

Distributed Collaborative Filtering for Peer-to-Peer File Sharing Systems

Jun Wang Marcel J.T. Reinders Reginald L. Lagendijk Johan Pouwelse

Information and Communication Theory Group,
Faculty of Electrical Engineering, Mathematics and Computer Science,
Delft University of Technology
{j.wang, m.j.t.reinders, r.l.lagendijk, j.a.pouwelse}@ewi.tudelft.nl

Keywords: Recommendation, Collaborative Filtering, Peer-to-Peer Networks, Distributed Systems

Abstract

Peer-to-peer networks are becoming more and more popular to share information such as, for example, multimedia files. Since this information is stored locally at the different peers, it is necessary to facilitate the search in an intelligent way. Collaborative filtering is such a search technique that enables to incorporate the preferences of a user that can be learned from the download activities of the users. To be effective collaborative filtering requires a large database that captures these activities. Within a peer-to-peer network this is, however, not readily available. Here, we propose a collaborative filtering approach that is *self-organizing* and operates in a *distributed* way. Information about the similarity between multimedia files (items) is stored locally at these items in so called *item-based buddy tables*. We propose to use the language model (popular within information retrieval) to build recommendations for the different users based on the buddy tables of those items a user has downloaded previously (indicating the preference of the user). We have tested and compared our distributed collaborative filtering approach to centralized collaborative filtering and showed that it has similar performance. It is therefore a promising technique to facilitate the search for information in peer-to-peer networks.

1 Introduction

The rapid progress in information processing, communications, and storage technologies, peer-to-peer (P2P) networks have become a new way for people to exchange information that is stored on their local storage devices (examples are: Freenet¹, Gnutella², and BitTorrent³). Since P2P net-

works are inherently distributed, unstructured and unreliable (e.g. peers are not always connected), mechanisms for searching the available information are not straightforward. There have been some efforts to organize the information in P2P networks, such as, for example, unstructured approaches (i.e. Gnutella), semantic-free structured approaches ([19, 22]) and semantic structure approaches ([21, 24]). Most of these efforts aim to make meta-data based content search possible ([17]) by means of keyword search.

Alternatively, in this paper, we propose a self-organizing distributed binary collaborative filtering approach that enables to organize and recommend content within the context of a P2P network. The similarity between content (items) are derived from the profiles of the different users and stored in a distributed way as *item based buddy tables*. By using the item-buddy tables, items are organized in the form of relevance links. Recommendation can then be done according to the similarities stored in the item-buddy tables. We show that our distributed approach is statistically equivalent to a centralized item-based collaborative filtering. We consider the application where users share multimedia files with other users.

1.1 Related work on collaborative filtering

Generically, collaborative filtering is any algorithm that filters information for a user based on a collection of user profiles. In major collaborative filtering literature, only a fraction of the formulations has been studied in depth ([14]). One common characteristic of these algorithms is that they require a centralized user-item rating matrix as the input source. These approaches can be divided into two categories: 1) memory-based ([15, 2, 10]), and 2) model-based methods ([8, 3, 14]).

Recently, a few early attempts towards decentralized collaborative filtering have been introduced

¹<http://freenet.sourceforge.net>

²<http://www.gnutella.com>

³<http://bitconjurer.org/BitTorrent/>

([3, 16]). Canny ([3, 4]) proposed a distributed EM (Expectation Maximization) update scheme. However, the update is partially distributed since a "taller" (server) is still required to connect with all the peers. In addition, the assumption made about the missing data makes it incapable for binary recommendation problems. In [23], an unstructured routing algorithm similar to Gnutella has been applied to forward the query (rating) of the neighbors. In [16], a DHTs based technique was proposed to store the user rating data. A key lookup procedure was performed to collect the rating data of users before making any recommendation. Those solutions aimed to aggregate a portion of data in the P2P storage in order to make a recommendation and they hold independently of any semantic structure of the networks. This inevitably increases the volume of traffic within the networks.

1.2 Related work on searching in P2P environments

For a recent comprehensive survey on P2P networks we refer to [17]. Different index techniques for resources located at the different peers exists, such as: a local index (the owner of the data is only able to index the data, like in early Gnutella, the central index (a centralized server organizes indices to data residing at peers, like in Napster), and the distributed index (other peers are also able to index the data residing at a peer, like in Freenet).

One dominant indexing approach is the Distributed Hash Tables (DHTs). In a DHT each location (index) is mapped to a unique key, and each peer maintains a certain range of the keys. In this way each peer generates a well-defined structure that can be used for routing queries that is scalable to some extent. An extension, however, is necessary to perform a search based on arbitrary queries rather than key lookups ([6]).

Recently, to capture relationships between resources, semantic indexing and routing techniques have been proposed ([6, 17]). Distinct semantic groups of documents ([5]) or users ([21, 24]) are identified to create Semantic Overlay Networks (SONs). A document request is then handled by the overlay to which this document presumably belongs (based on either clusters of documents or peers). However, to match queries to documents a content description (in the form of meta-data) is required. Identifying similarities between peers or documents turns out to be difficult to establish in the absence of meta-data (content description).

2 System Design

In section, we introduce our self-organizing distributed collaborative filtering.

2.1 Definitions

We first give some notations and definitions.

Peer: A peer is represented by:

$$P_i, i = \{1, \dots, M\} \quad (1)$$

where M is the total number of peers within the P2P network. Without loss of generality we assume that there is only one user for one peer. Consequently, we use the terms peer and user interchangeably throughout the paper.

Item: For reasons of simplicity, here we assume that there are no replicas of the content within the P2P network (We will discuss the situation when there are numbers of replicas later). Available content files (visual, audio, and textual data, etc.) are denoted as a set of *items*. One item represents one file located at a specific peer. The set of items is denoted by:

$$R_t = \{I_i^k \text{ is available} | i \in \{1, \dots, M\}; k \in \{1, \dots, K_i\}\} \quad (2)$$

where R_t is the set of available multimedia files at a certain moment in time t . Items I_i^k are identified by their location (i.e., the k th item in the i th peer). K_i denotes the number of items physically located at peer P_i . Since not all the peers may be online at a certain moment in time, the availability of items will vary over time which is indicated by the subscription t .

For simplicity, the set of content files can be represented otherwise (by combining index k and i into a new index j):

$$R_t = \{I_j \text{ is available} | j \in \{1, \dots, N = \sum_{i=1}^M K_i\}\} \quad (3)$$

where N is the total number of items in the networks. By using this, the unique identity of each item is established by the location itself.

Shelf: A *Shelf* is a set of items belonging to one peer, which that peer is willing to share with other peers. Based on this definition, items can be organized by *Shelves*. If we assume that all the items are allowed to be shared, Equation (2) can be written otherwise as follows:

$$R_t = \{S_{i,t} | i = \{1, \dots, M\}\} \\ \text{Shelf: } S_{i,t} = \{I_i^k \text{ is available} | k = \{1, \dots, K_i\}\} \quad (4)$$

Transaction: When a user connects to another peer to download a multimedia file of that peer, this creates a transaction. A transaction is denoted as:

$$\text{Trans: } T_t = \{P_i, I_j, t\} \quad (5)$$

where a user P_i connects to the peer where item I_j is located and download item I_j at time t .

Cart: A *Cart* is a list of items that a peer has once downloaded in the past.

$$C = \{C_i | i = \{1, \dots, M\}\} \\ \text{Cart: } C_i = \{I_j | P_i \text{ has once downloaded item } I_j\} \quad (6)$$

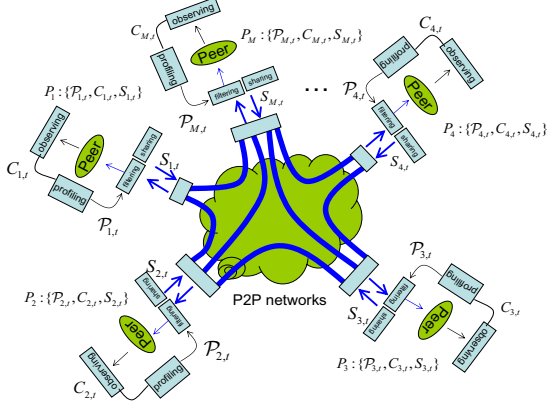


Figure 1: A schematic overview of different notations that have been introduced for the sharing multimedia files within a P2P network.

where C is the set of carts of all users, and C_i the cart of user P_i .

The cart, therefore, lists all items that a users once has represented interest in, and consequently represents the preference of that user. However, the interest of a user may evolve over time. Therefore, a time window is applied to the cart to filter out the old items. The filtered cart is denoted as $C_{i,t}$. Furthermore, in a P2P network peers are frequently not available. Consequently, the items in the cart can be classified as either *active* and *non-active* items depending on whether the peer that possesses these items is available or not available. The time-dependent cart can thus be split accordingly: $C_{i,t} = C_{i,t}^{active} + C_{i,t}^{non-active}$. Since the non-active items are not accessible these items within the cart they will not be considered when a recommendation is being created.

Profile: The profile of a user represents the current interest of a user. This is represented by the set of *active* items. Thus, the profile for peer P_i at a give time t is given by:

$$\text{Profile: } \mathcal{P}_{i,t} = C_{i,t}^{active}, i = \{1, \dots, M\} \quad (7)$$

Individual items in the profile ($\mathcal{P}_{i,t}$) of peer (P_i) are denoted as c_i^q indicating the q th item of peer P_i , with:

$$c_i^q \in \{I_j | j = \{1, \dots, N\}\}, q = \{1, \dots, Q_i\} \quad (8)$$

Fig. 1 gives a schematic overview on how multimedia files are shared within the P2P network. The items in the shelves of the individual peers ($S_{i,t}$) together represent all the multimedia files that can be downloaded. Users download those multimedia files based on their current interest that is represented by their profiles ($\mathcal{P}_{i,t}$). The figure also shows that both the multimedia files, the user profiles as well as the lists of previously downloaded items ($C_{i,t}$) are distributed throughout the entire network.

2.2 Collaborative filtering by language modelling

In the information retrieval field, a relevance model [11, 1, 18] is used to formulate the relevance of documents with respect to the query. Here, we too adopt this relevance model. We introduce a binary random variable R to denote the relevance of an item based on the profile of a user. This random variable takes on two values: r (“relevant”), and \bar{r} “not relevant”). In probabilistic terms, the relevance between a target item I_T and a user’s profile $\mathcal{P}_{i,t}$ can be expressed as:

$$P(R = r | I_T, \mathcal{P}_{i,t}) = 1 - P(R = \bar{r} | I_T, \mathcal{P}_{i,t}) \quad (9)$$

where $I_T \in S_{s,t}$, $s \in \{1, \dots, i-1, i+1, \dots, M\}$, i.e. item I_T is an item that is not within the shelf of peer P_i . For reasons of clarity we abbreviate $P(R = r | I_T, \mathcal{P}_{i,t})$ with $P(r | I_T, \mathcal{P}_{i,t})$.

When these probabilities are known for any item, they can be used to rank the items that are not in the cart of the user. A recommendation can then be established by taking the *top-N* items of the ranked list.

The relevance rank of the target item I_T for a peer P_i can be formulated as:

$$R_{I_T, \mathcal{P}_i} = \log \frac{P(r | I_T, \mathcal{P}_i)}{P(\bar{r} | I_T, \mathcal{P}_i)} \quad (10)$$

where the relevance rank $R_{I_T, \mathcal{P}_{i,t}}$ of item I_T for peer P_i is calculated by the conditional probability in the log form. Since the user profile \mathcal{P}_i equals the active items in the cart (equation (7)) we can rewrite equation (10). For reasons of clarity we substitute $C_{i,t}^{active}$ by C_i .

$$R_{I_T, \mathcal{P}_i} = \log \frac{P(r | I_T, C_i)}{P(\bar{r} | I_T, C_i)} \quad (11)$$

By factorizing $P(r | I_T, C_i)$ with $\frac{P(C_i | I_T, r) P(r)}{P(C_i | I_T)}$, the following log-odds ratio can be obtained:

$$R_{I_T, \mathcal{P}_i} = \log \frac{P(C_i | I_T, r)}{P(C_i | I_T, \bar{r})} + \log \frac{P(r | I_T)}{P(\bar{r} | I_T)} \quad (12)$$

Following the language model ([11]), we now assume that 1) the irrelevance of all the target items is equal, and, 2) that cart C_i is independent of item I_k when event \bar{r} is given, i.e.:

$$P(C_i, | I_T, \bar{r}) = P(C_i | \bar{r})$$

Then the relevance rank becomes:

$$R_{I_T, \mathcal{P}_i} \propto \log \frac{P(C_i | I_T, r)}{P(C_i | \bar{r})} + \log P(r | I_T) \quad (13)$$

$$R_{I_T, \mathcal{P}_i} \propto \log P(C_i | I_T, r) + \log P(r | I_T)$$

We further assume that given that item I_T is relevant, the items in the cart C_i are conditionally independent from each other. This leads to:

$$R_{I_T, \mathcal{P}_i} \propto \sum_{q=1}^{Q_i} \log P(c_i^q | I_T, r) + \log P(r | I_T) \quad (14)$$

The relevance rank R_{I_T, \mathcal{P}_i} according to equation (14) is build from two components, a term involving the likelihoods $P(c_i^q | I_T, r)$, and a bias term that involves the prior probability of the relevance of item I_T . This expression for the relevance rank is equivalent to the language modelling ranking expression in information retrieval ([11, 1, 18]). Cart C_i acts as a query in information retrieval and item I_T resembles a document that needs to be scored. Hence, each item in the cart acts as separate query terms.

By applying the Bayes rule once more we can rewrite the relevance rank into:

$$R_{I_T, \mathcal{P}_i} \propto \sum_{q=1}^{Q_i} \log \frac{P(r | I_T, c_i^q)}{P(r | I_T)} + \log P(r | I_T) \quad (15)$$

This definition shows that, in order to make a recommendation, we need to estimate the prior relevance probability of the target item as well as the relevance probability of the target item when the target item is combined with another item. For reasons of clarity these probabilities are, from now on, denoted as:

$$\frac{P(r | I_a, I_b)}{P(r | I_a)} \quad (16)$$

where $a, b \in \{1, \dots, N\}$.

A naive way to get these estimates is to broadcast the user profiles $\mathcal{P}_{i,t}$ throughout the P2P network like [23] or using DHTs to map keys to profiles like [16]. Clearly, this is not efficient. In the following, we propose to build the relevance links in a dynamic way, minimizing the communication overhead between the peers.

2.3 Self-organizing Item-buddy tables

Users profiles provide the necessary information about the relevance between items ([13, 20, 9]). First, we propose a dynamic approach to update the relevance probabilities. Then, we introduce *buddy tables* that store these relevance links in a distributed way.

2.3.1 Dynamically updating relevance probabilities

If a user likes two items I_a and I_b , then this increases the relevance of item I_a with respect to item I_b (and vice versa). Consequently, the relevance of item I_a given item I_b can thus be calculated from the profiles of all users (see also, [9]):

$$P_t(r | I_a, I_b) = \frac{|cart(I_a \cap I_b)|}{M} \quad (17)$$

where $|cart(I_a \cap I_b)|$ is the number of times that items I_a and I_b appear in the same cart.

In a P2P network, it is not desirable to calculate the relevance between two items using equation (17) since the carts are distributed throughout the entire

network. We have found the solution in dynamically updating these relevances.

At a given time, the relevance between two items is updated according to:

$$\begin{aligned} P_t(r | I_a, I_b) \\ = P_{t-\Delta t}(r | I_a, I_b) + \Delta P_t(r | I_a, I_b) \end{aligned} \quad (18)$$

where $\Delta P_t(r | I_a, I_b)$ is the update of the relevance between two items from time $t - \Delta t$ to t . The update is only non-zero when there is a user that downloads one of the items I_a or I_b while that user in the past already expressed interest in the opposite item. Hence, relevance updates only occur when multimedia files are downloaded. The relevance update can thus be expressed in terms of the transactions that take place within $t - \Delta t$ to t :

$$\Delta P_t(r | I_a, I_b) = \sum_{\forall T_k: t-\Delta t < k < t} \Delta P_k(r, T_k | I_a, I_b) \quad (19)$$

where T_k is the transaction at time k , and $\Delta P_k(r, T_k | I_a, I_b)$ represents the relevance update between items I_a and I_b when considering transaction T_k at time k .

Since the relevance between I_a and I_b is not changed when considering a transaction that does not involve the downloading of either two items, the relevance update can be simplified to:

$$\begin{aligned} \Delta P_t(r | I_a, I_b) = \\ \sum_{\forall T_k: t-\Delta t < k < t} \Delta P_k(r, I_a = item(T_k), I_b \in C_{peer(T_k)} | I_a, I_b) + \\ \sum_{\forall T_k: t-\Delta t < k < t} \Delta P_k(r, I_b = item(T_k), I_a \in C_{peer(T_k)} | I_a, I_b) \end{aligned} \quad (20)$$

where $item(T_k)$ indicates the item being downloaded in transaction T_k and $peer(T_k)$ indicates the peer that performs the download.

Equation (20) shows that the relevance between two items can be updated using only the information about the item that is being downloaded and the cart of the peer that is downloading the item. Now we show how to store the relevance information between two items I_a and I_b in a distributed way.

2.3.2 Item buddy table

Equations (19) and (20) show that the relevance probabilities between two items I_a and I_b can be calculated incrementally. We can store these probabilities locally at the location of both items. This is realized by attaching to each item a so called *buddy table*. This buddy table stores the information (including an index to their location) about the *top-N* relevant items according to their relevancies with the buddy table item.

Updating strategy: The table is updated each time when the item is being downloaded by a peer. Then for all items in the cart of that peer the relevancies

with item being downloaded are updated. The relevancy between a cart item and the buddy table item are calculated based on equation (20).

Careful, investigation of equation (20) shows that when the two items I_a and I_b will appear in one cart (which automatically happens when a peer that has one of the items in the cart downloads the other), that the relevancy between both items needs to be updated. Or, in other words, that the relevancy between the two items need to be updated for both the buddy table of item I_a as well as the buddy table of item I_b .

Since the buddy tables store the relevance between two items I_a and I_b locally at both the items, and we would like to update *only* the relevancies within the buddy table of that item that is being downloaded (to minimize the communication), the updating strategy needs some adaptation.

To enable that we *only* need to update the buddy table of the item that is being downloaded, we make the assumption that the *order* in which items are being downloaded is arbitrary. This implies that for the increase in the relevance between the two items I_a and I_b , it does not matter whether I_a is downloaded by a peer with I_b in the cart, or vice versa:

$$\begin{aligned} \Delta P_k(r, I_b = item(T_k), I_a \in C_{peer(T_k)}|I_a, I_b) = \\ \Delta P_k(r, I_a = item(T_k), I_b \in C_{peer(T_k)}|I_a, I_b) \end{aligned} \quad (21)$$

It can be easily proved that both two terms in the above equation are equal to $(1/2) \cdot (1/M)$.

$\Delta P_t(r|I_a, I_b)$ in equation (20) can then be rewritten as:

$$\begin{aligned} \Delta P_t(r|I_a, I_b) = \\ \sum_{\forall T_k} 2 * \Delta P_k(r, I_a = item(T_k), I_b \in C_{peer(T_k)}|I_a, I_b) \end{aligned} \quad (22)$$

In other words, the relevance between items I_a and I_b can be stored locally at item I_a (denoted as R_{I_b, I_a}). Similarly, this relevance between the two items is also stored locally at the buddy table of I_b (denoted as R_{I_a, I_b}). Then the relevance is updated each time a peer with I_a in the cart is downloading I_b :

$$\begin{aligned} \Delta P_t(r|I_a, I_b) = \\ \sum_{\forall T_k} 2 * \Delta P_k(r, I_b = item(T_k), I_a \in C_{peer(T_k)}|I_a, I_b) \end{aligned} \quad (23)$$

Caching: For efficiency reasons during the recommendation, only the *top-N* highest ranked items are used to generate a recommendation based on the buddy table item. In practise, a longer list of items are stored to improve the recommendation accuracy.

Item availability: One of the characteristics of P2P networks is that peers are frequently not online. Consequently, the items stored at these peers are then also not accessible. The *top-N* highest ranked items

in the buddy table can be periodically screened for item availability so that no unavailable items will be recommended.

Replicas: One of the characteristics of P2P networks is that peers are frequently not online. Consequently, the items stored at these peers are then also not accessible. The *top-N* highest ranked items in the buddy table can be periodically screened for item availability so that no unavailable items will be recommended.

2.3.3 Recommendation

Using the relevance rank of equation (15), a recommendation can be generated as follows:

$$\begin{aligned} R_{I_T, \mathcal{P}_i} \propto & \sum_{q \in \{1, \dots, Q_i\} \cap I_T \in c_i^q.BT} \log \frac{R_{I_T, c_i^q}}{R_{I_T}} \\ & + \sum_{p \in \{1, \dots, Q_i\} \cap p \neq q} \log \frac{\partial}{R_{I_T}} + \log R_{I_T} \end{aligned} \quad (24)$$

where ∂ is the default ranking for the target item that does not present in the buddy tables and $c_i^q.BT$ denotes the buddy table of the item c_i^q . Since this is much smaller than the other relevance ranks it can be ignored. R_{I_T} denotes the overall relevance of the item I_T ($P(r|I_T)$). It is easily obtained by counting the downloading times ($R_{I_T} = \frac{|cart(I_T)|}{M}$). The final ranking for recommendation then becomes:

$$Rank_{I_T, \mathcal{P}_i} = \sum_q \log R_{I_T, c_i^q} - (|q| - 1) \log R_{I_T} \quad (25)$$

where $q \in \{1, \dots, Q_i\} \cap I_T \in c_i^q.BT$ and $|q|$ is number of the elements belonging to q .

3 Experiments

To validate our proposed self-organizing distributed collaborative filtering approach, we simulated a situation in which users exchange music files on a P2P network. This artificial data is generated from a centralized music playlist data set from the *Audioscrobbler*⁴ community. This data set is continuously updated and at the time that we captured it contained 1,862,766 transactions from 6,359 user IDs and 857,020 item IDs.

Pre-processing: The data set is strongly polluted and needs to be processed first:

- 385 redundant item IDs and 2903 inactive user IDs (users that once registered but never played any song) are removed.
- Items that were played by less than 20 users are removed. This was done since: 1) many titles of the songs are incorrect (since most of the time the title of the song is extracted from the name of the filename), and 2) 77.9% of the songs were only played by

⁴www.audioscrobbler.com

one user. When randomly selecting 100 songs played by only one user, 80% titles are wrong or odd. The percentage of the incorrect titles is extremely reduced when items played by more than 5 users are considered.

- Users that have played less than 2 items have been removed since their profiles do not add relevancies within the network (they have only one item in their cart).

After the pre-processing we are left with 475531 transactions from 3854 userIDs and 10869 itemIDs. The sparsity is 98.86%.

We randomly divided the data set into a training set (80% of the users) and a test set (20% of the users). We have used the training set to calculate the buddy tables, the relevance links and build the recommendations. The test set is used for evaluating the accuracy of the recommendations.

In the training set, there are 3067 users and 374530 transactions. To simulate the P2P network, each item is uniformly distributed across the different peers. Each transaction (play action in the data set) is labeled with a time index. As before, each transaction then represents a user (*userID*) that downloads an item (*itemID*) from another peer at the attached time (*t*).

In the test data set there are 767 users. The play actions of these users are used to test the accuracy of the recommendations. For each test user, 50% of the items of a test user were put into the cart of that user (the user profile). The other 50% of the items are used to test the recommendations. By doing so, the number of items in the carts of the users reflect the distribution in the overall data set.

3.1 Self Organizing Relevance Links

Each time a transaction takes place (i.e. when a multimedia file is downloaded) the relevance links in the buddy table of the item that is being downloaded are updated. The dynamic behavior of the relevancy between items can thus be studied. Figure 2(a) to (f) illustrate the links that are being created after: (a) 0, (b) 74.906, (c) 149.812, (d) 224.718, (e) 299.624, and, (f) 374.530 transactions.

The figure shows the songs (items) of nine artists that are selected such that they reflect different music genre and that they have a different amount of songs within the database. For each item (song) links to the top-5 relevant items (according to their buddy table) are displayed as directed arrows (pointing outwards the buddy table item). For reasons of clarity, only the links between the displayed items are shown.

From the figure, we observed that:

- The number of relevance links increases when the number of transactions increases.
- The relevance links converge and cluster between songs (items) of the same artists, which can be seen

from the large number of links between songs of the same artist that arise when the number of transactions increase.

To further measure this, we plot the percentage of links created among songs from the same artist as a function of the number of relevant links considered (top-N) and the number of transactions. Figure 3 shows these dependencies for two artists: *Avril Lavigne* and the *Beatles*.

This figure strengthens the observation that with the increase of transactions, the percentage of the links between songs of the same artist increases. It further shows that songs from the same artists are more relevant (have a better ranking position) than songs of other artists. This can be noted from the increase in the percentage of songs of the same artists in the top-N of the buddy tables of the songs of that artists when N is decreased.

- Figure 2 also shows that, besides relevance links between songs of the same artists, relevance links have been created between songs that have the same genre (reflecting the interest of a group of users). For instance, relevance links have been created between *U2*, *Radiohead* and *Nirvana*, groups that belong to the rock genre. Links between *Avril Lavigne*, *Pink* and *Shakira* may indicate a group of young female pop music artists. *Norah Jones* is somehow isolated since she belongs to the jazz genre, a different style compared to the other eight artists.

3.2 Recommendation performance

The recommendation performance is measured for the users in the test set. For each user, 50% of their items are put in their cart. Then for the other 50% of the items, the ground truth, a recommendation is performed. These recommended items are then compared to the ground truth items.

The performance is then measured using the *coverage* and *precision* (which are widely used information retrieval). The coverage measures the proportion of the ground truth items (known by the play-lists) that are recommended. The precision measures the proportion of the recommended items that are ground truth items. Note that the items in the cart of the test user represent only a fraction of the items that the user *truly* likes. Therefore, the resulting precision is smaller than the true precision.

The coverage and precision of the recommendation in the five iterations are shown in Fig. 4 (a) and (b). The results indicate that as the number of transactions increases, the recommendation results become better.

We compared our distributed collaborative filtering approach with the *Top-N suggest* recommendation engine, a well-known centralized collaborative filtering approach ([9])⁵. Both the item-based version as well the a user-based version were compared. The parameters were set according to the user manual.

⁵<http://www-users.cs.umn.edu/~karypis/suggest/>

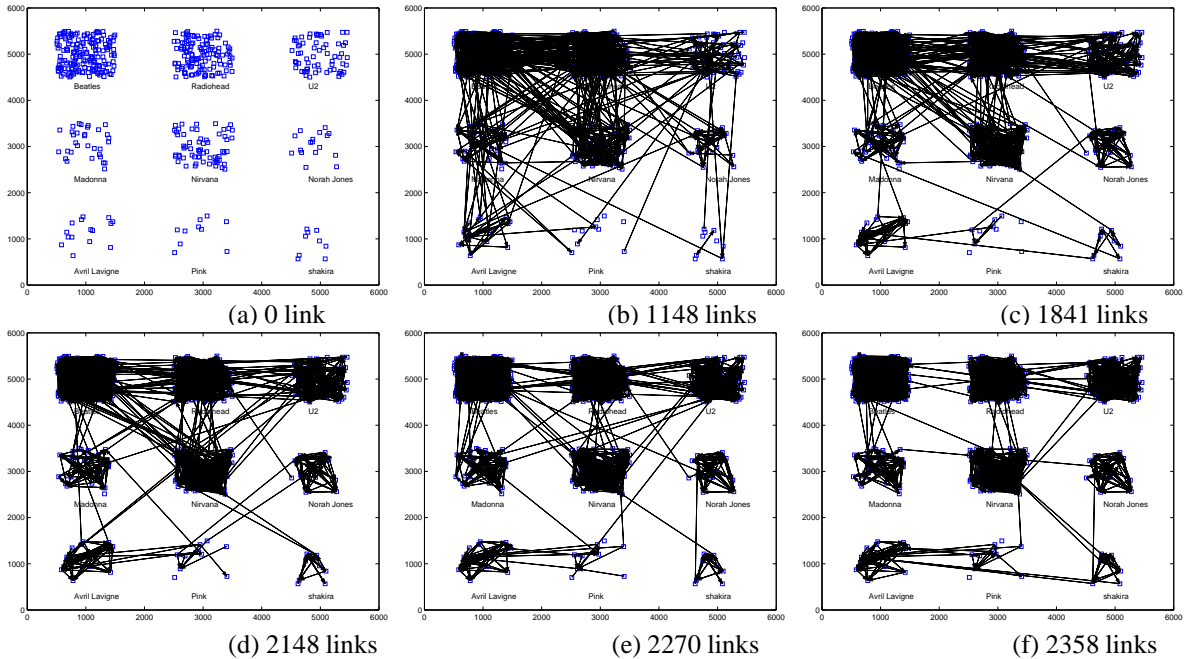


Figure 2: Illustration of dynamically created relevance links between the songs of nine artists. Each song is represented by a blue rectangle. Songs from the same artist are clustered within a grid, resulting in the nine rectangle regions. For clarity, the links to songs of other (than the nine showed) artists are removed. The panels show the relevance links after (a) 0, (b) 74.906 (iteration 1), (c) 149.812 (iteration 2), (d) 224.718 (iteration 3), (e) 299.624 (iteration 4), and, (f) 374.530 (iteration 5) transactions.

Additionally, we compared the three recommenders to a non-personalized recommendation approach. Here, for each item, its overall relevance $P(r|I_T)$ is calculated. The items are then ranked and recommended accordingly.

For our distributed approach, we show the results of two different settings: one with an item prior (second term of equation (15)) and one without the item prior.

For computational reasons, we randomly sampled the pre-processed data set to limit the number of users to 1300 users and the number of items to 4807.

We then randomly divided the sampled data set into 80% (1040) users in the training set and, the remaining, 20% (260) users in the test set. In the training set there are 159036 transactions. For our approach these are transactions are labeled with a time index and used to build up the relevance links stored in the distributed item buddy tables. For the centralized approaches these transactions are used to build a user-item rating matrix. For the test users (from the test set), 50% of the items are put into the cart of the user and the other 50% items act as the ground-truth.

Since the precision and coverage vary according to the number of recommended items, we adopt a normalized precision to compare the methods. Hereto, the precision is normalized according to the precision of a random recommendation of the same number of elements.

The normalized precision of the five methods are shown in Fig. 4 (c). It shows that the performance of

our distributed recommendation is comparable with the centralized methods. Our approach with the item prior outperforms even the centralized top-N user-based recommendation method and approximates the best top-N item-based method.

4 Conclusion

In this paper, we have proposed a self-organizing distributed collaborative filtering approach. This has been realized by introducing the concept of buddy tables that are attached to items that are distributed throughout a P2P network. Simulation experiments with music play-list data showed that our approach is a promising technique for searching songs by recommendation in a P2P network.

Other issues that need to be addressed in the future are cheating attacks ([12]), piracy ([3]), and trust ([7]).

References

- [1] A. Berger and J. D. Lafferty. Information retrieval as statistical translation. In *Research and Development in Information Retrieval*, 1999.
- [2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of UAI*, 1998.
- [3] J. Canny. Collaborative filtering with privacy via factor analysis. In *Proc. of ACM ICIR*, 1999.
- [4] J. Canny. Collaborative filtering with privacy. In *Proc. of the IEEE Symposium on Security and Privacy*, 2002.
- [5] A. Crespo and H. Garcia-Molina. Semantic overlay networks for p2p systems. Technical report, Comp. Sci. Dept., Stanford University, 2003.
- [6] O. D. Gnawali. A keyword set search system for peer-to-peer networks. Master's thesis, Massachusetts Institute of Technology, June 2002.

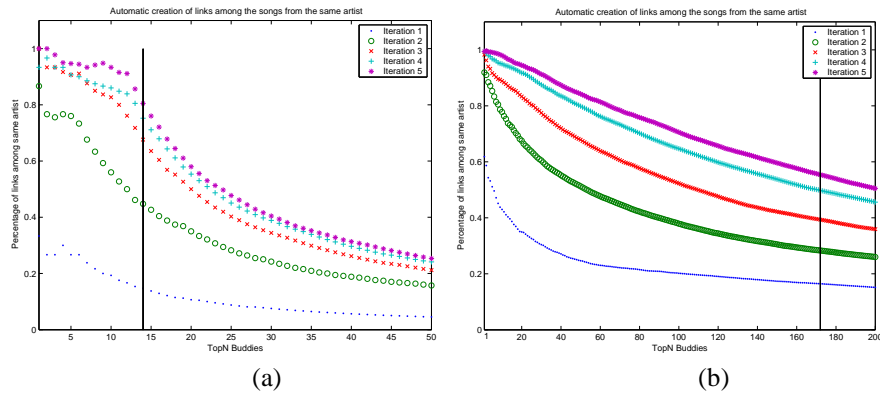


Figure 3: Percentage of relevance links towards songs of the same artist within the top-N ranked items in a buddy table for different settings of N and after a different number of transactions (iterations). (a) The average percentage of links between the 15 songs from *Avril Lavigne* within their respective top-N buddy tables. (b) Similarly for 173 items from the Beatles.

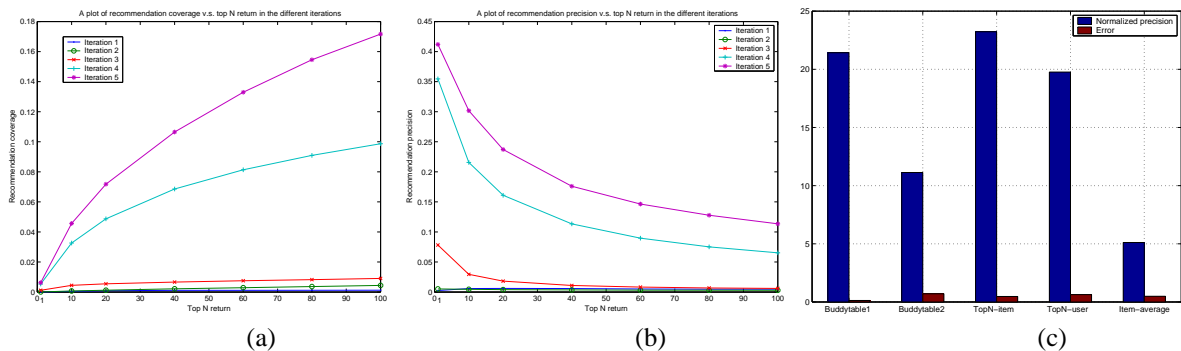


Figure 4: Recommendation results. (a) Coverage as a function of the N (top-N) most relevant items after training using the five different iterations. (b) Similarly for the precision. (c) The normalized precision for: (1) our proposed distributed collaborative filtering approach with the prior term of equation 15; (2) without the prior term; (3) the centralized item-based top-N suggest method; (4) the centralized user-based top-N suggest method; and (5) a reference recommendation method based on a average ranking.

- [7] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proc. of the WWW conference*, 2004.
- [8] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *Proc. of IJCAI*, 1999.
- [9] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proc. of the tenth international conference on Information and knowledge management*, 2001.
- [10] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.
- [11] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. *Language Modeling and Information Retrieval, Kluwer International Series on Information Retrieval*, V.13, 2003.
- [12] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *Proc. of the WWW conference*, 2004.
- [13] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, Jan/Feb.:76–80, 2003.
- [14] B. Marlin. Collaborative filtering: a machine learning perspective. Master's thesis, Department of Computer Science, University of Toronto, 2004.
- [15] D. Y. Pavlov and D. M. Pennock. A maximum entropy approach to collaborative filtering in dynamic, sparse, high dimensional domains. In *Proc. of the NIPS*, 2002.
- [16] H. Peng, X. Bo, Y. Fan, and S. Ruimin. A scalable p2p recommender system based on distributed collaborative filtering. *Expert systems with applications*, 2004.
- [17] J. Pisson and T. Moors. Survey of research towards robust peer-to-peer networks: search methods. Technical report, University of New South Wales, 2004.
- [18] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proc. of International Conference on Research and Development in Information Retrieval*, 1998.
- [19] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proc. of SIG-COMM*, Aug. 2001.
- [20] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. of the WWW Conference*, 2001.
- [21] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *Proc. of Infocom*, 2003.
- [22] I. Stoica, D. K. R. Morris, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. of SIG-COMM*, Aug. 2001.
- [23] A. Tveit. peer-to-paper based recommendation for mobile commerce. In *Proc. of the First International Mobile Commerce Workshop*, pages 26–29, 2001.
- [24] S. Voulgaris, A.-M. Kermarrec, L. Massoulie, and M. van Steen. Exploiting semantic proximity in peer-to-peer content searching. In *Proc. of the 10th IEEE Int'l Workshop on Future Trends in Distributed Computing Systems*, 2004.