when the number of users is large. Furthermore, the systems must track the changing interests of users which requires rebuilding communities. Authors in [28] propose a decentralized recommender system to cope with scalability problems. Their solution is to partition the rating matrix and store its different fragments in a DHT based P2P network. This system avoids overloading a central server but the user privacy is not guaranteed since their votes are copied in clear in the P2P network.

To recommend products in a mobile commerce platform, another P2P recommender system is proposed in [58]. The author proposes to implement collaborative filtering using Gnutella (*a flooding based algorithm*). Users/product rating is stored as a vector in each peer and the flooding algorithm is used to collect similar vectors. Each peer that receives the request vector evaluates the similarity between the request and the cached vectors before similar vectors are sent back. The user looking for recommendations collects similar vectors and applies the collaborative filtering algorithm to them. This approach is a good adaptation of the collaborative filtering concept for P2P systems, however, it inherits many of the drawbacks of flooding algorithms such as being non-deterministic and network-data intensive. In addition, cached message may contain expired data.

The flooding of rating vectors is also used in [65]. Users store their rating vectors and the request is also a rating vector. After spreading the request in the network, the recommended items are sent back in the same path. Every peer should collect and aggregate item ratings before sending them to her neighbor. This approach can be criticized because of considerable waiting time in every node before returning the recommended item to her neighbor.

As a P2P implementations of model based collaborative filtering, decentralized matrix factorization approaches are proposed in [21] and [6]. This method pushes the computation of the recommendation model to the users devices. In addition, the peers have not to share all the rating stored in their local matrix.

### 3. Proposed solution

#### 3.1. General overview

In this section, we depict the general architecture of our solution P2PCF – *a new recommender system for P2P social networks*. When a user joins the network, P2PCF is launched to select the most interesting publications posted by her friends. The collected publications are used to create a timeline (a.k.a. newsfeed) similar to the ones we see in traditional social networks such as Facebook and Twitter. For this, P2PCF uses an adapted version of collaborative filtering, the most popular technique in the field of social recommender systems.

The first challenge is the absence of a central entity where to store the user/publications ratings. In addition, P2P social networks are based on strict privacy constraints.

To solve these challenges, we first propose to store the ratings in a decentralized scheme in which each user stores a local matrix of ratings or interactions about her content. Then, the recommendation process is defined as a three-steps process:

- **Friends selection:** executed on the requester side (*the user for whom we generate recommendations*) to select a set of her friends who would contribute to the recommendations.
- **Publications selection:** to be executed on the friends' side. Each selected friend responds with a set of publications deemed relevant to the requester interests. This selection is based on the local matrix of rating stored within each friend, i.e., every friend evaluates the relevance of items/publications according to the previous ratings given by the requester.
- **Recommendation aggregation:** this step is responsible of the fusion of all the recommendations received from different friends. While each friend responds with a list of publications and their relevance score, computed from their local rating matrix, P2PCF needs to homogenise these scores by integrating the concept of friendship strength, in order to pick the publications that would form the user timeline.

### 3.2. Illustration scenario

P2PCF is executed in each peer without any central entity. The recommendation process is divided into several steps:

- User  $u$  selects a subset of friends to contribute in the recommendation.
- User  $u$  sends separated requests to the selected friends.
- Each friend  $v$  that receives the request applies locally collaborative filtering to select a set of publications to recommend.
- The selected publications are then sent back to user  $u$ .
- User  $u$  collects the friends' responses, aggregates them, and creates a timeline.

Figure 2 illustrates the workflow of this recommendation process.

In the following subsections, we will detail the different components of P2PCF. We start by explaining how data is stored in the distributed networks. Then, we dive into the recommendation process. Finally, we discuss how cold-start problem is handled.

### 3.3. Decentralized storage of interactions

The core challenge in our work is to produce recommendations on a rating matrix that is distributed among peers. In this case, each peer has access to a matrix that reports ratings made by her friends on her own publications only. Table 1 shows an example of such a matrix. In other words, the interactions that a user John makes are stored within his friends distributed storage in a way that none of his friends has access to all of John's interactions and preferences. It is important to emphasize on the decentralized nature of local rating matrices as no central entity has access to the whole data generated in the social network, which constitute a major difference with standard recommendations systems implemented in widely used social networks. In fact, our approach expects each user to store reactions taking place on his timeline, which increases significantly the privacy of users.

The ratings in this matrix represent an aggregate score of different actions that users perform on each publication. These actions include: *share*, *comment*, and *like*. To comply with CF scheme adopted in this research, we propose to translate user reactions (*share*, *comment*, *like*) into numerical ratings.

A common practice in social networks is to assign different weights to different actions based on the level of engagement they induce [47]. For instance, it is widely accepted that sharing a publication is more engaging than posting a comment on it, which is again more engaging than liking it [47].

Our distributed and privacy preserving storage scheme, coupled with the methodology of representing and converting user interactions into numerical values will make it easier for us to generate meaningful recommendations, as we will see in the next subsection.

### 3.4. P2PCF process

As mentioned earlier, once the interactions are captured within different peers as local rating matrices, one can exploit the P2P network of friends to generate recommendations as follows.

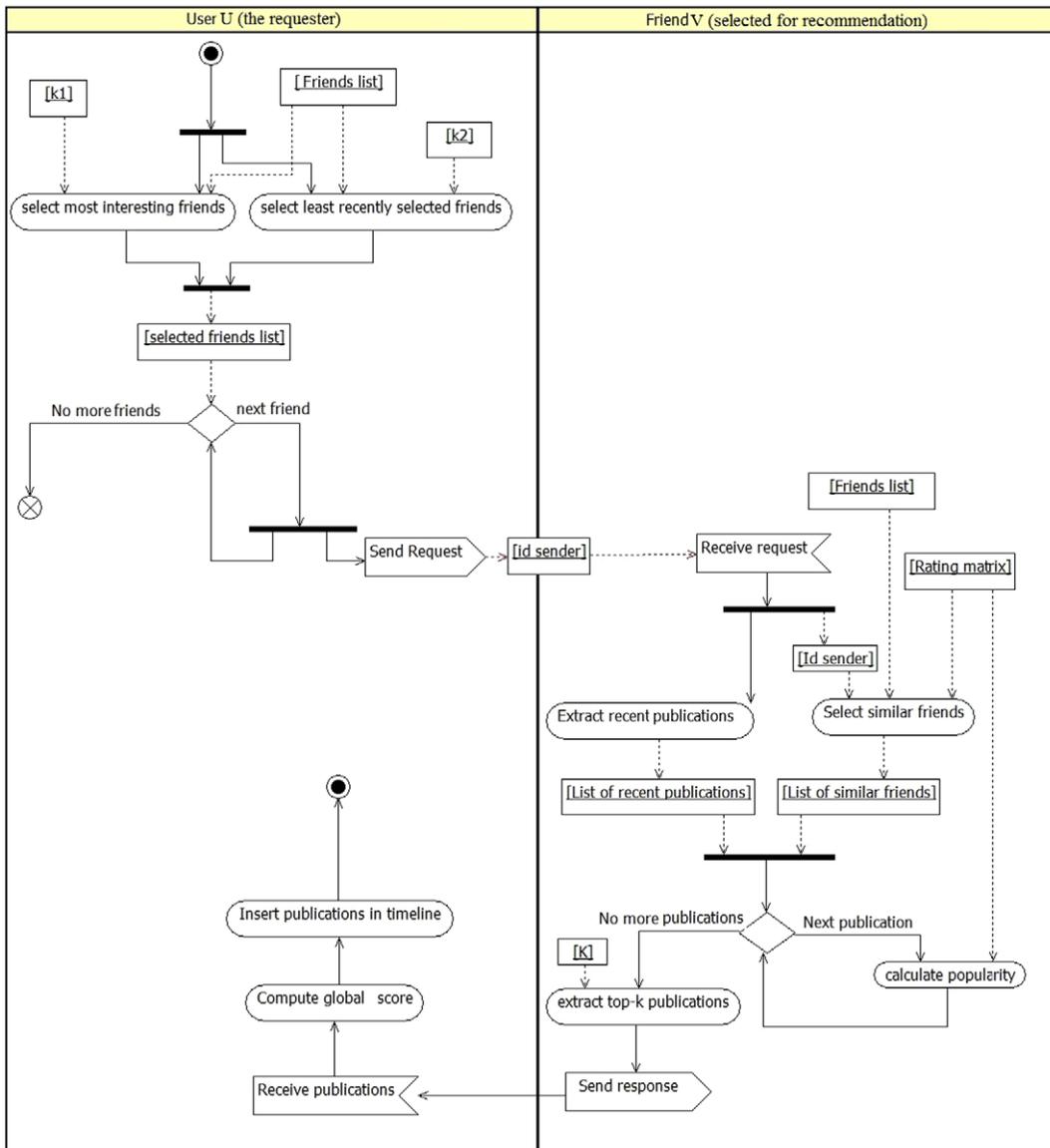


Fig. 2. The workflow of the recommendation process.

Table 1  
Example of a local rating matrix

Friends	Publications					
	P1	P2	P3	P4	P5	P5
John	0.65	0	0	0.30	0	1
Sarah	0	0	1	0.66	0.35	0
Brahim	0.75	0.4	0	1	0.65	0

### 3.4.1. Friends selection

The first step in our recommendation process is the friend selection, which takes place on the requester side. Existing distributed social networks such as Decent [33] and Cachet [42] load the timeline of users with a stream of publications generated by all her friends, which leads to the recommendation of so many unimportant items. Indeed, it is well established that users of social networks do not give the same interest to the content published by all their friends [9,55,64].

Thus, P2PCF builds on these findings to not send the recommendation request to all friends. The idea is to assume that if a user  $u$  expressed interest in the content of a user  $v$  in the past, there is a high expectation that  $u$  will be interested in the content of  $v$  in the future. The user reactions to a given content can be a good indication about their level of interest. These reactions should be recorded to value friends who share interesting content. Thus, we propose to create a counter for each friend: **Reaction\_count**. This counter is incremented each time a user reacts to a publication of a given friend. This is particularly important, as the actual reactions of the user on her friends' publications are stored on the friends' side, which makes it hard to the system to retrieve the counts if no counters are used.

Given that users' interests can change over time, the recommender system should consider the actual interests (*observed in the recent reactions*) more than the archived reactions. Our solution for this problem, is to record the timestamp of the last reaction the user has made on each of her friends' timeline: *Last\_reaction*. This allows the system to capture emerging friends who have lower counts compared to others.

For each friend, we compute the level of interest as the **Reaction\_count** decayed according to the interval between the **Last\_reaction** and the current time. We propose formula (1) to compute this level of interest.

$$Interest\_level = Reaction\_count \times e^{-\alpha \times Delta(t)} \quad (1)$$

Where  $delta(t)$  is the time in seconds since last reaction,  $Reaction\_count(u, v)$  is the accumulation of interactions made by user  $u$  on publications of  $v$ .  $\alpha$  is an adjustable parameter.

The friends are then sorted according to their *Interest\_level*. Friends with the highest *Interest\_level* (*top-k1*) are selected to be contacted for content recommendations.

This strategy feeds the users timeline with recent publications curated by their best friends. However, this can lead to the “*filter bubble*” problem [41], in which the user sees the publications of the same subset of friends. To avoid this situation, we should give a chance to all the friends in order to contribute in the recommendation process, which will eventually increase the diversity and coverage of the recommendations. For this, we propose to consolidate the list of the selected friends by other candidates chosen according to their selection history. That is, for each friend, we store the date of the last time she was selected/solicited for recommendations. Friends in this list are sorted from least recently solicited to most recently solicited, and *top-k2* users on this list are added to the initial list of users.

As shown in Figure 2 and Figure 3, the list of the selected friends is the fusion of two lists: ***top-k1 most interesting friends and top-k2 least recently selected friends***.

### 3.4.2. Publications selection

In this section, we discuss how to select the most relevant publications from each friend. We assume that user  $v$  has been selected in the previous phase, and received a request for a recommendation from a user  $u$ . In this case,  $v$  should respond with his publications that were well rated by users similar to  $u$ . This process is the essence of collaborative filtering. The similarity between users in  $v$ 's rating matrix and  $u$  is calculated in a local way. Indeed, two users  $u_1$  and  $u_2$  may be very similar within  $v_1$ 's local rating matrix, whereas they may turn to be very dissimilar within  $v_2$ 's local rating matrix. This is for instance the case where two users share for example the same love for music genres; hence they are similar within their common music expert friend, whereas they are fans of different sport clubs, which is reflected by a weak similarity within the rating matrix of their sport expert friend. This is a very interesting feature that is not captured in centralized collaborative filtering schemes.

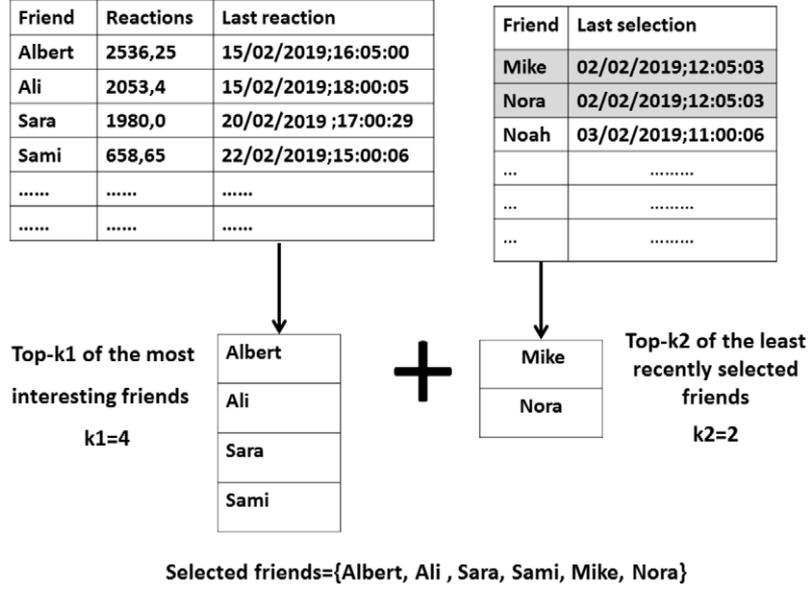


Fig. 3. Friend's selection strategy.

To compute the similarity between users, we adopt the Cosine Similarity [7] formula (2):

$$Sim(u, f) = \frac{\sum_{p \in RRp} Rating(u, p) \times Rating(f, p)}{\sqrt{\sum_{p \in RRp} Rating(u, p)^2} \times \sqrt{\sum_{p \in RRp} Rating(f, p)^2}} \quad (2)$$

Where  $u$  is the requester,  $f$  is a friend of the user  $v$  receiving a request,  $RRp$  is the set of the publications recently rated by the requester  $u$  on the timeline of the user  $v$ .

After receiving a request, the above formula is used to calculate the similarity between the requesting user  $u$  and all the friends connected to the receiver  $v$ . We consider the users with a similarity greater than a predefined threshold ( $MinSim$ ); thus, we create  $SF$ : the set of the most similar friends.

$$SF(u) = \{f \in friends(v) \text{ s.t. } sim(u, f) > MinSim\}$$

The next step is to select the relevant subset of publications produced by  $v$ , according to the ratings provided by the set of similar friends  $SF$ . To this end, we introduce the popularity of publications that we define as a weighted average of the ratings provided by each user in the set  $SF$ , and the similarity of this user to the requester  $u$ . The popularity score is computed following the equation (3):

$$PopSim(u, p) = \frac{\sum_{f \in SF(u)} Sim(u, f) \times Rating(f, p)}{\sum_{f \in SF(u)} Sim(u, f)} \quad (3)$$

Where  $PopSim(u, p)$  is the popularity of the publication  $p$  according to the similar friends to the requester  $u$ ,  $Sim(u, f)$  is the similarity level between the requester  $u$  and a friend  $f$  (formula (2)), and finally,  $Rating(f, p)$  is the rating given by  $f$  to the publication  $p$ .

As shown in Figure 4, the next step is to sort publications according to their popularity. The Publications with the highest popularity scores are sent back to the requester. The number of publications to return is a parameter embedded in the request.

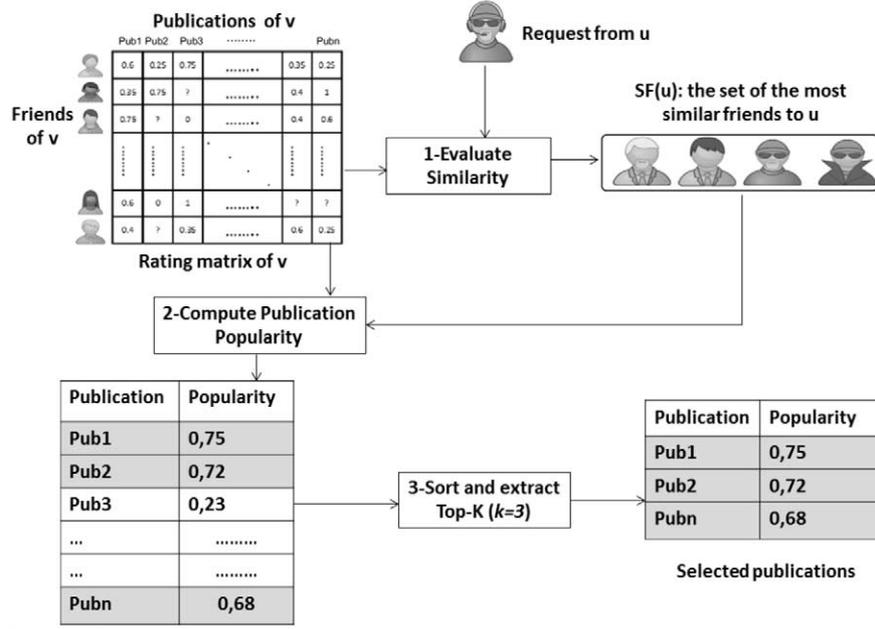


Fig. 4. Publications selection based on similar friends' ratings.

### 3.4.3. Recommendations aggregation

Recommendation gathered from different sources need to be evaluated and curated before they are inserted into user timelines. Global evaluation should be applied to expose the most relevant content to the user who requested recommendations. Another concern is to detect unfair recommendations. For example, in [10] a solution using association rules is proposed to detect unfair recommendations based on the concept of trust. In our case, we rely on the interest level computed using (formula (1)) that embeds some aspects of trust. Indeed, it is easy to verify that levels of trust are proportional to the frequency of interactions. We propose to compute a global score of popularity to give advantage to the publications recommended by users for whom the requester user has higher interest (close friends). We propose to weight the popularity of publication computed by each friend (formula (3)) by the level of interest given by the requester to the friend as per formula (4):

$$Score(p) = PopSim(P) \times Interest\_Level(u, v) \quad (4)$$

Given this, a publication with a high popularity score recommended by a less interesting friend will be dismissed in favor of a publication recommended by a more interesting friend.

As users communicate via P2P infrastructure, the friends' responses arrive in asynchronous way. This is due to the network structure and the different heterogeneous computing capabilities of user devices. To make the latency transparent to the user who receives recommendations, publications are placed in a priority queue according to the computed score. Then, publications are pushed into the user's timeline periodically (see Figure 5).

### 3.5. Cold start problem

One of the main challenges of collaborative filtering recommender systems is the cold start problem [23]. We face this problem when the necessary information that enables recommendation is scarce or not found. In this section, we describe a new mechanism in our proposed solution to overcome the cold start problem that may arise for both new users and newly posted publications.

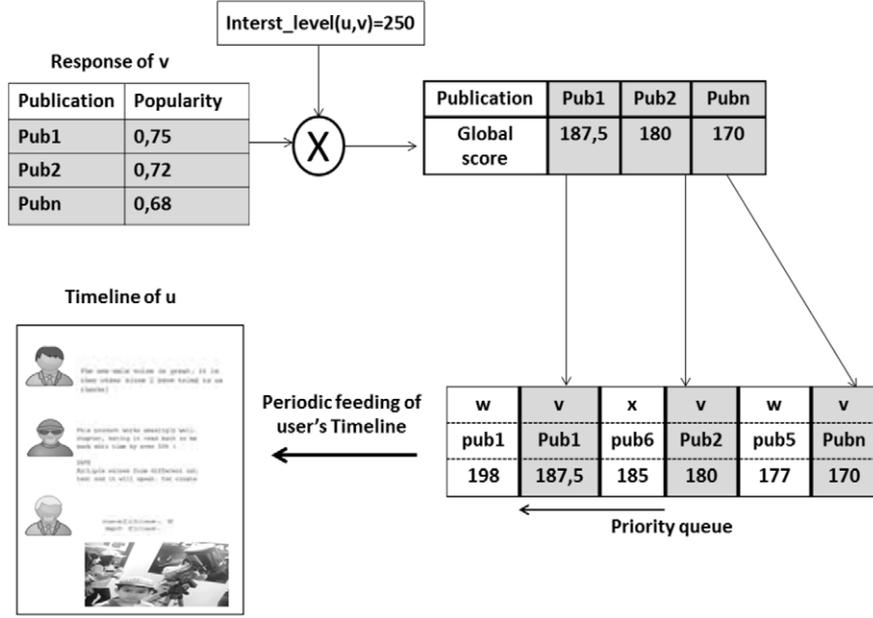


Fig. 5. Publications aggregation and timeline feeding.

### 3.5.1. The case of new users

The cold start user in social networks is illustrated in [18] as a user who does not have any friend or rated items. Many approaches have been proposed to resolve this problem [23], where the main idea is to exploit external information in the recommendation process. However, as the users in P2P networks do not share lots of information about their identity or interests, existing approaches to tackle the cold start problem are not applicable.

In our situation, any new user who joins the system is considered as a cold start user. We also consider users who have less than  $x$  friends with a similarity higher than a predefined threshold as cold start users. In this case, the formula (3) for publication popularity is irrelevant. The receiver  $v$  has not enough information about the requester interests to compute recommendations.

As a solution, we propose that the receiver of the request,  $v$ , returns the publications that are highly appreciated by all his friends. This will return the most popular content to the requester. So, we propose that the user  $v$  computes a popularity score for his own publications, by considering the ratings of all friends. This is illustrated in the formula (5):

$$Pop(u, p) = \frac{\sum_f Rating(f, p)}{Nbf} \quad (5)$$

Where  $Nbf$  is the number of friends connected to user  $v$ . Note that this formula considers all the friends ( $f$ ) of the user  $v$ .

### 3.5.2. The case of new publications

In order to give new publications a chance to be recommended, it is important to put in place a process that bypasses equations (3) and (5). Indeed, when a publication is first posted, it has zero ratings. In this case, each new publication gets a random popularity score between 0 and 1. At the same time, we set a counter to track the number of times the publication has been recommended. Once the counter reaches a predefined number of views (e.g., 5), we substitute the initial random rating with the actual ratings that the publication received. In this way, we make sure that new publications are only favored in the early stages without penalizing the already posted publications.

Table 2  
Summary of basis statistics of the dataset

Nodes (all users)	40,122
Authors of publications	5,714
Links (friendship)	57,222
Publications (tweets)	13,158
Reactions (retweets)	73,002

## 4. Evaluation

In this section, we evaluate the proposed decentralized implementation of collaborative filtering based recommender system for P2P social networks (P2PCF). Our main objective here is to study the behavior of our decentralized technique compared to a conventional centralized collaborative filtering system. To do this, we are interested in two aspects: (i) the recall which measures the ability of the system to return the expected publications, (ii) the coverage, which is the percentage of publications that are picked by the recommender system.

### 4.1. Dataset

The dataset used is extracted from Twitter. We filtered a sample of tweets that dealt with politics events in Algeria in 2019. For each tweet, the collected data contains the user who published the tweet and the users who retweeted (*shared*) this tweet. The collected tweets are then used to build a social network. We generate a friendship link between two users A and B if A retweets at least on tweet published by B. Table 2 describes some characteristics of the social network.

The social network consists of 40,122 users in total, among which 5,714 users are authors of publications.

### 4.2. Implementation details

To implement the collaborative filtering system, we should create the rating matrix. In centralized system, one global matrix is needed. Each row in the matrix represents a user and each column represents a publication (a tweet id in our case). The rating is a Boolean value: 1 if the user retweeted the publication, 0 otherwise. In the decentralized recommender system, every user manages his own local matrix, which is limited to the ratings of his friends on his publications (tweets).

In the context of P2P-SN, the recommendation is limited to friends' publications. We respect this restriction even in the centralized implementation. That is, for every user, the recommender algorithm explores the entire rating matrix, but recommends the friend's publications only.

The basic differences between P2PCF and the centralized baseline are highlighted hereafter:

- **Similarity between requester and other users:** the similarity is calculated according to the previous rating of the users in the whole network. In P2PCF, friends recipient of requests use their local matrices to calculate local similarity between the requester and other similar users.
- **Cold start decision:** some cold start users (*in the decentralized situation*) are not qualified as cold start in the centralized approach. In decentralized scheme, a user with less interactions with a given friend is considered as cold start by this friend. The central server manages all users' interactions in the network, therefore it has access to more data about the user.

### 4.3. Recall

It is important to observe the recall of our system compared to the centralized baseline. In this experiment, and in order to have a fair comparison, all friends are selected. We will evaluate later the effect of selecting a subset of friends on the algorithm performance.

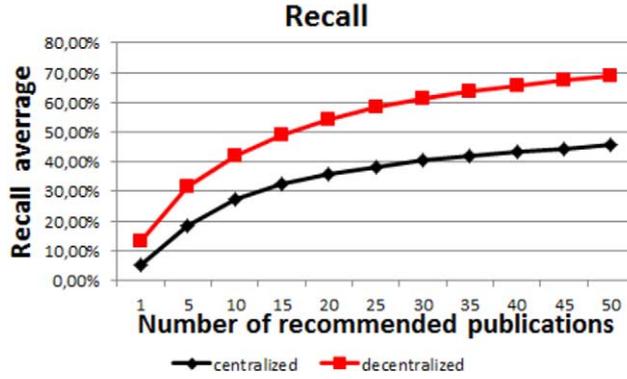


Fig. 6. Recall average.

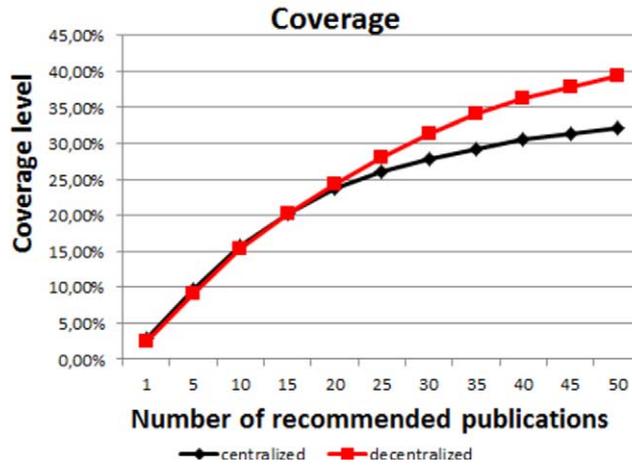


Fig. 7. Coverage level.

We impute 20% of rated publications by each user, randomly chosen. We then execute the recommendation process. The recall is the ability of the system to recommend those deleted publications. It is calculated as the percentage of the expected publications (the imputed ones) that are correctly recommended by the system. The recommended publications are sorted according to formula (4) and the recall is computed for different sizes of recommendation sets. The recall is calculated for each user and the average of recalls is shown in Figure 6.

#### 4.4. Coverage

We calculate the coverage level as the percentage of publications that have been recommended (*at least once*). This metric shows the ability of the recommender system to give a chance to all publications to be seen by users. Figure 7 shows the coverage level in both centralized and decentralized systems.

#### 4.5. Friends selection

In the two previous experiences, we compared the centralized and decentralized approaches in similar conditions. We did not limit the number of friends concerned by recommendations. In addition, all the relevant publications to recommend are returned to the requester.

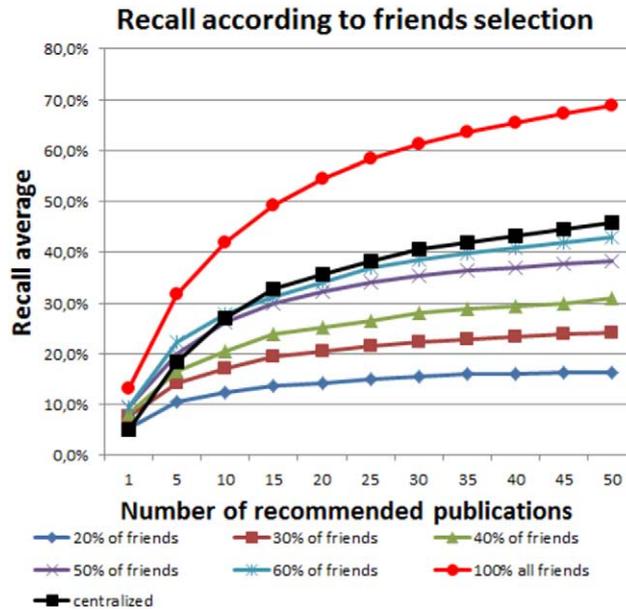


Fig. 8. Recall average according to the percentage of friends selected.

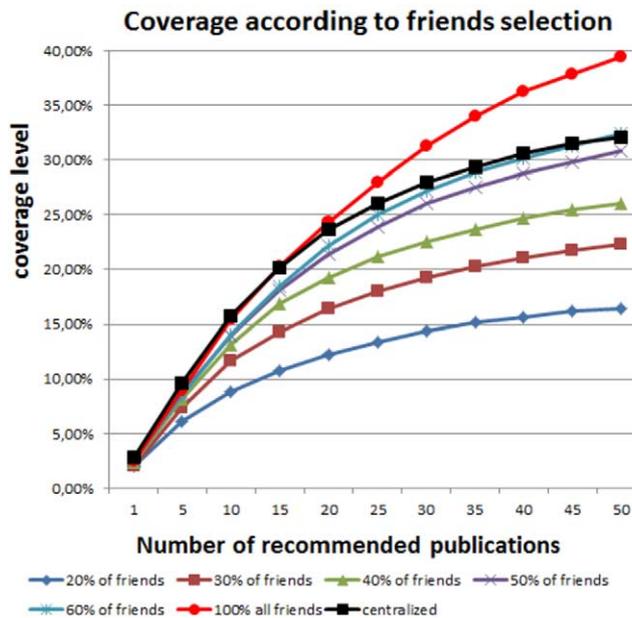


Fig. 9. Coverage level according to the percentage of friends selected.

In practice, sending a recommendation request to all the friends is expensive in terms of network overload. Thus, we evaluate in this section the effect of the number of selected friends on the recommender system. Figure 8 shows the average recall as a function of to the percentage of selected friends.

Similarly, we report the coverage level according to the percentage of selected friends in Figure 9.

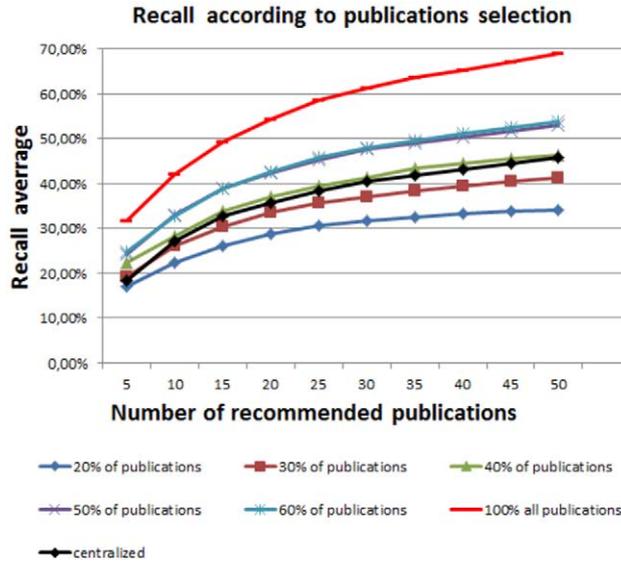


Fig. 10. Recall average according to the percentage of publications selected by every friend.

#### 4.6. Publications selection

When a friend is asked for recommendation, he should respond by sending to the requester a set of relevant publications. The number of the returned publications affects the networks overload and the quality of recommendation. In this experiment, we study the effect of the size of the recommended publication sets on our system's performance. Figure 10 shows the recall scores according to the percentage of publications selected by every requested friend.

To show the effect of limiting the number of publications to recommend on the publications visibility, we report in Figure 11 the coverage level function of the number of requested publications.

#### 4.7. Discussion

The first observation we can make here is that the collaborative filtering algorithm can be effectively implemented in decentralized way to enrich the peer to peer social networks.

In the first experiment, we set the algorithm with the best case parameters: all friends are selected and all relevant publications are sent back to the user. Figure 6 shows that the recall of P2PCF is better than the centralized CF. Coverage improvement is also shown in Figure 7.

This experiment proves the utility of distributing the filtering task among users. All friends contribute by sending their publications to the requester. This improves the coverage by giving the chance for unpopular users' publications. Compared to the centralized implementation where the filtering is performed on a global matrix, unpopular publications and friends can be marginalized against popular ones. Some users may have specific interests to those unpopular publications (*real life friends or cousins' publications*). In this case, the decentralized CF can reach better these publications by browsing all friends.

Unfortunately the previous configuration is very expensive in term of networks overload. Our algorithm proposes that the requester should select a subset of his friends to ask them for recommendations. In the second experiment, we tuned the percentage of friends to select and reported recall and coverage. By analyzing Figure 8 (*recall*) and Figure 9 (*coverage*), we note that at least 50% of friends' should be selected to get similar results as those achieved by the baseline centralized system.

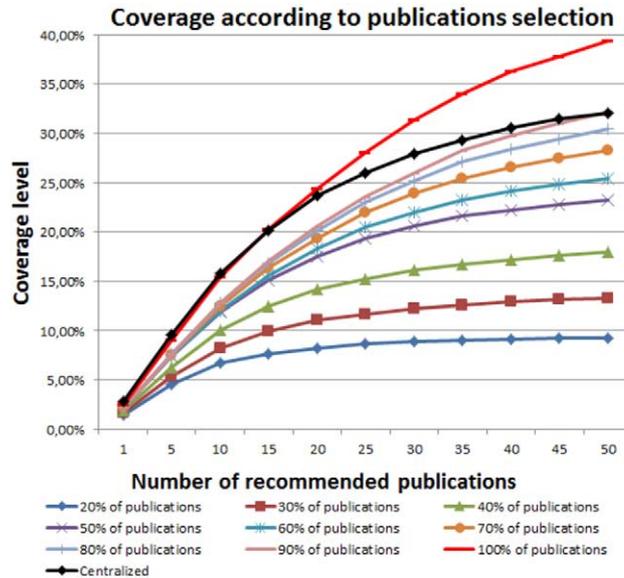


Fig. 11. Coverage level according to the percent of publication selected by every friend.

The last experiment concerns the number of recommendations returned by friends. We found that returning 40% of recommendations from each friend is enough to achieve a comparable recall to that of the centralized system (Figure 10). In term of coverage (Figure 11), at least 90% of publications should be recommended to cover the same percent of publications as the centralized algorithm.

## 5. Conclusions and future works

In this paper we presented P2PCF – a collaborative filtering-based recommender system for P2P social networks – which primarily focuses on ensuring data privacy. To respect the privacy restrictions of P2P social networks, we assume that the interaction matrix is distributed within peers, with no peer having access to the global matrix. That is, each user manages locally the reactions occurred on her timeline without exchanging this information with other users. When a user requests recommendations, a request is submitted to her friends who generate locally relevant lists of items. These lists are aggregated upon reception and prioritized based on the strength of the ties between the requester and her friends. We have evaluated the effectiveness of our scheme and compared it to centralized recommender systems, by adapting real-life data from Twitter. Our evaluation revealed a significant improvement in terms of recall and coverage when all the friends are involved in recommendation process. We also found that requesting recommendations from only 50% of the one's friends is enough to achieve similar scores as those reported by the centralized system. Our system is designed with strict privacy constraints. Thus, we would like to focus on future works on the effect of relaxing some of these constraints on the recommendation quality. This includes for instance exchanging parts of the local ratings with trusted friends, which could lead to more accurate and personalized recommendations.

## References

- [1] <https://www.nytimes.com/2019/08/30/technology/youtube-childrens-privacy-fine.html>.
- [2] <https://www.nytimes.com/2020/08/03/technology/ftc-twitter-privacy-violations.html>.

- [3] S. Abbar, R. Stanojevic and M. Mokbel, STAD: Spatio-temporal adjustment of traffic-oblivious travel-time estimation, in: *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, Versailles, France, 2020, pp. 79–88. doi:[10.1109/MDM4852.2020.00029](https://doi.org/10.1109/MDM4852.2020.00029).
- [4] L. Badis, D. Aissani and M. Amad, A log based update of replicated profiles in decentralized social networks, *Journal of Digital Information Management* **16**(5) (2018), 231. doi:[10.6025/jdim/2018/16/5/230-245](https://doi.org/10.6025/jdim/2018/16/5/230-245).
- [5] S. Buchegger, D. Schioberg, L.-H. Vu et al., PeerSoN: P2P social networking: Early experiences and insights, in: *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, ACM, 2009, pp. 46–52. doi:[10.1145/1578002.1578010](https://doi.org/10.1145/1578002.1578010).
- [6] C. Chen, Z. Liu and P. Zhao et al., Privacy preserving point-of-interest recommendation using decentralized matrix factorization, in: *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [7] G.G. Chowdhury, *Introduction to Modern Information Retrieval*, Facet Publishing, 2010.
- [8] M. Conti, A. De Salve, B. Guidi et al., Epidemic diffusion of social updates in Dunbar-based dosn, in: *European Conference on Parallel Processing*, Springer, Cham, 2014, pp. 311–322. doi:[10.1007/978-3-319-14325-5\\_27](https://doi.org/10.1007/978-3-319-14325-5_27).
- [9] M. Conti, A. De Salve, B. Guidi, F. Pitto and L. Ricci, Trusted dynamic storage for Dunbar-based P2P online social networks, in: *OTM Conferences*, 2014, pp. 400–417. doi:[10.1007/978-3-662-45563-0\\_23](https://doi.org/10.1007/978-3-662-45563-0_23).
- [10] G. D'Angelo, F. Palmieri and S. Rampone, Detecting unfair recommendations in trust-based pervasive environments, *Information Sciences* **486** (2019), 31–51. doi:[10.1016/j.ins.2019.02.015](https://doi.org/10.1016/j.ins.2019.02.015).
- [11] G. D'Angelo, R. Pilla, J.B. Dean et al., Toward a soft computing-based correlation between oxygen toxicity seizures and hyperoxic hyperpnea, *Soft Computing* **22**(7) (2018), 2421–2427. doi:[10.1007/s00500-017-2512-z](https://doi.org/10.1007/s00500-017-2512-z).
- [12] G. D'Angelo, R. Pilla, C. Tascini et al., A proposal for distinguishing between bacterial and viral meningitis using genetic programming and decision trees, *Soft Computing* **23**(22) (2019), 11775–11791. doi:[10.1007/s00500-018-03729-y](https://doi.org/10.1007/s00500-018-03729-y).
- [13] A. Datta and R. Sharma, Godisco: Selective gossip based dissemination of information in social community based overlays, in: *International Conference on Distributed Computing and Networking*, Springer, Berlin, Heidelberg, 2011, pp. 227–238. doi:[10.1007/978-3-642-17679-1\\_20](https://doi.org/10.1007/978-3-642-17679-1_20).
- [14] A. De Salve, M. Dondio, B. Guidi et al., The impact of user's availability on on-line ego networks: A Facebook analysis, *Computer Communications* **73** (2016), 211–218. doi:[10.1016/j.comcom.2015.09.001](https://doi.org/10.1016/j.comcom.2015.09.001).
- [15] M. Del Carmen Rodriguez-Hernandez, S. Ilarri, R. Trillo et al., Context-aware recommendations using mobile P2P, in: *Proceedings of the 15th International Conference on Advances in Mobile Computing & Multimedia*, ACM, 2017, pp. 82–91. doi:[10.1145/3151848.3151856](https://doi.org/10.1145/3151848.3151856).
- [16] A. Demers, D. Greene, C. Houser et al., Epidemic algorithms for replicated database maintenance, *ACM SIGOPS Operating Systems Review* **22**(1) (1988), 8–32. doi:[10.1145/43921.43922](https://doi.org/10.1145/43921.43922).
- [17] R. Djerbi, M. Amad and R. Imache, A new model for communities' detection in dynamic social networks inspired from human families, *Int. J. of Internet Technology and Secured Transactions, Inderscience* **10**(1/2) (2020), 24–60. doi:[10.1504/IJITST.2020.104574](https://doi.org/10.1504/IJITST.2020.104574).
- [18] Y. Dou, H. Yang and X. Deng, A survey of collaborative filtering algorithms for social recommender systems, in: *12th International Conference on: Semantics, Knowledge and Grids (SKG)*, IEEE, 2016, pp. 40–46. doi:[10.1109/SKG.2016.014](https://doi.org/10.1109/SKG.2016.014).
- [19] F. Draïdi, E. Pacitti, M. Cart et al., Leveraging social and content-based recommendation in P2P systems, in: *AP2PS11: 3rd International Conference on Advances in P2P Systems*, 2011, pp. 13–18.
- [20] F. Draïdi, E. Pacitti and B. Kemme, P2Prec: A P2P recommendation system for large-scale data sharing, in: *Transactions on Large-Scale Data-and Knowledge-Centered Systems III*, Springer, Berlin, Heidelberg, 2011, pp. 87–116. doi:[10.1007/978-3-642-23074-5\\_4](https://doi.org/10.1007/978-3-642-23074-5_4).
- [21] E. Duriakova, E.Z. Tragos, B. Smyth et al., PDMFRec: A decentralised matrix factorisation with tunable user-centric privacy, in: *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 457–461. doi:[10.1145/3298689.3347035](https://doi.org/10.1145/3298689.3347035).
- [22] W. Fan, Y. Ma, Q. Li et al., Graph neural networks for social recommendation, in: *The World Wide Web Conference*, 2019, pp. 417–426.
- [23] J. Gope and S.K. Jain, A survey on solving cold start problem in recommender systems, in: *International Conference on Computing, Communication and Automation (ICCCA)*, IEEE, 2017, pp. 133–138. doi:[10.1109/CCAA.2017.8229786](https://doi.org/10.1109/CCAA.2017.8229786).
- [24] K. Graffi, C. Gross, D. Stingl et al., LifeSocial.KOM: A secure and P2P-based solution for online social networks, in: *Consumer Communications and Networking Conference (CCNC)*, IEEE, 2011, pp. 554–558. doi:[10.1109/CCNC.2011.5766541](https://doi.org/10.1109/CCNC.2011.5766541).
- [25] B. Guidi, M. Conti, A. Passarella and L. Ricci, Managing social contents in decentralized online social networks: A survey, *Online Social Networks and Media* **7** (2018), 12–29. doi:[10.1016/j.osnem.2018.07.00](https://doi.org/10.1016/j.osnem.2018.07.00).
- [26] I. Guy and D. Carmel, Social recommender systems, in: *Proceedings of the 20th International Conference Companion on World Wide Web*, ACM, New York, 2011, pp. 283–284. doi:[10.1007/978-1-4899-7637-6\\_15](https://doi.org/10.1007/978-1-4899-7637-6_15).
- [27] A. Hacini, M. Amad and F. Semchedine, A scalable and hierarchical P2P architecture based on pancake graph for group communication, *Journal of High Speed Networks* **23** (2017), 287–309. doi:[10.3233/JHS-170572](https://doi.org/10.3233/JHS-170572).
- [28] P. Han, B. Xie, F. Yang et al., A scalable P2P recommender system based on distributed collaborative filtering, *Expert Systems with Applications* **27**(2) (2004), 203–210. doi:[10.1016/j.eswa.2004.01.003](https://doi.org/10.1016/j.eswa.2004.01.003).
- [29] W. Hariri, K.I. Ghauth and C. Eswaran, A multimedia content recommender system using table of contents and content-based filtering, *Advanced Science Letters* **24**(2) (2018), 1119–1123. doi:[10.1166/asl.2018.10699](https://doi.org/10.1166/asl.2018.10699).
- [30] A. Hinduja and M. Pandey, An intuitionistic fuzzy AHP based multi criteria recommender system for life insurance products, *International Journal of Advanced Studies in Computers, Science and Engineering* **7**(1) (2018), 1–8.
- [31] L. Huang, W. Tan and Y. Sun, Collaborative recommendation algorithm based on probabilistic matrix factorization in probabilistic latent semantic analysis, *Multimedia Tools and Applications* **78**(7) (2019), 8711–8722. doi:[10.1007/s11042-018-6232-x](https://doi.org/10.1007/s11042-018-6232-x).

- [32] P.D. Hung and D. Le Huynh, E-commerce recommendation system using mahout, in: *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, Singapore, 2019, pp. 86–90. doi:[10.1109/CCOMS.2019.8821663](https://doi.org/10.1109/CCOMS.2019.8821663).
- [33] S. Jahid, S. Nilizadeh, P. Mittal et al., DECENT: A decentralized architecture for enforcing privacy in online social networks, in: *International Conference on Pervasive Computing and Communications Workshops*, IEEE, 2012, pp. 326–332. doi:[10.1109/PerComW.2012.6197504](https://doi.org/10.1109/PerComW.2012.6197504).
- [34] P. Karamolegkos and N. Doulamis, Multimedia file exchange and P2P social networking: Efficient allocation of users into communities of interest, in: *International Conference on Ultra Modern Telecommunications & Workshops. ICUMT '09*, IEEE, 2009, pp. 1–4. doi:[10.1109/ICUMT.2009.5345647](https://doi.org/10.1109/ICUMT.2009.5345647).
- [35] D. Koll, D. Lechler and X. Fu, SocialGate: Managing large-scale social data on home gateways, in: *25th International Conference on Network Protocols (ICNP)*, IEEE, 2017, pp. 1–6. doi:[10.1109/ICNP.2017.8117590](https://doi.org/10.1109/ICNP.2017.8117590).
- [36] D. Koll, J. Li and X. Fu, Soup: An online social network by the people, for the people, in: *Proceedings of the 15th International Middleware Conference*, ACM, 2014, pp. 193–204. doi:[10.1145/2663165.2663324](https://doi.org/10.1145/2663165.2663324).
- [37] D. Liu, A. Shakimov, R. Caceres et al., Confidant: Protecting OSN data without locking it up, in: *Proceedings of the 12th International Middleware Conference*, International Federation for Information Processing, 2011, pp. 60–79.
- [38] G. Mega, A. Montresor and G. Picco, Efficient dissemination in decentralized social networks, in: *International Conference on Peer-to-Peer Computing*, IEEE, 2011, pp. 338–347. doi:[10.1109/P2P.2011.6038753](https://doi.org/10.1109/P2P.2011.6038753).
- [39] M. Millecamp, N.N. Htun, Y. Jin et al., Controlling Spotify recommendations: Effects of personal characteristics on music recommender user interfaces, in: *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, 2018, pp. 101–109. doi:[10.1145/3209219.3209223](https://doi.org/10.1145/3209219.3209223).
- [40] R.J. Mooney and L. Roy, Content-based book recommending using learning for text categorization, in: *Proceedings of the Fifth ACM Conference on Digital Libraries*, ACM, 2000, pp. 195–204. doi:[10.1145/336597.336662](https://doi.org/10.1145/336597.336662).
- [41] T.T. Nguyen, P.-M. Hui and F.M. Harper et al., Exploring the filter bubble: The effect of using recommender systems on content diversity, in: *Proceedings of the 23rd International Conference on World Wide Web*, ACM, 2014, pp. 677–686. doi:[10.1145/2566486.2568012](https://doi.org/10.1145/2566486.2568012).
- [42] S. Nilizadeh, S. Jahid, P. Mittal et al., Cachet: A decentralized architecture for privacy preserving social networking with caching, in: *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, ACM, 2012, pp. 337–348. doi:[10.1145/2413176.2413215](https://doi.org/10.1145/2413176.2413215).
- [43] J.A. Obar and A. Oeldorf-Hirsch, The biggest lie on the Internet: Ignoring the privacy policies and terms of service policies of social networking services, *Information, Communication & Society* (2018), 1–20. doi:[10.1080/1369118X.2018.1486870](https://doi.org/10.1080/1369118X.2018.1486870).
- [44] M.J. Paani, A framework for collaborative, content-based and demographic filtering, *Artificial Intelligence Review* **13**(5–6) (1999), 393–408. doi:[10.1023/A:100654452](https://doi.org/10.1023/A:100654452).
- [45] T. Paul, A. Famulari and T. Strufe, A survey on decentralized online social networks, *Computer Networks* **75** (2014), 437–452. doi:[10.1016/j.comnet.2014.10.005](https://doi.org/10.1016/j.comnet.2014.10.005).
- [46] A.A. Ranjan, A. Rai, S. Haque et al., An approach for Netflix recommendation system using singular value decomposition, *Journal of Computer and Mathematical Sciences* **10**(4) (2019), 774–779. doi:[10.29055/jcms/1063](https://doi.org/10.29055/jcms/1063).
- [47] A. Rao, N. Spasojevic, Z. Li et al., Klout score: Measuring influence across multiple social networks, in: *International Conference on Big Data (Big Data)*, IEEE, 2015, pp. 2282–2289. doi:[10.1109/BigData.2015.7364017](https://doi.org/10.1109/BigData.2015.7364017).
- [48] R.L. Rosa, E.L.L. Junior and D.Z. Rodriguez, A recommendation system for shared-use mobility service through data extracted from online social networks, *Journal of Communications Software and Systems* **14**(4) (2018), 359–366. doi:[10.24138/jcomss.v14i4.602](https://doi.org/10.24138/jcomss.v14i4.602).
- [49] S.M. Satapathy, R. Jhaveri, U. Khanna et al., Smart rent portal using recommendation system visualized by augmented reality, *Procedia Computer Science* **171** (2020), 197–206.
- [50] C.O. Schneble, B.S. Elger and D. Shaw, The Cambridge Analytica affair and Internet-mediated research, *EMBO Reports* **19**(8) (2018), e46579. doi:[10.15252/embr.201846579](https://doi.org/10.15252/embr.201846579).
- [51] N. Shahriar, S.R. Chowdhury, R. Ahmed et al., Availability in P2P based online social networks, in: *4th International Conference on Networking, Systems and Security (NSysS)*, IEEE, 2017, pp. 1–9. doi:[10.1109/NSYS2.2017.8267785](https://doi.org/10.1109/NSYS2.2017.8267785).
- [52] Q. Shambour and J. Lu, A trust-semantic fusion-based recommendation approach for e-business applications, *Decision Support Systems* **54**(1) (2012), 768–780. doi:[10.1016/j.dss.2012.09.005](https://doi.org/10.1016/j.dss.2012.09.005).
- [53] A. Sivanathan, H.H. Gharakheili and V. Sivaraman, Managing IoT cyber-security using programmable telemetry and machine learning, *IEEE Transactions on Network and Service Management* **17**(1) (2020), 60–74. doi:[10.1109/TNSM.2020.2971213](https://doi.org/10.1109/TNSM.2020.2971213).
- [54] T. Strufe, Safebook: A privacy-preserving online social network leveraging on real-life trust, *IEEE Communications Magazine* **95** (2009). doi:[10.1109/MCOM.2009.5350374](https://doi.org/10.1109/MCOM.2009.5350374).
- [55] U. Tandukar and J. Vassileva, Selective propagation of social data in decentralized online social network, in: *International Conference on User Modeling, Adaptation, and Personalization*, Springer, Berlin, Heidelberg, 2011, pp. 213–224. doi:[10.1007/978-3-642-28509-7\\_20](https://doi.org/10.1007/978-3-642-28509-7_20).
- [56] J. Tang, H. Gao, X. Hu and H. Liu, Exploiting homophily effect for trust prediction, in: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, ACM, New York, 2013, pp. 53–62. doi:[10.1145/2433396.2433405](https://doi.org/10.1145/2433396.2433405).
- [57] J. Tang, X. Hu and H. Liu, Social recommendation: A review, *Social Network Analysis and Mining* **3**(4) (2013), 1113–1133. doi:[10.1007/s13278-013-0141-9](https://doi.org/10.1007/s13278-013-0141-9).
- [58] A. Tveit, Peer-to-peer based recommendations for mobile commerce, in: *International Workshop on Mobile Commerce: Proceedings of the 1st International Workshop on Mobile Commerce*, 2001, pp. 26–29. doi:[10.1145/381461.381466](https://doi.org/10.1145/381461.381466).

- [59] D. Wang, Y. Liang, D. Xu et al., A content-based recommender system for computer science publications, *Knowledge-Based Systems* **157** (2018), 1–9. doi:[10.1016/j.knosys.2018.05.001](https://doi.org/10.1016/j.knosys.2018.05.001).
- [60] X. Wang, X. Yang and L. Guo et al., Exploiting social review-enhanced convolutional matrix factorization for social recommendation, *IEEE Access* **7** (2019), 82826–82837.
- [61] J. Weng, E. Lim, J. Jiang and Q. He, Twitterrank: Finding topicsensitive influential twitterers, in: *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, ACM, New York, 2010, pp. 261–270. doi:[10.1145/1718487.1718520](https://doi.org/10.1145/1718487.1718520).
- [62] G. Xu, Z. Wu, Y. Zhang et al., Social networking meets recommender systems: Survey, *International Journal of Social Network Mining* **2**(1) (2015), 64–100. doi:[10.1504/IJSNM.2015.069773](https://doi.org/10.1504/IJSNM.2015.069773).
- [63] L. Yao, Z. Xu, X. Zhou et al., Synergies between association rules and collaborative filtering in recommender system: An application to auto industry, in: *Data Science and Digital Business*, Springer, Cham, 2019, pp. 65–80. doi:[10.1007/978-3-319-95651-0\\_5](https://doi.org/10.1007/978-3-319-95651-0_5).
- [64] Y. Zhong, L. Du and J. Yang, Learning social relationship strength via matrix co-factorization with multiple kernels, in: *International Conference on Web Information Systems Engineering*, Springer, Berlin, Heidelberg, 2013, pp. 15–28. doi:[10.1007/978-3-642-41230-1\\_2](https://doi.org/10.1007/978-3-642-41230-1_2).
- [65] Y. Zhong, W. Zhao and J. Yang, Personal-hosting restful web services for social network based recommendation, in: *Service-Oriented Computing (SOC)*, 2011, pp. 661–668. doi:[10.1007/978-3-642-25535-9\\_53](https://doi.org/10.1007/978-3-642-25535-9_53).
- [66] M. Zignani, S. Gaito and G.P. Rossi, Follow the “Mastodon”: Structure and evolution of a decentralized online social network, in: *Twelfth International AAAI Conference on Web and Social Media*, 2018.
- [67] M. Zihayat, A. Ayanso, X. Zhao et al., A utility-based news recommendation system, *Decision Support Systems* **117** (2019), 14–27. doi:[10.1016/j.dss.2018.12.001](https://doi.org/10.1016/j.dss.2018.12.001).