Wi-Fi Walkman

Jun Wang
Delft University of Technology, Delft, The Netherlands

A.P. de Vries
Delft University of Technology, Delft, The Netherlands and CW, Amsterdam, The Netherlands

Marcel J.T. Reinders
Delft University of Technology, Delft, The Netherlands

Abstract
Wireless home entertainment center refers to a device able to handle heterogeneous media and to connect client devices located within the house and the outside world (i.e., the Internet).

INTRODUCTION

In recent years, with the rapid progress in information processing, communications, and storage technologies, the amount of information that we deal with in our daily lives has rapidly increased. Although we enjoy the entertainment and convenience brought to us by a variety of sources, the volume of information is increasing far more quickly than our ability to digest it. For instance, the World Wide Web is the most significant media source for most Internet users and is growing at an exponential speed. However, the ability of obtaining useful information grows slowly. Most importantly, the retrieval of information relevant to the user's interest remains an unsolved problem. Above all, the gap between the amount of information that is available and the information that people are able to extract is increasing.[1] Unfortunately, today's computers merely act as information providers. One of the solutions to close this information gap is to increase the ability of computers to steer the user's interests and select/represent relevant information on the user's behalf.

To this regard, the research on information filtering is aroused to filter out, refine and systematically represent the relevant information and intuitively ignore superfluous computations on redundant data. One of the solutions for overcoming the information overload is to provide personalized suggestions based on the history of a user's likes and dislikes. In the domain of human–computer interactions (HCIs), especially for the interface of e-commerce, the information overload has created increasing interests in recommender systems that recommend products such as books, CDs, movies, TV programs, and music.

Personalized Services

In human society, people are extremely experienced in person-to-person communication. People have a common understanding of each other both conceptually and perceptually, i.e., it is easy for them to obtain an understanding of each other in terms of the interests, tastes and expectations. Therefore, it is easy between people to provide different services. This, however, comes at a price: it costs a lot in terms of time and labor which severely hampers to cover required demands from large masses of people.

In recent times, more and more services are available in the form of HCIs, especially due to the increasing interest in Internet. For example, business-to-consumer (B2C) services in the e-commerce domain broadly extend the range of the traditional services and provide a convenient and low-cost way to deliver services to a large group of consumers. For instance, people can purchase books, CDs, electronics and other items from the Web anytime, anywhere, without the need to go to the different shops accordingly.

When compared with the person-to-person service, the current computer-based service is not operating in a friendly manner. The computer just acts as the information provider and provides the necessary transactions. The computer does not know the interest and intent of the individual user and therefore is only able to supply services of a general nature, i.e., not adapted to the specific customer that it is dealing with. As a result it may damage the quality of the service (in contrast with the person-to-person communication) when there is a large amount of options and the user is overloaded with this information and is not able to make an instant decision. To this end, it is necessary to develop methods that allow the computer to

Encyclopedia of Wireless and Mobile Communications DOI: 10.1081/E-EWMC-120043607

infer the user's interest or intent such that it can provide personalized services. This will eventually support companies to realize a shift from offering mass products and services to offering customizing goods and services that efficiently fulfill the desires and needs of individual customers.[2] Obviously, recommender systems can be one of the solutions to such a shift.

Folk Computing

Personalization is sited in the folk computing environment. Original computing environments were designed for scientists and intensive training was needed before users were able to use them. The recent progress in the information and communication technology (ICT) has supplied a pervasive or ubiquitous computing environment. In this environment, the computer user need not necessarily be the "mouse-clicker" in front of the desktop anymore. The target user in the folk computing environment is the common user. The interactions between a human and a computer in this environment are required to be natural: "users apply their senses to observe data and information of interest related to an event (conceptual and perceptual analysis) and interact with the data based on what they find interesting."[1]

In the folk computing environment the central issue is to realize a natural interaction between user and computer. For example, one of the important design strategies is "what you see is what you get." Another strategy is that the interface should be more compelling and natural and less intimidating to people than a keyboard and mouse. This could be achieved by making use of multimedia (audio, images, graphics, video, and touch) and multiple sensors (camera, motion detector, voice capture, GPS, etc.). Therefore, in this environment, personalization should be based on this natural interaction instead of the traditional "mouse-click" styled interaction.

Moreover, for personalization, one of the benefits in this environment is that having the possibility of multimodal input devices will make it possible to infer the intent of a user through such sources, e.g., the emotional state of a user can be interpreted from recognizing his/her expressions from video recordings. Having information about the intent of the user also opens up the possibility to react on this intent, e.g., by recommending the desired services to the user.

Peer-to-Peer Networks

Not only the availability of the sufficient types of the information, but also the way people access information is changing. Peer-to-peer and ad hoc networks, as new network topology, become a new way for people to distribute, exchange, and consume resources from their local storage devices in many different locations, such as the future home, office, or university campuses. There are two significant advantages of peer-to-peer and ad hoc networks: 1) the replicas of the content among peers increases the content availability; 2) for the exchange of information, no centralized storage and management from third parties is necessary, which makes these networks very low-cost. In recent years, these attributes have attracted a large body of people in the Internet domain. For instance, Internet based peer-to-peer networks have increased rapidly and they have given a large number of people the possibility of sharing resources in their local storage devices.[3,4] Recently, sharing resources in wireless networks has received some attention. The TunA system[5] allows users to "tune in" to other nearby TunA music players and listen to what someone else is listening to. Another system, SoundPryer[6] allows drivers to jointly listen to music shared between cars on the road. Interestingly, these two applications show that the upcoming technologies have started to care about their social impact on everyday life, i.e., they bring people together that have been socially separated by the technologies for the last decades (such as TV, Internet, portable music player, etc.) Clearly, these technologies[3–6] are different from the traditional technologies in that they encourage people to make social interactions such as sharing and exchanging information. However, those applications are implemented on devices that are far away from so-called intelligent devices which aims to provide personalized services on user's behalf. We present here a different system that has the ability to react to the user's interests and select relevant information on the user's behalf accordingly.

In ad hoc network environments, the volume of information is increasing far more quickly than our ability to digest it. The traditional textual keywords-based information retrieval approaches[7–10] can no longer be used as filter mechanisms since they suffer from three major problems. First, the transition from textual data to heterogeneous data requires large amount of textual metadata on the one hand. It is practically intractable to ask people to provide content as well as associated metadata at the same time. On the other hand, automatic content analysis on the non-textual data is far from being efficient to get the metadata that we need. Second, keywords are not semantically expressive enough to enable a seamless search, i.e., people hardly issue a textual query when they cannot exactly express what they are looking for. Thirdly, in mobile environments, the user interface is constrained and consequently does not permit complex interactions between users and their handheld devices.

WI-FI WALKMAN

The Wi-Fi walkman that we developed is a case study that investigates the technological and usability aspects of

HCI with personalized, intelligent and context-aware wearable devices in ad hoc wireless environments such as the future home, office, or university campuses. It is a small handheld device with a wireless link that contains music content in the environment or from the user. Users carry their own Wi-Fi walkman around and listen to the music content. All this music content can be shared using mobile ad hoc networking. The Wi-Fi walkman is situated in a peer-to-peer environment and naturally interacts with the users. Without annoying interactions with users, it can learn the users' music taste and consequently provide personalized music resources to fit the user's interest according to the user's current situated context.

Music Recommendation

In the Wi-Fi walkman scenario, the multimedia content data the user intends to access are music files (MP3 formatted). Those music files are possible stored in the hard disk of each Wi-Fi walkman and can be accessed through the Wi-Fi mobile network. Users are able to share music content through the network. However, as the network size increases, the music content available is increasing as well. This consequently causes an information overload problem. To address this, in this scenario, music recommendation is implemented as a user oriented music file filter to help user to find relevant or desired music files according to current situated context and learned user interest.

Scenarios

We now discuss detail descriptions of some possible concise scenarios for Wi-Fi walkman:

A business man called Frank is a music fan. He has just bought a Wi-Fi walkman attached with a personalized music recommender system (MRS). This personalized MRS can recommend music files (such as play-lists) to Frank based on his interests (profiles) and the context anytime anywhere.

***Scenario* 1 (*During jogging in the morning*).** As usual, Frank, bringing along with his favorite Wi-Fi walkman, is jogging in a nearby park. Due to the fact that the mobile network in the park area does not have good quality, Frank's Wi-Fi Walkman may not download music in this area. However, since Wi-Fi walkman knows Frank usually enjoy sport music during this time. The MRS knows Frank's long-term interest (profile) and the current situation (that Frank is engaged in sports and the network quality is poor). So the system has already pre-cached and recommends a bunch of music (Frank's favorite sport music) which best fits Frank's interest and current situation.

***Scenario* 2 (*During a trip*).** Frank with his friends joins a tourist group to a church. He switches on his Wi-Fi walkman and asks the MRS to recommend some music to fit this environment. The MRS knows he is in the church by communicating with both the situated network and his friends' Wi-Fi walkmans and consequently recommend some church music fitting Frank's favorites. By using his Wi-Fi walkman, Frank enjoys a complete church experience.

***Scenario* 3 (*Before sleeping*).** Frank usually sleeps at 12 midnight. The Wi-Fi walkman knows his schedule. At 12 midnight, the system finds and recommends desired music from the Internet. Since Frank been having trouble recently getting to sleep, the system recommends some light music to help Frank go to sleep.

Problem Definition and Formalization

From above mentioned scenarios, we can simply articulate the problem as:

According to user's interest or taste and current situated context, the system recommends the appropriate music service to the user.

In this definition, some factors need to be clarified.

Interest

Many aspects affect the interest of a user. If we treat interest as a whole, it is difficult to grasp the latent patterns behind. Hence, we classify the user's interest into long-term interest and short-term interest.

Long-term interest is the user's preference or taste. It evolves slowly and smoothly as the user experiences and socially interacts with other people and the outside world. We assume it is comparatively static and diverse. On the contrary, short-term interest is the current intent or task at hand. It evolves sharply based on the context and the user's willingness. It is not stable but focused. If we model the two types of interest differently, it could help us to accurately understand the user's interest.

Context

In the definition, there are two important factors that need to be considered: user's interest or preference and context. Context plays an important role to understand the user's current short-term interest or task.

Services

Here we would like to state that, rather than the music/song itself, the service of the music should be the target item. This is because we are in a pervasive computing environment (in particular owing to the foreseen ad hoc network). The deliverability and quality of the service are also important factors to be taken into account. For instance, as shown in the first scenario, if the recommended music is hard to reach, the system may pre-cache it at a suitable time. This extends the system to consider the service rather than to provide only the content.

Resources to Build up User Preferences

To provide personalized content services, the starting point, clearly, is to understand the user's interests and/or preferences. The more information we have about the user and the content itself, the better we know what the user wants in a certain context. There are three channels of information to acquire such information: the people-to-people correlation, the music-to-music correlation, and the demographic data of the users.

The people-to-people correlation information

The people-to-people correlation information reflects the correlations among people's "tastes" for the music. A collaborative filtering-based recommender system is able to recommend music to a user based on the other users in the system who have similar "tastes." The music play-list of a user reflects the taste of that user. The information in the play-list could be useful including song's name, playing times, playing frequency etc. Initially, we will utilize a dataset in the AudioScrobbler community.[3] Currently this dataset has 857.020 tracks and 4.175.146 playback actions.

Since collaborative filtering has been widely investigated, there are other datasets available but in other (than music) domains. For instance, the movie lens research group at the University of Minnesota provides two ratings datasets (Each movie and Movie lens). Numerous collaborative filtering publications have employed those datasets for evaluations. If necessary, we will use these "standard" datasets for evaluation purposes.

Music-to-music correlation information

Correlation between different pieces of music can be obtained by analyzing the content description of those music pieces. Automatic content analysis is, however, still an unsolved problem. On the other hand, manual annotation is a time-consuming and annoying task.

Fortunately, a community, called MusicBrainz, provides a music meta-database of the content description of music songs purpose. Some learning methods can be applied on this content description to find the correlations among music.

In addition, for the new MP3 files which are not annotated in the database, we can use acoustic fingerprinting (FFT transformation compressed to a few bytes), unique to each piece of music to find the correct metadata for them as well.

Demographic data

Demographic data about users (such as age, gender, social position etc.) is useful for categorizing users. This data could especially be useful to solve the start-cold problem, i.e., providing recommendations when the system does not yet have any information about the user ratings or play-lists. The demographic data allows that users to be compared. However, user's demographic data need to be collected from other applications.

RECOMMENDER SYSTEM FOR WI-FI WALKMAN

There are two approaches for implementing recommendation, namely data-driven, and rule-based or knowledge-driven approaches. Both of them have to understand user beforehand either explicitly given or implicitly learned. Data-driven approach achieves recommendation by observing the behavior of the user and learning patterns in this behavior while the rule-based/knowledge-driven approach implements recommendation where a user defines his/her likes/dislikes through questionnaires.

Recommender system is a popular form to generate personalization. It is employed in the e-commerce domain (Web-based shops) to create personalized environment for selling items such as books, CDs. In our research here, we also focus on the recommender system as a form of personalization.

Definition

A recommender system can be defined as:

A system which "has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options."[11]

This definition makes it clear that user oriented guidance is critical in a recommender system. This means that, during the interactions between a human and the computer, the computer needs to provide not only the information but also the guidance toward that information on user's behalf.

Characteristics of User and Product Information

To be able to perform a personalized recommendation one needs to understand the user's interest or preference. Hereto we need to acquire and analyze information about the user. On the other hand, we also need information about the products to be customized.

Generally, the information about the user and the products has the following characteristics:

1. It comes in tremendous volumes;
2. It is dynamic, i.e., it varies over time;
3. It may be continuous in time, i.e., streaming information;
4. It does not exist in isolation, i.e., it exists in its ambient context with other data;
5. It is inherently heterogeneous, i.e., it is collected or sensed from a set of distinct sources.

For example, multimedia (audio, images, graphics, video, and touch) and multiple sensors (camera, motion detector, voice capture, GPS, etc.) are usually employed to furnish the computing environment for more compelling and natural and less intimidating to people.

1. The quality varies (greatly).

When designing a recommender system that obtains relevant information from the user and product information, we need to take these characteristics info full consideration.

Research Issues

Clearly, there are numerous issues related to recommender systems. Here, we would like to mention some of them which do exist in our domain and are of particular interest to us. In addition, we also clarify some research issues which have not been stated clearly in literature.

Context awareness (task focus)

Human perception is greatly aided by the ability to probe the environment through various sensors along with the use of the situated context. In return, the context has a large influence on the interest and intent of one particular user. This causes the interest and intent of a user to vary dynamically over time. Thus, knowing the current context of the users is critical to correctly understand the interest and intent of a user. Context awareness is thus a major factor when dealing with personalization and recommender systems.

The user's preference is determined by both the general taste of the user (long-term taste) and the current task of the user (his/her context or short-term taste). Ignoring one of them reduces the quality of the recommendation considerably. When exploiting this context in the recommendation, there are two major problems: 1) the determination of the current context; and 2) the integration of this context with the general taste.

It is certainly not trivial to acquire information about the current task of the user. Clearly, this information can be revealed either implicitly, i.e., derived from other services contacted by the user, or explicitly, i.e., when the user indicates his/her current task (e.g., defining the task type using a menu-driven mechanism).

When information about the current context is available it still should be combined with the general taste of the user. One simple approach could be that the general taste recommendation is filtered based on the information about the context, or the other way around. We will address the question of how both types of information should be combined such that an optimal and efficient recommendation can be provided to the user.

Proactive resource caching

Within an ad hoc mobile network the availability of resources is not guaranteed. The recommendation system should take this into account, either 1) by incorporating the availability of the data into the recommendation engine; or 2) by predicting near-future recommendation so that the necessary data can be pre-cached. Both forms of adaptation change fundamentally the way the recommendations are done.

Adaptability

The interest or taste of a user may change over time. A recommender system should be aware of this change and consequently adapt the recommendations accordingly. When the change in the ratings of a user are known, collaborative filtering is quite capable of adapting to these changes since the position of the user in the rating space changes and consequently the recommendations change. However, this requires that we know the change in the ratings of the user. Thus we need some kind of feedback mechanism to let the system know the changes in the user's tastes. One way is to ask the user feedback on the recommendations made. Clearly, this is not a desirable way. Therefore, suitable and user friendly feedback mechanisms should be developed.

Sparsity

Since the amount of items is extremely large and most users do not rate most items, the matrix for measuring the people-to-people correlation is typically very sparse. Therefore, there is no guarantee of finding a set of neighbors who have similar taste. This typically happens when the ratio between the number of items to the number of users is very high or when the system is in the initial stage of use.

Some potential solutions include making use of the content descriptions or the, demographic data. These, and possibly others, should be investigated with respect to the music recommendation scenario.

Scalability

The standard collaborative filtering approach needs to know the user-to-user correlations based on the ratings. However, correlations need to change when new users are added. Therefore the computation of these correlations has to be on-line. To this end, using collaborative filtering to generate recommendations is a computationally expensive task. The nearest neighbor algorithm that is used in traditional collaborative filtering requires an amount of computations that grows with both the number of users and items. The algorithms which achieve fast results do not guarantee computationally efficient results when they are applied on large practical datasets.

To deal with this scalability problem, some solutions are proposed. For instance, one can reduce the data size by exploiting dimensionality reduction techniques such as principal component analysis (PCA)[12] or singular value decomposition (SVD).[13] As these methods approximate the data, they have the side-effect of reducing the recommendation quality.

As an alternative solution, instead of computing the user-to-user correlations on-line, item-based collaborative filtering, which computes item-to-item correlations (based on ratings as well), has recently been proposed to improve the scalability.[14,15] Due to the fact that the correlations between items are relatively static, they can be computed off-line. Therefore, the item-based collaborative filtering approach could make most of the computation off-line. This intuitively improves the scalability in large datasets.

Cold-start problem

The cold-start problem[16] is one of the common difficulties in a collaborative filtering-based recommender system. This problem can be divided into user cold-start problem and item cold-start problem. Since the correlations are obtained by the ratings, the algorithm fails where there are no/less correlations available. User cold-start problem happens when there is a new user in the system on whom no or few rating information is available. The collaborative filtering method then does not have enough information to reliably estimate a similarity (correlation) between users so that only poor or even no recommendations can be made. Similarly, the item cold-start problem occurs when there is no (or few) rating information on a new item. Then the similarity estimates between items is very inaccurate.

As already mentioned, correlations coming from the content information (for instance, the item-to-time correlation regarding to the content) link the old items and new items and intuitively provides some solutions to the item cold-start problem. For instance, when a new item is added, there is no rating information available about this item. Collaborative filtering cannot recommend this item to the users. However, by knowing the content information of the new item and the old items, the correlations between them can be built and we can actually recommend the new item to the users who like the highly correlated old items.

Meanwhile, demographics of a new user can categorize the user into some classes and as well correlate the existing users to the new user. Those findings could help us for the further research on this issue.

A Basic System

The Wi-Fi walkman is implemented on the Sharp Zaurus PDA (personal digital assistant), see Fig. 1, by using C++. It runs on an ad hoc wireless network. It features audio playback, audio storage, audio recommendation, and ad hoc wireless connectivity for audio exchange.

The Wi-Fi walkman itself contains an audio agent, a transport agent, and a wireless interface shown in Fig. 2. The audio agent is responsible for the communication with the recommendation services, manages the MP3 files on the storage devices (e.g., a fresh card), and selects which MP3 to play. The transport agent uses the wireless ad hoc network to communicate with other transport agents and enables the sharing of the music files. Due to the dynamic nature of an ad hoc network, the transport agents must keep track of the other walkmans around them. The enhanced ad hoc wireless interface also informs the transport agent of new walkmans and walkmans that can no longer be reached.

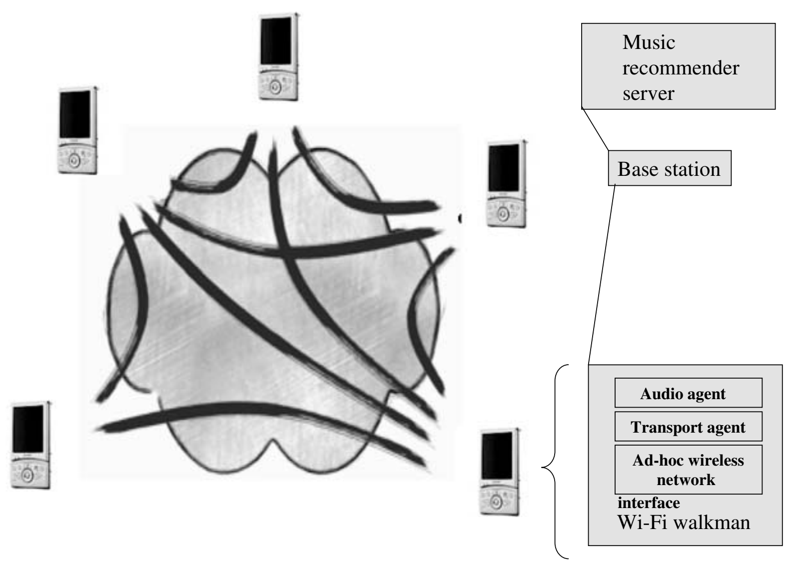

Fig. 1 Illustration of the Wi-Fi walkman in client/server model.

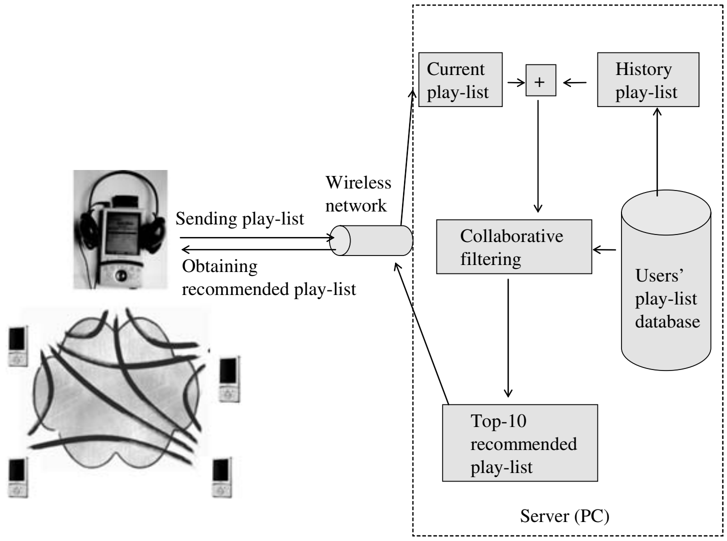

Fig. 2 Recommendation in the client/server model.

Peers and play-lists

Each peer represents a Wi-Fi walkman used by a particular user. Let's define the set of peers as:

$$P_i, i = \{1, \ldots, M\} \tag{1}$$

where M is the number of the peers currently on-line in the peer-to-peer network. That means they can be located and accessed with the available bandwidth. Since the peers (Wi-Fi walkman) and users exist in pairs, we will use the term peer and user interchangeably.

The music content in the network is defined as a set of items, denoted by the set I. Each item has a specific physical location, i.e.,

$$I = \{I^{i,j} \mid i = \{1, \ldots, M\}; j = \{1, \ldots, N_i\}\} \tag{2}$$

where N_i is the number of items physically located in the local storage device by the peer P_i. $I^{i,j}$ denotes the j-th item owned by user P_i. The set of items owned by peer P_q is denoted as:

$$I_q = \{I^{i,j} \mid i = q; j = \{1, \ldots, N_q\}\} \tag{3}$$

Users will retrieve music content according to their own interests. At a particular time, a user, however, will have a particular interest. The interest can be obtained either explicitly or implicitly. For instance, it could be explicitly obtained by asking users to rate items. Alternatively, this can also be implicitly indicated by the music items that the user is playing. In our Wi-Fi walkman, we assume the user's music play-list to be indicative of the user's music interest. Formally, we use a vector $V_q = \{v_q^{i,j}\}$, $i = \{1, \ldots, M\}; j = \{1, \ldots, N_i\}$ to represent the play-list of the user P_q, where the element $v_q^{i,j} = 1$, if user P_q played the item $I^{i,j}$; otherwise $v_q^{i,j} = 0$.

It may be noted that generally the interest of the user will change over time. It in fact depends on the current context. Therefore, the play-list (representing the current users' interest) should ideally be dependent on the time also, i.e., $V_q \rightarrow V_q(t)$.

We utilize a sliding time window to ignore the old music items users have played, as shown in Fig. 2. By doing so, the system focuses on the user's current interest.

The current recommender system is implemented by using the collaborative filtering technique. Collaborative filtering utilizes the correlations (commonalities) between users on the basis of their ratings (in this case, the play-lists of the users) to predict and recommend music items which have the highest correlations to the user's preference.

The accuracy of the collaborative filtering directly relies on the number of users, who provide their ratings. In mobile networks, the density of peers may vary strongly depending on the local situation. For instance, on a bus, there are only a dozen people while at an airport there are thousands. Depending on the current density of peers, we perform recommendation by two different approaches, namely the flooding model and the client/server model.

Flooding model

When the density of peers is large (i.e., thousands of users) and the play-lists from those users are enough to obtain a good recommendation, we use the flooding approach to find the correlations between users.

By using the correlation,[17,18] the similarity between the play-lists V_q and V_p of two users is calculated as follows:

$$\mathrm{Sim}(V_q,V_p) = \frac{\sum_{i,j}^{M,N_i}(v_q^{i,j}-\overline{v}_q)(v_p^{i,j}-\overline{v}_p)}{\sqrt{\sum_{i,j}^{M,N_i}(v_q^{i,j}-\overline{v}_q)\sum_{i,j}^{M,N_i}(v_p^{i,j}-\overline{v}_p)}} \tag{4}$$

where $\overline{v}_q$ and $\overline{v}_p$ are the mean rating of the user P_q and P_p respectively, that are used for removing the bias.

$$\overline{v}_q = \frac{1}{\sum_i^M N_i}\sum_i^M\sum_j^{N_i} v_q^{i,j}, \overline{v}_p = \frac{1}{\sum_i^M N_i}\sum_i^M\sum_j^{N_i} v_p^{i,j} \tag{5}$$

The distance measurement between a music item $I^{i,j}$, not known to user P_q, and the play-list from user P_q can be calculated as the weighted average rating,[17,18] as follows:

$$\mathrm{d}(I^{i,j},V_q) = \overline{v}_q + k\cdot\sum_{\{V_p|V_p\in N_q,\mathrm{sim}(V_q,V_p)>T\}} \cdot\,\mathrm{sim}(V_q,V_p)(v_p^{i,j}-\overline{v}_p) \tag{6}$$

where k is a normalization constant. In the flooding model, the play-list V_q of the user P_q is broadcast to all its neighbors P_p to determine the recommendation for that user. The neighboring peers check the similarity (using in Eq. 5) between the received play-list and their own play-list. They decrease the TTL (time to live) field of the broadcast play-list and then pass it to their neighboring peers until the TTL count reaches 0. We use set N_q to denote all the neighboring peers that the querying play-list V_q can reach. If one of the neighboring peers has a play-list that has a similarity to the broadcasted play-list that is higher than T, then the items in the play-list of the neighbor P_p (including the locations) are sent back to the peer P_q that posed the query V_q. We use I_q^* to denote the set of these returned items. Finally all items I_q^* received by the querying peer are ranked according to the distance measurement (Eq. 6) and consequently the top-N ranked items are recommended to the user (Eq. 7).

$$\mathrm{Rec}_q^N = \mathrm{Top}N\{\mathrm{rank}\{\mathrm{d}(I^{i,j},V_q)|I^{i,j}\in I_q^*, I^{i,j}\notin I_q\}\} \tag{7}$$

Client/server model

When the density of the peers is small and consequently the play-lists (rating) from those users are not enough to obtain a good recommendation, we have to access a predefined rating database and use that to calculate the recommendation. In this model, we assume the peer has a chance to access a server which has a rating database. The rating database stores the play-lists of all the users in the networks.

Fig. 3 illustrates the procedure of obtaining the recommended play-list. In order to reduce the computational complexity, we apply the item-based recommendation algorithm proposed in Refs. 14, and 15 to calculate the recommendations.

In item-based recommendation, each music item can be represented by who has played it. More formally, each item $I^{i,j}$ can be represented by a vector $U^{i,j}$, where its element $u_q^{i,j} = 1$, if the item $I^{i,j}$ has been played by the peer P_q and zero otherwise.

Item-based recommendation is then performed by exploring the correlations between the items rather than the correlations between users. Recommendations are created by finding items that are similar to other items that the user prefers according to:

$$\mathrm{sim}(I^{i,j},I^{i',j'}) = \frac{\mathrm{Freq}(I^{i,j},I^{i',j'})}{\mathrm{Freq}(I^{i,j})\times\mathrm{Freq}(I^{i',j'})} \tag{8}$$

where Freq($I^{i,j}$) is the number of times that item $I^{i,j}$ is in any of the play-lists. Freq($I^{i,j}$, $I^{i',j'}$) is the number of times that item $I^{i,j}$ and $I^{i',j'}$ are in the same play-list.

Due to the fact that the item-to-item matrix is relatively static, it is possible to compute this matrix off-line, which extremely reduces the computational demands. That is, by applying Eq. 8, for each item $I^{i,j}$, its top N similar items can be obtained off-line and it is denoted as $I_q^{\mathrm{Top}N}$.

When the play-list V_q of user P_q is sent to the server, the recommendation then is calculated according to the following equation:

$$\begin{aligned}\mathrm{Rec}_q^N = \mathrm{Top}N\{&\mathrm{rank}\{\mathrm{sim}(I^{i,j},I^{i',j'})\,|\,I^{i',j'}\\ &\in I^{i,j}_{\mathrm{Top}N}, I^{i,j}\notin I_q; I^{i,j}=1\cap I^{i,j}\\ &\in V_q\}\}\end{aligned} \tag{9}$$

Implementation

The recommendation is implemented in the server part. We utilize a dataset of the AudioScrobbler community[3] as our play-list dataset. Currently this dataset has 857,020 tracks and 4,175,146 playback actions. The interaction between each peer and the server is illustrated in Fig. 4.

Snap-shots of the Wi-Fi walkman application are shown in Fig. 5. The procedure to obtain the music files

$$\ldots I_i^{10}(t-T-1), I_i^{11}\underbrace{(t-T),\ldots,I_i^{12}(t-2), I_i^{13}(t-1)}_{\text{Time window}}, I_i^{14}(t) \qquad \text{Play sequence}$$

Fig. 3 Time window for forgetting.

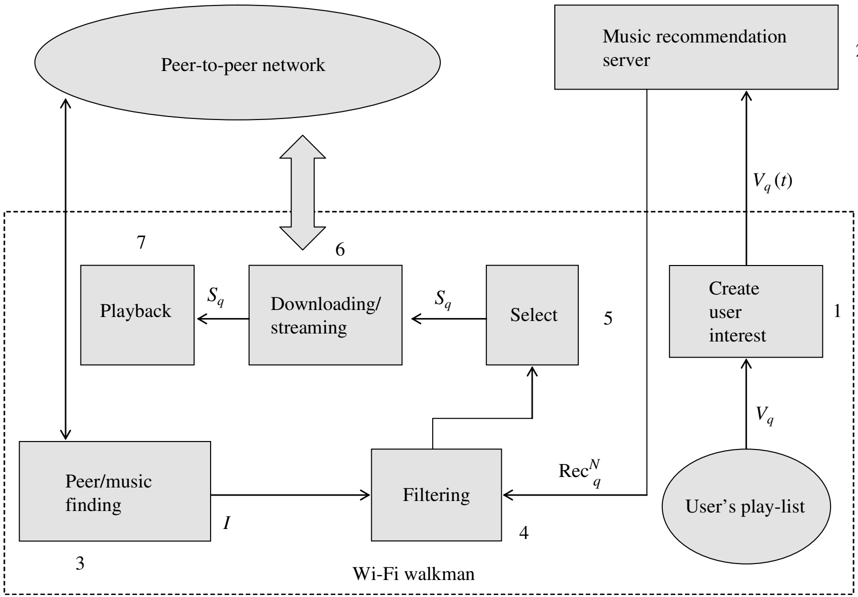

Fig. 4 System diagram of the Wi-Fi walkman application.

that fit the user's interest is illustrated in Fig. 4 and each step is described in the following flowchart:

Wi-fi_walkman()
Begin

1. Create $V_q(t)$ to represent the user's current interest from the play-list by utilizing a time window
2. Get recommendation Rec_q from server
3. Find on-line peers and obtain the music item list I (resources) from those peers
4. Filter the music list I to get the recommended list Rec_q by the top N recommended items Rec_q^N.
   $Rec_q = I \cap Rec_q^N$
1. Select the downloading/streaming items by users through GUIs $S_q \subset Rec_q$
1. Locate the recommended items S_q and download/ stream them
2. Playback the obtained items S_q

End.

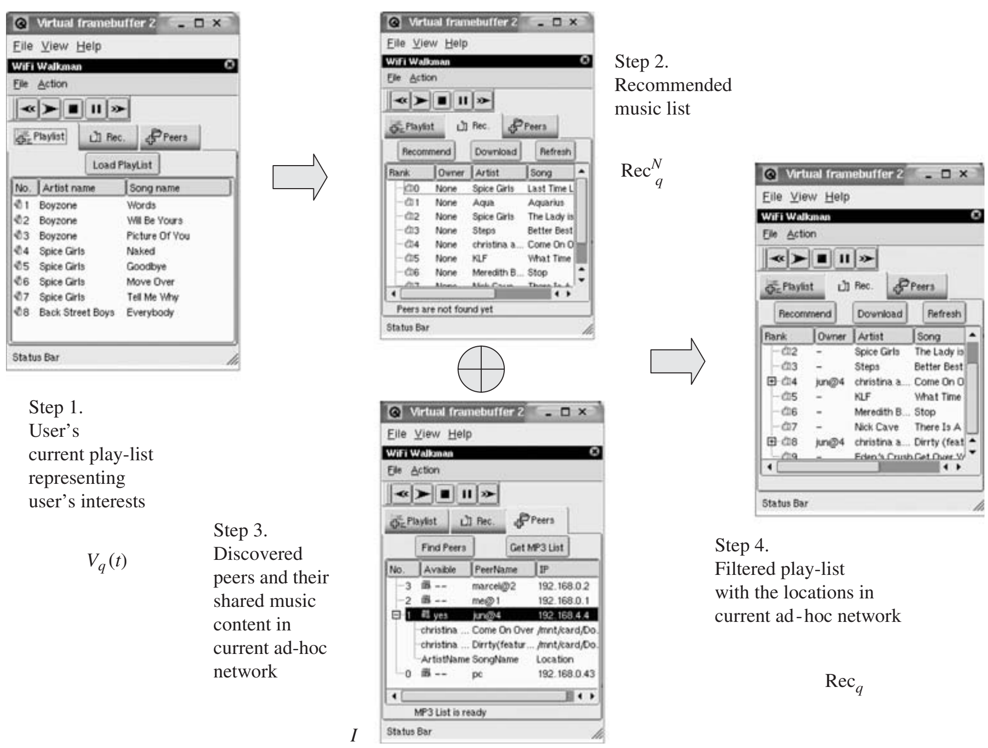

Fig. 5 Snap-shots of the Wi-Fi walkman prototype.

STATE-OF-THE-ART IN RECOMMENDER SYSTEMS

In this section, we will review the state-of-the-art of commercially available recommender systems as well as existing research solutions. Since most of the methods are not domain-constrained and can be further generalized, this review is not constrained to music recommendation. Other recommender systems for movies, books, CDs, TV programs, etc. as well as the general recommender system in the e-commerce field are included as well.

Commercially Available Recommender Systems

Recommender systems are best known for their use in the e-commerce domain. Here, it is employed to furnish personalized environments for selling items such as books, CDs, etc. Many on-line retailers utilize recommender systems, like, e.g., Amazon, CDnow, BarnesAndNoble, IMDB, etc. It is greatly successful as the appeal of personalized content created by recommender system exceeds those of untargeted content such as banner advertisements and top-seller lists which are usually used on the Web. This success has also boosted a number of successful start-up companies like Firefly Net Perceptions, LikeMinds and ChoiceStream to provide recommending solutions.

Amazon.com

Based on item-to-item collaborative filtering, Amazon.com developed a practical book recommender system.[19] Rather than the traditional user-based collaborative filtering (matching current user to similar customers), the item-to-item (item-based) collaborative filtering approach matches each of the items a user purchased and ratings to other similar items in the database and then constructs a recommendation list based on those similar items.

In the Amazon.com homepage, users can obtain recommendation lists based on the items in their shopping carts and can also filter out their recommendations by product line and subject area based on previous rates and purchases.

For large on-line retails, the two important factors for designing a recommender system are: 1) the scalability with respect to the (large) number of customers as well as items; and 2) the ability for real-time processing. In contrast to the similarity matrix of users-to-users, the matrix of items-to-items can be computed off-line. The item-to-item collaborative filtering method in Amazon.com has proven to produce recommendations in real-time and scales to massive datasets.

The MyBestBets personalization platform

ChoiceStream, Inc., a software development company headquartered in Cambridge, Massachusetts, developed The MyBestBet personalization platform. This platform provides a personalization solution for content providers to deliver personalized content such as movies, TV, music, commerce, and community throughout their applications.

The MyBestBets personalization engine combines content-based classification and users' preferences to match each individual with the contents which is best suited to his or her particular tastes and preferences. Users should complete a survey about their preferences (rating the items) before the system is able to give the recommendations. However, when filling such survey one needs some knowledge about the item classification, which common users usually lack. Moreover, in order to make the recommendations more accurate, the user should keep providing feedback, constantly re-valuating their preferences. Currently this engine is integrated on the AOL and Winamp websites.

MyBest TV

MyBest TV is a category-specific recommendation service currently embedded on AOL. This magazine-like service provides users with TV program recommendation that are delivered on-line or via e-mail. This service narrows options for consumers, helping them choosing a TV program they will really enjoy.

MyBestBets for music

MyBestBets for music personalization engine is currently utilized as a tab in the Winamp browser to provide a personalized music experience for the Winamp community. The services provided include: recommendations for CDs, short lists of radio station or music on TV, discovery of music buddies, etc.

Smart radio

The smart radio system[20] is a Web-based client-server application. It ties together the concept of a music program with a personalized recommendation to allow users to have personalized stream music programs.

User ratings in the smart radio system are gathered using either explicit user feedback (explicitly rating track items or individual programs) or an implicit way (scratching an initial play-list). After the system collects user ratings, it performs a users-to-users top-N collaborative filtering algorithm to construct and recommend the music programs to the user.

PTV

PTV is an Internet system offering personalized TV guides for each individual user. Its embedded ClixSmart personalization engine is a hybrid recommendation system combining content-based and collaborative filtering approaches. The benefits of this hybrid system are the ability to make diverse program recommendations, to cope with new or one-of-a-kind programs, and to cope with new or unusual users.

In this system, there are two databases used: a program database and a schedule database. The program database provides the content information regarding the programs such as program name, genre, country of origin, cast, studio, director, writer, and so on while schedule database stores the current channel schedules such as program name, its channel and time information, and a textual episode description. The contents in both databases are vital for obtaining the recommendations.

TiVo

TiVo is an automatic personal video recorder (PVR) that also adapts to user's interest. TiVo allows user to rate what he/she enjoys by using "Thumbs Up" and "Thumbs Down" buttons on the remote. With the user's preferences stored on the local receiver, TiVo matches those preferences with the program data it receives from the TiVo service and meanwhile searches through thousands of programs to create the own personalized suggestions and record programs into the hard disks accordingly.

Basic Solutions

To close the increasing gap between the amount of information that is available and the amount of useful information that one is able to extract, recommender systems have aroused more and more attention in the fields of electronic commerce as well as information retrieval. There are two prevalent approaches: content-based filtering and collaborative filtering. Recently, more and more research aims to combine the two approaches in order to gain better (more accurate) performance with fewer drawbacks than any of the individual approaches.

Collaborative filtering

One of the most promising, widely implemented and familiar technologies is collaborative filtering.[14,15,17–19] Collaborative filtering-based approaches utilize the correlations (commonalities) between customers on the basis of their ratings, to predict and recommend items which have the highest correlations to the user's rated /purchased items (user's preference).

In RINGO,[21] a personalized music recommendation system, similarities between the tastes of different users are utilized to recommend music items. This user-based collaborative filtering approach works as follows: A new user is matched against the database to discover neighbors, who are other customer who, in the past, have had a similar taste as the user, i.e., who have bought similar items as the new user. Items (unknown to the new user) that these neighbors like are then recommended to the new user.

Since the relationships between users are relatively dynamic (they continuously buy new products), it is hard to calculate the user-to-user matrix on-line. This causes the user-to-user (user-based) collaborative filtering approach to be relatively computationally expensive.

To address this, item-based algorithms[14,15] are introduced that explore the correlations between the items rather than the correlations between users. Recommendations are created by finding items that are similar to other items that the user likes (has already bought). Due to the fact that the item-to-item matrix is relatively static, it is possible to compute this matrix off-line. This extremely reduces the computational demands. This method has been successfully applied in the on-line retails such as Amazom.com.[19]

One of the drawbacks that influence the performance of this technique is that the recommendation is solely based on the historical rating data. The current task at hand or current context is ignored, even though it greatly affects the current interest or intent of the user. Authors in Ref. 22 proposed a pure collaborative filtering task-focused recommendation method to tackle this problem. In their approach, besides the long-term user's interest profile, a task profile is established by either explicitly providing some items associated with the current task or implicitly observing the user's behavior (intent). By utilizing the item-to-item correlation matrix, items that resemble the items in the task profile are obtained for recommendation. As they match the task profile, these items fit the current task of the user. These items will be re-ranked to fit the user's interests based on the interest prediction before recommending them to the user.

Content-based filtering

Content-based approaches recognize the correlation between contents of different items to predict and recommend items that have the most correlations to the user's rated/purchased items (user's preference).[23]

One of the requirements for the content-based method is the content description. Usually, contents of items are represented by metadata in the form of textual information. Lang in Ref. 24 uses words as features to filter newsgroups in a newsgroup filtering system: NewsWeeder. Similarly, a machine learning method for text-categorization is applied to content-based recommending of Web pages in Ref. 25 and to book recommending in Ref. 26.

Nevertheless, other researchers apply content-based analysis methods directly to the raw media data where the metadata is absent. In Ref. 27, music is categorized based on features extracted from the raw music file such as the pitch, tempo and loudness. These features are then used to build up a recommendation. This is achieved by recommending the music that has similar features to the music the user has recently listened to. Alternatively, in Ref. 28, authors try to learn the user's preferences by mining the melody patterns from the music access behavior. Music recommendation is achieved through a melody preference classifier. However, due to the difficulty of feature extractions, their experiments are all based on midi files and not real music.

In contrast to collaborative filtering, the item-to-item correlation is learned based on the representations (as features) of the items' content rather than based on the users' ratings.

Other approaches

There are many other recommendation approaches available. Here we just mention demographic recommender systems and knowledge-based recommender systems since they have unique attributes to help solving some problems (such as user cold-start etc.) that content-based and collaborative approaches cannot tackle.

Demographic recommender systems are aimed to categorize the user regarding to personal demographic data (e.g., age and gender) and classify items into the user classes. Approaches falling into this group can be found in Ref. 29 for book recommendation, and in Ref. 30 for marketing recommendation. Like collaborative filtering, demographic techniques also employ user-to-user correlations but differ in the fact that they do not require a history of user ratings.

Knowledge-based recommender systems attempt to reason about the relationship between a need and a possible recommendation. The user profile should encompass some knowledge structure that supports this inference. The system proposed by Ref. 31 tried to employ case-based reasoning to achieve the knowledge-based recommendation.

Hybrid approach

Although the collaborative filtering approach has significant advantages, often it is combined with other techniques to improve the recommendations. The other techniques include weighting, switching, mixing, feature combination, cascade, feature augmentation, meta-level, etc. A complete review about these techniques can be found in Ref. 11.

In Ref. 16, authors employed aspect modeling to model both the user rating based item-to-item correlation and content-to-user correlation. In this general probabilistic framework, content information and user rating information are systematically integrated in an attempt to solve the cold-start problem.

In Ref. 23, authors tried to boost the pure collaborative filtering by utilizing a content-based predictor. In order to provide content-based predictions, they treat the prediction task as a text-categorization problem. According to this, a bag-of-words naïve Bayesian text classifier is employed on the textual metadata to construct the content-based predictor. Consequently, the predictions from the predictor are treated as pseudo user-ratings vector and added to the user-based item-to-item correlation matrix to fill up the sparse spaces. This intuitively provides some pseudo ratings to the new items which are not rated by the customers.

Comparison and Analysis

All recommendation techniques have their advantages and drawbacks. Table 1 shows a comparison of the different recommendation techniques.

Collaborative filtering utilizes the correlations of users to recommend items liked by other similar users. The user profile is only based on the user ratings about the items and there is no content knowledge needed. This makes the technique extremely simple and general. In addition, it has the ability to provide cross-genre recommendation

Table 1 The advantages and drawbacks of the recommending technologies.

	Advantages	Drawbacks
User-based collaborative filtering	Can cross-genre recommending; non domain constrained	User/item cold-start problem; sparse problem; rating and history interaction data required; non task focus/context aware-less; expensive computation; less scalability
Item-based collaborative filtering	Can cross-genre recommending; no domain-constrained; off-line computations; scalability	User/item cold-start problem; sparse problem; rating and history interaction data required; non task focus/less context-aware
Content-based filtering	No item cold-start problem; no sparse problem; task-oriented	User cold-start problem; computation expensive; content description/training required; domain constrained
Demographic	Can cross-genre recommending; non domain constrained	Item cold-start problem; demographic data required; less accurate
Knowledge based	Non item/user cold-start problem; adaptive; including non-product features	Knowledge discovery required

(recommending items which are significantly different from previously obtained items according to the contents). The scalability problem of collaborative filtering can be solved by item-based collaborative filtering. The off-line computation of the item correlation matrix allows on-line processing of a large amount of items and users.

Collaborative filtering, however, also has some significant drawbacks. First, it suffers from the user and item cold-start problem. Lack of rating information on new items and new users cause that new items and new users cannot be categorized. The item cold-start problem can be tackled by utilizing the content information of new items while the user cold-start problem can be dealt with by extracting demographic data of new users. Second, collaborative filtering suffers from the so-called sparsity problem since the recommendation depends on the neighbors of the user. If the user rating space is sparse or when the target user is "an unusual user" the algorithm will fail since in both cases there are no relating neighbors.

Content-based recommendation approaches ignore the user correlation and only consider the correlations between the contents. They overcome the item cold-start problem by matching the content descriptions between the new item and the existing items. In addition, it is easy to be task-oriented by matching the user task and the content descriptions (metadata).

Nevertheless, content-based recommendation approaches suffer from some drawbacks as well. First, given the fact that most of the media data (audio, video) is opaque to the system, obtaining content descriptions is a problem. Second, the content-based user profile constrains (prunes) the region of the item space to a particular content. It effectively hampers the recommendation from dealing with the diverse taste of a user. For instance, a user could possible like both jazz music and dance music but a content-based approach will not be able to find the correlations between jazz music and dance music since their content descriptions are far apart.

CONCLUSIONS

In this entry, we presented the state-of-the-art in and our view on recommender systems. We then introduced a new wireless application called Wi-Fi walkman. Without bothering users with any annoying keywords input, the Wi-Fi walkman can steer user's music interest and recommend appropriate music in the peer-to-peer networks. We described scenarios for an MRS in the Wi-Fi walkman, and gave our scope and basic MRS by using the collaborative filtering technique.

REFERENCES

1. Jain, R. Folk Computing. IEEE Multimedia. **2002**.
2. Pine, B.J., II. *Mass Customization*, Harvard Business School Press: Boston, MA, 1993.
3. Audioscrobbler, http://last.fm.
4. FreeNet, http://freenet.sourceforge.net.
5. Bassoli, A.; Cullinan, C.; Moore, J.; Agamanolis, S. TunA : a mobile music experience to foster local interactions (poster). Fifth International Conference on Ubiquitous Computing (UbiComp 2003), Seattle, October 12–15, 2003.
6. Östergren, M. Sound pryer field trials: learning about adding value to driving. Workshop Designing for ubicomp in the wild: Methods for exploring the design of mobile and ubiquitous services (MUM'2003), Norrköping, Sweden, Dec 10–12, 2003.
7. Gnawali, O.D. *A keyword set search system for peer-to-peer networks*; Master's thesis, Massachusetts Institute of Technology, June 2002.
8. Li, J.; Loo, B.; Hellerstein, J.; Kaashoek, F.; Karger, D.; Morris, R. On the feasibility of peer-to-peer web indexing and search. 2nd International Workshop on Peer-to-Peer Systems, 2003, Berkeley, CA, Feb 20–21, 2003.
9. Cooper, B.; Garcia-Molina, H. Studying search networks with SIL. 2nd International Workshop on Peer-to-Peer Systems, 2003, Berkeley, CA, Feb 20–21, 2003.
10. Bhattacharjee, B.; Chawathe, S.; Gopalakrishnan, V.; Keleher, P.; Silaghi, B. Efficient peer-to-peer searches using result-catching. 2nd International Workshop on Peer-to-Peer Systems, 2003, Berkeley, CA, Feb 20–21, 2003.
11. Burke, R. Hybrid recommender systems: survey and experiments. User Model. User-Adapt. Interact. **2002**, *12* (4), 331–370.
12. Goldberg, K.; Roeder, T.; Gupta, D.; Perkins, C. Eigentast: a constant time collaborative filtering algorithm. Information Retrieval J. **2001**, *4* (2), 133–151.
13. Sarwar, B.M.; Karypis, G.; Konstan, J.A.; Riedl, J. Application of dimensionality reduction in recommender system—a case study. Web Mining for E-Commerce Workshop, 2000 (WebKDD 2000), Boston, MA, Aug 20, 2000.
14. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. WWW10 Conference, Hong Kong, May 1–5, 2001; 285–295.
15. George, K. Evaluation of item-based top-N recommendation algorithms, Technical Report #00-046. Department of Computer Science, University of Minnesota, 1999.
16. Schein, A.I.; Popescul, A.; Ungar, L.H.; Pennock, D.M. Generative models for cold-start recommendations. 2001 SIGIR Workshop on Recommender Systems, 2001, New Orleans, LA, Sept 13, 2001.
17. Konstan, J.; Miller, Bo; Maltz, D.; Herlocker, J.; Gordon, L.; Riedl, J. GroupLens: applying collaborative filtering to Usenet news. Commun. ACM. **1997**, *40* (3), 77–87.
18. Breese, J.S.; Heckerman, D.; Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. 14th Conference on Uncertainty in Artificial Intelligence (UAI-98), Madison, WI, July 24–26, 1998; Cooper, G.F., Moral, S., Eds.; Morgan-Kaufmann: San Francisco, CA, 1998; 43–52.
19. Linden, R.; Smith, B.; York, J. Amazon.com recommendations. IEEE Internet Comput. **2003**, *7* (1), 76–80.
20. Hayes, C.; Cunningham, P. Smart Radio—Building Music Radio on the Fly. Expert Systems 2000, Cambridge, UK, 2000.
21. Shardanand, U.; Maes, P. Social information filtering: algorithms for automating 'word of mouth'. Conference on

Human Factors in Computing Systems (CHI95), Denver, CO, May 7–11, 1995; ACM Press: 1995; 210–217.
22. Herlocker, J.L.; Konstan, J.A. Content-independent task-focused recommendation. IEEE Internet Comput. **2001**, *5* (6), 40–47.
23. Melville, P.; Mooney, R.; Nagarajan, R. Content-boosted collaborative filtering for improved recommendations. AAAI 2002, Mar 25–27, 2002.
24. Lang, K. NewsWeeder: learning to filter Netnews. 12th International Conference on Machine Learning, Tahoe City, CA, 1995.
25. Pazzani, M.; Muramatsu, J.; Billsus, D. Syskill & Webert: identifying interesting web sites. 13th National Conference on Artificial Intelligence, Portland, OR, 1996; 54–61.
26. Mooney, R.J.; Roy, L. Content-based book recommending using learning for text categorization. 5th ACM Conference on Digital Libraries, San Antonio, TX, June 2–7, 2000; 195–204.
27. Chen, H.C.; Chen, A.L.P. A music recommendation system based on music data grouping and user interests. ACM International Conference on Information and Knowledge Management (CIKM 2001), Atlanta, Georgia, Nov 5–10, 2001.
28. Kuo, F.F.; Shan, M.K. A personalized music filtering system based on Melody style classification. IEEE International Conference on Data Mining, (ICDM 2002), Japan, Dec 9–12, 2002.
29. Rich, E. User modelling via stereotypes. Cogn. Sci. **1979**, *3* (4), 329–354.
30. Krulwich, B. Lifestyle finder: intelligent user profiling using large-scale demographic data. Artif. Intell. Mag. **1997**, *18* (2), 37–45.
31. Schmitt, S.; Bergmann, R. Applying case-base reasoning technology for product selection and customization in electronic commerce environments. 12th Bled Electronic Commerce Conference, Bled, Slovenia, June 7–9, 1999.

BIBLIOGRAPHY

1. Gnutella. http://www.gnutella.com.