Decentralised Social Filtering based on Trust

Tomas Olsson

Swedish Institute of Computer Science
Box 1263
SE-164 29 Kista
tol@sics.se

Abstract

This paper describes a decentralised approach to social filtering based on trust between agents in a multiagent system. The social filtering in the proposed approach is built on the interactions between collaborative software agents performing content-based filtering. This means that it uses a mixture of content-based and social filtering and thereby, it takes advantage of both methods.

Introduction

Today's recommender systems using social filtering are mainly centralised, such as Firefly [Firefly 1997], LikeMinds [LikeMinds 1997], etc. What is suggested is a system that proposes web documents to its users through decentralised social filtering based on trust.

The proposed approach consists of a network of users connected by personal agents. Each agent has a model of its user and based, not only on the content of the documents, but also on the *trust* for other agents, they propose documents to their users and help each other filter documents. The basis of the proposed approach is the *content-based filtering layer* and on top of it, there is the *collaborative filtering layer*, which is followed by the *social filtering layer*. However, it is also possible to get a working system at each layer.

Because the complete system uses a mixture of content-based filtering and social filtering, it takes advantage of both methods.

Content-based Filtering

Content-based filtering analyses the content of documents and compares them to a user model. The closer the match, the likelier the document will interest the user and at some degree of closeness, the document is proposed. Relevance feedback from the user for proposed documents is used to change the user model. Content-based filtering works quite well with text based documents, because it is relatively easy to build good user models based on text.

An advantage of content-based systems is that they can propose new documents not seen by a user if the document has a content similar to the content of previously encountered documents. However, this is also a problem because they are not able to propose documents with a kind of content not previously encountered (serendipity) [Firefly 1997].

Another problem for content-based systems is that it usually takes some time before they start to work well for new users (cold-start) [Lashkari, Metral & Maes 1994]. Content-based systems are often of stand-alone type and they can not benefit from the other users in the system. This means that they have to start from scratch for every new user and it will usually take some time before they have built new good user models.

Social Filtering

Social filtering analyses the users' ratings of documents and compares them to each other to be able to propose new documents [Firefly 1997]. If a user likes certain documents, the user will probably like some other documents because other users with the same preferences did. Because social filtering relies on other users' ratings of documents and not on the content, it works quite well, not only in the domain of text documents, but also in other less evaluative domains, such as pictures, movies, music, etc.

For social filtering, the problem of serendipity is reduced because it does not use the content of documents and thereby, a user can get proposals of documents with a different kind of content than previously encountered. The problem of cold-start is also reduced, because a new user benefits from the work of the other users and thereby, there is no need to start from scratch for every new user.

A problem for social filtering methods is that some users, and sometimes quite a few users, must have rated documents before the system starts to work well and the documents can be proposed to the users [Shardanand & Maes 1995]. Observe that this means that the documents must have been seen and rated by another user before they can be proposed. Notice as well that the time before a system has sufficient many users could be seen as cold-start, but it is only a problem in the initialisation of the system and not for every new user.

Yet another problem for systems based on traditional social filtering is that they are mainly centralised [Foner 1997]. This is a bottleneck for the scalability and the availability of the system, and a risk for the privacy of the users.

The scalability is a problem because by adding new users, the computational load on the computer is increased and more computational power is needed. In a centralised approach, this is probably handled by buying a more powerful computer. For a decentralised approach, the addition of new users is probably not even noticeable, because with the new users there will also be new computers (if we assume that each agent of a user is running locally on the user's computer). However, the increase of communication between the entities of a decentralised approach might also lead to scalability problems.

The availability is a problem because there is a single point of failure and the privacy of the users is at risk because of the sensitive information the system has gathered at one place. If a failure occurs, an unauthorised person might get hold of all the user models. Because of this, it might be a good idea to make the system decentralised. If an unauthorised person gets access to a user model in a decentralised approach, the user will still not have access to all the other user models. However, a decentralised approach has other kinds of security risks, for example, it might be a problem to protect the communication between the decentralised entities.

A Decentralised Approach

Yenta is a decentralised approach that matches people's interests to introduce them to each other [Foner 1996; Foner 1997]. However, Yenta could also be made to propose documents to its users. In Yenta, the agents first build models of their users' interests based on the content of the users' documents. Then they compare their models to find other agents with similar interests. The agents cache the other agents' models for later referral. Foner mainly addresses the problem for agents to find each other without any central control. His approach is to let the agents self-organise into clusters with other agents with similar interests. He achieves this self-organising in the same manner as we humans find other people with the same interests, that is by referrals; through knowledge about other people and the other people's knowledge about additional other people, etc.

Collaborative Filtering

Although Yenta does not propose documents, one could easily imagine it proposing documents based on its user models. An agent, could given a document, send it to the known agents with the most similar models. Because the system would propose documents based on their content, it seems reasonable to call it a content-based filtering system. Nevertheless, there is also a social aspect because the agents help each other filter documents. The concept '*collaborative filtering*' will be used for systems where the users help each other filter and propose interesting documents. The concept of 'social filtering', will be saved for the method previously described and they will not be used as synonyms. However, social filtering is a form of collaborative filtering, but not the other way around. Why this distinction is used will be clear later in the section about the proposed approach.

Additional Related Work

Related work to the proposed approach is also the collaborative filtering of net news in [Maltz 1994] and the collaborative filtering system presented in [Balabanovic & Shoham 1997]. Other related work, but not quite as obvious as the previous ones, is the system for finding documents described in [Marsh & Masrour 1997], the collaborative interface agents for filtering e-mail in [Lashkari, Metral & Maes 1994] and the referral-based collaborative system for finding experts described in [Kautz, Milewski & Selman 1996].

The Proposed Approach

The proposed approach for decentralised social filtering consists of a *content-based filtering layer*, a *collaborative filtering layer* and a *social filtering layer* on top of a multiagent system. However, it is possible to get a working system at each layer. The layers are shown in Figure 1.

The first layer, the *content-based filtering layer*, consists of the users' personal agents using content-based filtering. The second layer, the *collaborative filtering layer*, is based on the content-based filtering layer and its agents' use of the content of documents to route and recommend interesting documents to each other and to its users. The third

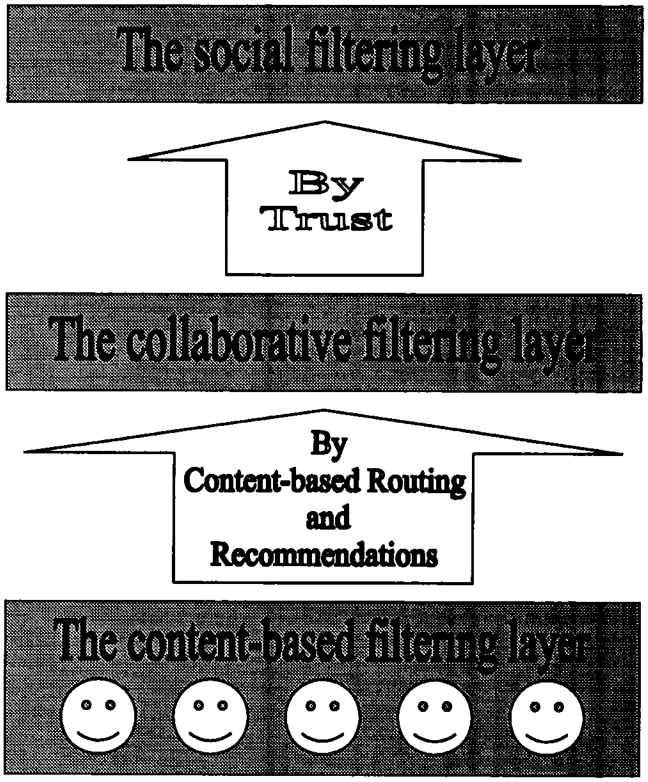

The social filtering layer
By
Trust
The collaborative filtering layer
By
Content-based Routing
and
Recommendations
The content-based filtering layer

Figure 1: The layered structure of the proposed approach. Happy faces symbolise the agents of the multiagent system.

layer, the *social filtering layer*, is constructed on top of the collaborative filtering layer by computing a confidence value that states how much an agent trusts another agent for recommendations of interesting documents. Based on the trust, the agents can choose what other agents to subscribe recommendations from and what documents they should propose to the users.

By using trust, an agent can propose documents based on a different criterion than the content similarity and because the trust is based on the users' ratings, one can say that it is using a form of social filtering.

In the proposed system, there are two types of agents, the Interface Agent and the Interest Agent. The agents in Figure 1 correspond to the Interest Agents. In Figure 2, one can see the connections between an Interface Agent and its Interest Agents.

The Interface Agent

An Interface Agent is an interface between the system and a user. The Interface Agent is associated to a web-browser where the user can browse web documents and sort them into user defined categories in a similar way as for bookmarks. There is also an area attached to the browser where the agent shows the documents it proposes to the user.

The Interface Agent is responsible for the formation of the user model, which is based on the user's categories. By putting a document in a category, the user rates it as an interesting document for that category and uninteresting for the other categories. A category can be said to correspond to an interest of a user and for each category, the Interface Agent creates an Interest Agent. The documents that the Interest Agents, based on the categories, propose to the Interface Agent, it will forward as proposals to the user.

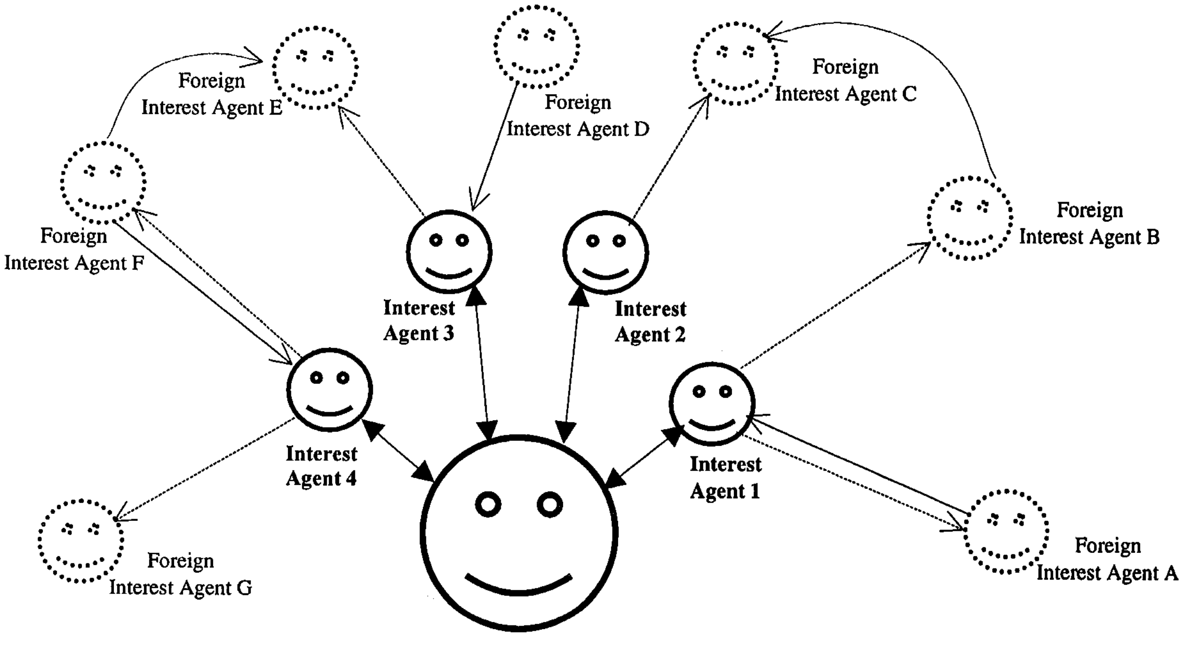
The Interest Agent

An Interest Agent is responsible for the content-based, collaborative and social filtering layers of the system.

There are mainly three tasks for an Interest Agent to perform:

- It forms an interest model from the content of the documents in the category it got from the Interface Agent.
- It proposes documents to the Interface Agent based on the comparison between the interest model and the documents (content-based filtering layer) or based on the trust in the recommending Interest Agent (social filtering layer).
- It recommends or routes documents to other agents and thereby, it finds users with similar interests represented by other Interest Agents (collaborative filtering layer).

When an agent *recommends* a document, it means that its user has rated it as interesting and implicitly it also means that the agent wants more of its kind. However, when an agent *routes* a document, it means either that its user rated it as uninteresting or that the agent has not proposed the document to the user. Hence, one can not draw any conclusions from the routing of documents. Nevertheless,



Interface Agent

Figure 2: An Interface Agent and its Interest Agents with Retrieval Agents and other users' Interest Agents. An arrow shows that the information goes from the agent pointed at to the pointing agent.

attached to every recommended or routed document, there is a list with the agents recommending it. By using this list, the agents are able to find new agents with similar interests.

For the Interest Agent to be capable of proposing documents to the user based on trust and for it to be capable of recommending or routing documents, it builds models of some other Interest Agents. The model of an Interest Agent consists of a confidence value for the Interest Agent and an interest model of its interest. The confidence value states how much it can be trusted for interesting recommendations and the value is based on the previously recommended interesting or uninteresting documents (a social aspect for the social filtering layer). The interest model of another agent is based on the content of all of its previously recommended documents. An agent can use the interest models to choose where to recommend or route documents (a content-based aspect for the collaborative filtering layer). To improve the performance of the system, an Interest Agent can also subscribe recommendations from the Interest Agents for which it has most trust.

This flow of interesting and uninteresting documents between the agents of the system makes it possible for the Interest Agents to find other agents and thereby, they can cluster into groups of agents with similar interests. In these clusters, it will be possible to spread interesting documents quite fast. However, the system must be able to limit the traffic in some way, for example, by using a time-to-live for each recommendation.

The Advantages of the Proposed System

By using the agents described in the previous sections, one can have both the advantages of social filtering and the advantages of content-based filtering.

The system can propose documents to a user based, not only on the content of the documents, but also on the trust computed from the other users' ratings. This means, that the system can find documents with properties earlier not encountered and thereby, it reduces the problem of serendipity.

The problem of cold-start is reduced since a new user can benefit from the work of the other users in the system (the collaborative filtering). However, it might take some time before the agent of a user has learned what other agents to trust, that is, before it has computed high confidence values for them.

One advantage from the content-based technique is that the agent can find documents of a kind that no other user has seen before. An agent could easily be made to work as a personal web search robot that automatically finds new documents based on the user model (In [Olsson 1998] a Retrieval Agent that works in this way is described). An agent can propose documents that only one single user has rated previously by using content-based filtering or by using the trust for the recommending agent. This means that the system does not require as many users as the social filtering method described in the first section to be able work.

A Comparison to Previous Approaches

The difference between the proposed approach and previous social filtering approaches is foremost that it tries to combine the techniques of content-based and social filtering in a decentralised solution with the expectation to take advantage of both techniques.

The proposed solution is completely decentralised, thereby, it has no centralised parts, and therefore it does not suffer from the corresponding disadvantages of scalability, availability, etc. Two differences compared to Yenta:

- The clustering of similar agents in the proposed system is not based on the similarities between the content of their interest models as in Yenta. Instead, the similarities between the Interest Agents are measured by the social connections (the trust).
- In Yenta, the agents must be able to compare their interest models, but the proposed system recommends or routes documents, which means that only the wrapper of a document sent to another agent must be standardised. The agents can represent the interests of users or the models of other agents in different ways.

Important Problems to Solve

An important problem to solve in the proposed approach is how the system can protect the privacy of the users' communication. This is discussed in [Foner 1996].

The users may also have many other reasons to not share what documents they find interesting, but the gain from using the system might be greater than the disadvantages.

Conclusions

A decentralised system will be more frequently available than a centralised system since there is no single point of failure.

The addition of new users is probably not noticed in a decentralised system, because when one adds new users, one also adds their computers (they already exist) instead of buying a more powerful computer.

In a centralised system, unauthorised access to, or misuse of, the central node would mean that all the user models are revealed. This means that a system with decentralised user models might protect the privacy of the users in a better way than a centralised system.

By using both social and content-based filtering, one gets the advantages of both methods. One can find documents rated by only one other user (or no user) and one reduces the problem of serendipity and the problem of cold-start.

Future Work

The Interface Agent and the Interest Agent has been implemented and tested in a Master Thesis [Olsson 1998] at Ellemtel Utvecklings AB [Ellemtel 1998] in Sweden. The test was rather limited and not all the expected advantages were tested.

The future work will be to incorporate the idea of decentralised social filtering in a digital library project at SICS [SICS 1998b]. In this project, we will create a Virtual Community Library based on a community of interacting personal library agents. The implementation of the system will be done with an agent toolbox developed in a market space project at SICS [SICS 1998a].

References

Balabanovic, M & Shoham, Y. 1997. Fab: Content-based, Collaborative Recommendation. In: *Communication of the ACM*, Vol 40, No 3

Ellemtel Utvecklings AB. 1998.
http://www.ellemtel.se

Firefly Network, Inc. 1997. Collaborative Filtering Technology: An Overview.
http://www.firefly.net/company/CollaborativeFiltering.fly
[Accessed 27/08/97]

Foner, L N. 1996. A Security Architecture for Multi-Agent Matchmaking. In: *Proceeding of The Second International Conference on Multi-Agent Systems (ICMAS'96).* Keihanna Plaza, Kansai Science City, Japan

Foner, L N. 1997. Yenta: A Multi-Agent, Referral-Based Matchmaking System. In: *Proceedings of The First International Conference on Autonomous Agents (Agents'97)*, 301-307. ACM Press

Kautz, K & Milewski, A & Selman, B. 1996. Agent Amplified Communication. In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, Vol 1, 3-9. AAAI Press/The MIT Press

Lashkari, Y & Metral, M & Maes, P. 1994. Collaborative Interface Agents. In: *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI'94)*, Vol 1, 444-450. AAAI Press/The MIT Press.

LikeMinds, Inc. 1997. LikeMinds Architecture. See:
http://www.likeminds.com/technology/architecture/
[Accessed 06/03/98]

Marsh, S & Masrour, Y. 1997. Agent Augmented Community Information - The ACORN Architecture. In: *Proceedings CASCON'97*, Meeting of Minds

Maltz, A D. 1994. Distributed Information for Collaborative Filtering on Usenet Net News. M. Sc. Thesis. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology

Olsson, T. 1998. Information Filtering with Collaborative Interface Agents. M. Sc. Thesis. Department of Computer and Systems Sciences, Royal Institute of Technology, Sweden

Shardanand, U & Maes, P. 1995. Social Information Filtering: Algorithms for Automating "Word of Mouth". In: *Proceedings of 1995 Conference on Human Factors in Computing Systems (CHI'95).* ACM Press.

SICS. 1998a. Agent-Based Market Space — Agent-Mediated Electronic Commerce.
http://www.sics.se/~market/

SICS. 1998b. The Agent based Digital Library Infrastructure Project.
http://www.sics.se/isl/diglib