When Media Gets Wise: Collaborative Filtering with Mobile Media Agents

Mattias Jacobsson, Mattias Rost, Lars Erik Holmquist
Future Applications Lab, Viktoria Institute
Hörselgången 4, 417 56 Göteborg, Sweden
www.viktoria.se/fal
{majak, rost, leh}@viktoria.se

ABSTRACT
We present a model where media (e.g. music files) are autonomous entities that carry their own individual information. Our goal is to turn such files into autonomous, rule-following agents capable of building their own identities from interactions with other agents and users. We are exploring how collaborative filtering-like behaviour could emerge out of large ensembles of interacting agents, which are distributed over mobile devices in social networks. We have implemented a first version of the model in the form of a music player application for mobile devices, called *Push!Music*. This system takes advantage of active recommendations as well as implicit user activity to build a profile for each media file.

Categories and Subject Descriptors
I.2.11 [**Distributed Artificial Intelligence**]: Multi Agent Systems

General Terms
Algorithms, Design, Human Factors

Keywords
Media Agents, Media Ecologies, Distributed Collaborative Filtering, Active Recommendation, Emergence

1.INTRODUCTION
We are developing a mobile recommender system where media – such as music files, movies, photos, etc. – can find people rather than the other way around. We approach this concept from a decentralized point of view, where media can move autonomously between users (peers) in a social network. Our model tries to make such media entities more dynamic, having them propagate by means of autonomous recommendation. To achieve this we have looked at traditional recommender systems, social networks as well as biologically inspired systems, such as ant trails. In this way we also intend to address some of the problems that are commonly related to centralized recommender systems such as scalability, trust and concept drift. We are initially deploying this model as a digital music player on a mobile platform (i.e. a handheld computer with wireless networking) which allows us to study both the technology and user aspects. The system takes advantage of user behavior, such as listening patterns and active recommendations to influence the behavior of the agents. The following scenario gives an idea of its intended use:

"Several people are sitting on the bus listening to music on their personal players. Some look out over the city dreaming away, while others are reading the morning newspaper. On your player, you can see that about ten other riders are using Push!Music. While you listen to a metal rock tune, you decide to send it to a nice-looking guy with long curly hair at the front of the bus, but your stop is coming up and you have to get off before you can see his reaction. As you leave the bus, you realize that a completely new song has entered your playlist… interesting tune! We will see what happens tomorrow…"

This scenario shows how a user has sent a song recommendation to another person on the same bus, without his knowledge. Meanwhile, a media agent on some other person's player has observed the user's device and decided to move itself to it – perhaps triggered by the active recommendation that just took place.

2.RELATED WORK
An example of a mobile recommender system that attempts a decentralized approach is *PocketLens* [5]. It seeks to bootstrap a centralized system in order to make it work on portable devices e.g. PDA's. The system still needs a central server to synchronize with from time to time hence it could be said to be quasi-decentralized. Our approach differs from theirs in that we seek to explore a fully decentralized system by binding information tightly to the content itself.

An example of a shared mobile media system is the music player *tunA* [1]. Here music is streamed over the ad-hoc network, allowing nearby users to eavesdrop onto each other's devices as a mean to create new types of social listening experiences. We advance on this by allowing media files to move between devices, either through explicit user interaction or on their own accord.

3.CONCEPT
In a pre-study to this project we defined the concepts of *media agents* and *media ecologies* [3]. The following definition of a *Maes Agent* [2], found in a taxonomy survey of agents, gives a good indication of how we want users to experience the agent aspects of a media agent. The main difference is that a media agent is directly joined to a content unit, such as an MP3-file, whereas a traditional agent does not have any corresponding content storage.

IUI'06, January 29–February 1, 2006, Sydney, Australia.
ACM 1-59593-287-9/06/0001.

"Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they were designed."

We also use the term media ecology to describe the "natural habitat" for a media agent. In our current application, where an MP3-file is the content-part of the media agent, such a habitat could for instance be the hard-disc of a portable MP3-player.

Further, we need to define a *context* in the form of a stream of media, e.g. a playlist, which a user experiences over time. It is in this natural context that we believe agents could "socialize" with each other. It is also in this context that the user would be exposed to the media content, and vice versa. Our hypothesis is that agents can feel a sense of similarity towards each other based on their contextual experiences with other agents.

In our preliminary study [3], we found that making recommendations to friends was an important part of the social experience of music. This observation together with the need for getting additional input for agents led to the concept of active recommendation. In reality it means that people can actively send – or "push" – a song to another person's device. This new song will then appear as the next item on his or her playlist.

4.IMPLEMENTATION

4.1Background

Collaborative filtering (CF) algorithms originated from Shardanand & Maes "automated word of mouth"-concept [8]. This work was based upon user profiles and finding neighborhoods among users. This led to the *User-User* CF algorithm, one of the first successful algorithms for recommender systems. The shortcomings of this early approach such as scalability, sparcity and computational expenses could soon be addressed through the next generation algorithms, *Item-Item* [7]. A basic similarity measure used here is the *cosine similarity* where the user's *u* who has rated both items *i* and *j* (co-occurrences) will contribute to the mutual similarity for those items.

One important benefit that the Item-Item approach has over the User-User is that newly joining users immediately can make use of the system. The similarity matrix can also be updated continuously without the need of calculating user neighborhoods. This observation also opens up for a more distributable approach where pieces of the similarity matrix can be separated and merged as a jigsaw puzzle.

Of equal, if not of more importance are the ratings. Any CF system needs to be fed with preferences of some kind. The two fundamental ways of retrieving information about user preferences is either implicit or explicit [6]. Either way has advantages and disadvantages, which are almost always perpendicular. The conscious way of explicitly providing information usually needs to be motivated by a profit to compensate for the effort. This gives the implicit way a head start in the area of CF. Not the least because of the "free-riding" problem, where users tend to use the system without providing input themselves. We address this problem by both introducing new mechanisms that could be used as input and also analyze existing inputs from moments when the user interacts naturally with media.

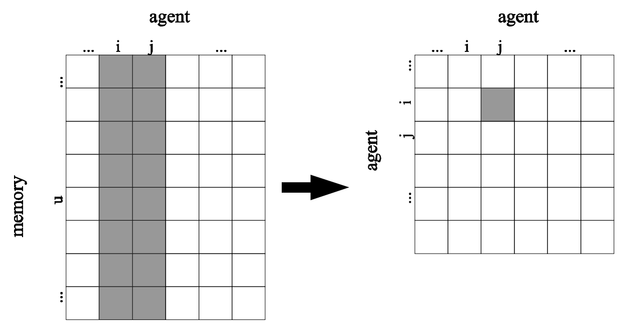

Figure 1: Contextual memories of agents *i* and *j* are weighted with corresponding significances, and then transformed into a similarity matrix.

4.2Our Model

In our model we build upon the ideas from the Item-Item algorithm but replace the users with the contexts of a media agent. The own rating *r* for an agent is then used to evaluate the significance of different parts of the context. The ratings within the context are denoted by *R*. The maximum possible rating value is denoted by R_{max}. All in all this results in a weight that we call significance which is defined by

$$\sigma_{u,i} = \frac{R_{u,i} r_{i,h}}{R_{\max}^2} \tag{1}$$

The interpretation here is that agents with similar and high values are more significant, i.e. higher ratings occurring in the same context *h* should contribute more to the overall similarity.

We then calculate the Agent-Agent similarity matrix (figure 1) by comparing agents from the ecologies.

$$sim(i,j) = \frac{\sum_u R_{u,i} R_{u,j} \sigma_{u,i} \sigma_{u,j}}{\sum_u \left(R_{u,i}\sigma_{u,i}\right)^2 \sum_u \left(R_{u,j}\sigma_{u,j}\right)^2} \tag{2}$$

The new measure is merely an extension of the cosine similarity, where the user now has been replaced by contextual memory. Several users contribute to this memory so that the collaborative aspects still would hold.

An agent would then probe other ecologies upon intersection (e.g. mp3-players featuring WiFi), keeping records of similarity scores e.g. average. Agents would then move or copy itself in the direction of good scores. Agents also exchange complementary parts and refines existing contextual memory when they encounter other agents with identical or even similar media content.

5.MUSIC LISTENING APPLICATION

To explore this model we have developed an application for mobile devices called *Push!Music* [4], which is implemented on PocketPC handheld computers and uses WiFi for ad hoc communication between peers. In this case the media agent consists of two files that travel together: the music file and a data file which contains all contextual information. We decided to keep the active part of contributing to the context at a minimum. Users are not obliged to rate songs, but there is a simple system where they can decide with a single button-press if a song "rocks" or "sucks".

Rule	Symbol
20% rule	T
90% rule	N
Active recommendation (Push!)	P
Autonomous recommendation	A
Local recommendation	L
Positive vote	+
Negative vote	-
Delete	D

Table 1: Rules extracted from how users behaves when using a mobile music device, e.g. an MP3-player.

However, the most important context information is being collected implicitly. Here we take advantage of the user's normal listening behaviour. Our music player has a standard play/pause button to start/stop a stream of music. There is also the option of skipping back and forth on a playlist. Finally, the system allows for active recommendations – "pushing" music to another user in the ad-hoc network. The user can select another device in the vicinity and send a music file complete with context information, i.e. a media agent, to that device. This can be done either collaboratively, between two users who are aware of each other, or anonymously, where the receiver is not aware of the action until the new song pops up in his or her playlist.

From these natural interactions together with the simple rating feature, we derived a table of binary rules (Table 1). The two first rules try to capture whether the user skips the song quite fast or really listens to the song. The next rules regarding recommendation captures how the song got there in the first place. Such rules then allows for more complicated, but still interpretable schemes, to be formed. A scheme in this case is a combination of one or more rules into a sequence of events that can be ordered in a good-bad fashion.

An example of a scheme is '+NP', which tells us that a user listened to more than 90% of the song, voted it as a good song and also pushed the song to someone else. This would result in an implicit rating value of 10, which is the maximum in this case. Another example would be '-TD', which corresponds to negative vote, up to 20% of the song were listened to and it was also marked for deletion.

6.CONCLUSIONS AND FUTURE WORK

Preliminary results from tests with the Push!Music application show that two disjunct sets of music files do not mix until there is some common denominator pulling the ecologies together. It then shows that agents do move autonomously between devices and deploy themselves in their new ecology.

We are now preparing a user test where several groups of external users are to use the application for a longer period of time. We will get a mix of users, some who know each and others who are not acquainted, but spend time in the same location. The results from this study will help us understand the viability of our model and see how valuable the recommendations that emerge will really be.

Finally, major issues such as privacy and copyright need to be addressed. There are already payment models that allow for legal filesharing over computer networks, so called super distribution, e.g. the SnoCap system where each user in effect becomes a re-seller of media (www.snocap.com). This together with the current interest in filesharing and mobile devices indicates that there is need for new and more suitable recommendation and distribution models, which is exactly what we hope to achieve with the work presented here.

7.ACKNOWLEDGMENTS

We would like to thank the rest of our group at FAL for valuable input and interesting discussions. This project is supported by IST-FET in the EC-Agents project (IST-2003-1940) and the Mobile Services project funded by the Swedish Foundation for Strategic Research.

8.REFERENCES

[1] Bassoli, A., Moore, J., and Agamanolis, S., tunA: Local Music Sharing with Handheld Wi-Fi Devices, *In Proc. of 5th Wireless World Conference 2004*, Surrey, UK.

[2] Franklin, S., and Graesser, A., Is it an Agent or Just a Program?: A Taxonomy for Autonomous Agents, *In Proc. of the 3rd International Conference on Agent Theories, Architectures and Languages*, Springer-Verlag, 1996.

[3] Håkansson, M., Jacobsson, M., and Holmquist, L. E., Designing a Mobile Music Sharing System Based on Emergent Properties. *In Proc. of AMT 2005*, Takamatsu, Japan.

[4] Jacobsson, M., Rost, M., Håkansson, M., and Holmquist, L. E. Push!Music: Intelligent File Sharing on Mobile Devices, *In Adjunct Proc. of UbiComp 2005*, Tokyo, Japan. (demo)

[5] Miller, B. N., Konstan, J. A., and Riedl, J., PocketLens: Toward a personal recommender system. *ACM Trans. Inf. Syst.* 22, 3 (Jul. 2004), 437-476.

[6] Nichols, D. M. Implicit Rating and Filtering*, In Proc. 5th DELOS Workshop on Filtering and Collaborative Filtering 1997*, Budapest, Hungary.

[7] Sarwar, B., Karypsis, G., Konstan, J., and Riedl, J., Item-based Collaborative Filtering Recommendation Algorithms, *In Proc. of the 10th International World Wide Web Conference (WWW10) 2001*, Hong Kong.

[8] Shardanand, U., and Maes, P., Social Information Filtering: Algorithms for Automating "Word of Mouth", *In Proc. of CHI '95*, Denver, Colorado, USA.