Opinion-Based Filtering Through Trust

Miquel Montaner, Beatriz López, and Josep Lluís de la Rosa

Institut d'Informàtica i Aplicacions
Agents Research Laboratory
Universitat de Girona
Campus Montilivi
17071 Girona, Spain
{mmontane, blopez, peplluis}@eia.udg.es

Abstract. Recommender systems help users to identify particular items that best match their tastes or preferences. When we apply the agent theory to this domain, a standard centralized recommender system becomes a distributed world of recommender agents. Therefore, due to the agent's world, a new information filtering method appears: the opinion-based filtering method. Its main idea is to consider other agents as personal entities which you can rely on or not. Recommender agents can ask their reliable *friends* for an opinion about a particular item and filter large sets of items based on it. Reliability is expressed through a trust value with which each agent labels its neighbors. Thus, the opinion-based filtering method needs a model of trust in the collaborative world. The model proposed emphasizes proactiveness since the agent looks for other agents in a situation of lack of information instead of remaining passive or providing either a negative or empty answer to the user. Finally, our social model of trust exploits interactiveness while preserving privacy.

1 Introduction

Recommender systems make recommendations to users according to the information available. Such information includes data on items as well as different profiles of other users on the web. Since there is so much information, a fundamental issue is to select the most appropriate information with which to make decisions. In other words, an information filtering method is essential. Usually, three information filtering approaches have been used in the state of the art for making recommendations [11]: demographic filtering, content-based filtering and collaborative filtering. Moreover, hybrid approaches among them have been proved useful.

However, when we apply the agent theory to recommender systems, a standard centralized recommender system becomes a distributed world of recommender agents [7]. Each user has his/her own recommender agent that is able to interact with others. In an open environment such as Internet, however, the interaction of a recommender agent with all possible agents in order to obtain the best recommendation for the user seems unapproachable. The solution we propose in this paper is a new information filtering method: the opinion-based filtering method. Its main idea is to consider other agents as personal entities which you can rely on or not. Reliability is expressed through a trust value with which each agent labels its neighbors. Trust is one of the most important social concepts that helps human agents to cope with their social environment, and is present in all human interaction [5]. Some efforts have been made in the study of social models of trust in market environments [12], where several agents compete for their individual profit as well as in other environments where agents need to delegate actions to other agents [2]. Trust, however, is also important in filtering information environments where recommender agents asses users. Just as in the real world people ask their friends for advice on interesting items, an agent should be able to ask only reliable agents. For example, a common situation is when somebody asks a friend for advice about a new restaurant. Another common situation is when someone discovers a new restaurant and wants to know the opinion of his/her friends about it or when somebody tells you something about a new restaurant, you want to check this information with your friends. If they already know the restaurant, they can give you their opinion, whereas if they do not know it, as from the features of the restaurant (e.g., cuisine, price,...) they can guess an opinion. But people do not ask just anyone for advice. People only ask for advice to friends with similar tastes and interests who can be trusted. And, how do people know whether other people have similar tastes and interests? Usually, through interaction. If you want to know someone's tastes and interests, you ask him/her his opinion. For instance, in the restaurant example, you ask someone his/her opinion about restaurants that you love and about restaurants that you hate. If this person has a similar opinion, you consider him/her someone with similar preferences. In the information filtering context, agents are not considered reliable either because their honesty or their trustworthy information but because of similar preferences, interest, styles.

Therefore, the opinion-based filtering method we propose is based on a model of trust in the collaborative world of recommender agents. Mainly, we provide recommender agents with a technology that allows them to look for similar agents that can offer them advice. The model proposed emphasizes proactiveness since the agent looks for other agents in situation of lack of information instead of remaining passive or providing either a negative or an empty answer to the user. Finally, our social model exploits interactiveness while preserving privacy.

The new approach of the information filtering method is presented as follows. Section 2 justifies the need of trust in recommender agents. With trust, a new information filtering method comes up that is explained in section 3. Section 4 introduces the formal social model of our approach to trust for recommender systems. Section 5 presents related work and, finally, in section 6 we provide some conclusions.

2 The Need of Trust in Recommender Agents

Recommender agents are used to asses the user by filtering information. Three information filtering methods have been proposed in the current state of the art [11]: demographic filtering, content-based filtering and collaborative filtering. Demographic filtering approaches use descriptions of people to learn about a relationship between a single item and the type of people that like that object. Content-based filtering approaches use descriptions of the content of the items to learn a relationship between a single user and the description of the items. Collaborative filtering approaches use the feedback of a set of people on a set of items to make recommendations, but ignore the content of the items or the descriptions of the people. Recently, researchers claim the outperformance of hybrid systems. Hybrid systems exploit features of content-based and collaborative filtering, since they will almost certainly prove to be complementary.

Traditional collaborative filtering systems employ a simplistic approach that directly recommends new items on the basis of the similarity among profiles of different users. This means that users with similar profiles exchange recommendations. However, when a similar user gives unsuccessful advice, there is no way of ignoring it. Over and over again this agent causes a descent in the performance of the other agents.

Marsh proposes the concept of trust to make our agents less vulnerable to others [8]. Trust is basic in any kind of action in an uncertain world; in particular it is crucial in any form of collaboration with other autonomous agents [1]. There is no standard definition for trust [5, 2]. Elofson gives a definition closer to our approach [3]. He claims that observations are important for trust, and he defines trust as:

"Trust is the outcome of observations leading to the belief that the actions of another may be relied upon, without explicit guarantee, to achieve a goal in a risky situation"

Elofson notes that trust can be developed over time as the outcome of a series of confirming observations (also called the dynamics of trust). From his experimental work, Elofson concludes that information regarding the reasoning process of an agent, more than the actual conclusions of that agent affect the trust in those conclusions.

Trust is formed and updated over time through direct interactions or through information provided by other members of society about experiences they have had. Each event that can influence the degree of trust is interpreted by the agent either as a negative or a positive experience. If the event is interpreted as a negative experience the agent will loose his trust to some degree and if it is interpreted to be positive, the agent will gain trust to some degree. The degree to which trust changes depends on the trust model used by the agent. This implies that the trusting agent carries out a form of continual verification and validation of the subject of trust over time.

When applying the concept of trust in the collaborative world approach, we can solve de problem that arises when a similar agent gives frustrated recommendations by decreasing the trust in this agent and ignoring its advice in the future. Trust provides, therefore, a new method for filtering information. Taking advantage of the communication among them, an agent can ask other agents for the opinion of a given item. It differs from the typical collaborative filtering approach in the way that the agent does not ask for a recommendation, but an opinion. The opinion is the interest that the other agent thinks that his/her user has about the given item. Instead of using this opinion directly as a recommendation, the agent includes it in its own reasoning and combines it with other agents' opinions in order to decide whether to recommend a given item. We call this new process of filtering information based on agents opinions the opinion-based information filtering method.

It is important to note that this new approach emphasizes proactiveness of agents. That is to say, when an agent has not enough knowledge to decide about a recommendation, it will turn to other agents on the web, in order to look for similar agents from which to gather information.

3 The Opinion-Based Information Filtering Method

The main idea is to consider other agents as personal entities which you can rely on or not. Reliability is expressed through a trust value with which each agent labels its neighbors. The trust value is initially computed through interaction, following a proactive *playing agents* procedure [15]. Each agent ask the other agents about a list of known items and gathers their opinion on such items. The agents ask the queried agents about their opinion on the item that the user either "loves" or "hates". According to similarity between the opinion provided and their own, agents are able to infer a trust value for each neighbor. Only the contact address of *friend* agents (i.e. agents with a high trust value) are kept.

Once the agent has a set of *friends*, it can use them to filter information. When the agent is not sure about a recommendation or discovers a new item, it asks the reliable agents for their opinion and uses their trust values to decide whether the item is interesting for the user or not (see Figure 1). Once the agent has the opinion of the other agents, a consensus is achieved through the use of an aggregation measure. The result of the consensus provides a confidence value upon which the agent can decide on the convenience of recommending an item to the user or not.

We suppose that similar agents will provide pertinent opinions, but they may also give inadequate ones. Trust, therefore, should be modified as goes by depending on the results of the recommendations, in order to improve acquaintance.

When applying an agent-based approach to recommender systems with trust in the collaborative world, the typical information filtering methods (content-based and collaborative filtering) can also be applied. The performance of the content-based filtering method is the same in this approach, but the collaborative filtering method is improved, since agents only believe in the recommendations of agents with a high trusting value. Finally, we get a hybrid approach among opinion-based, content-based and collaborative filtering.

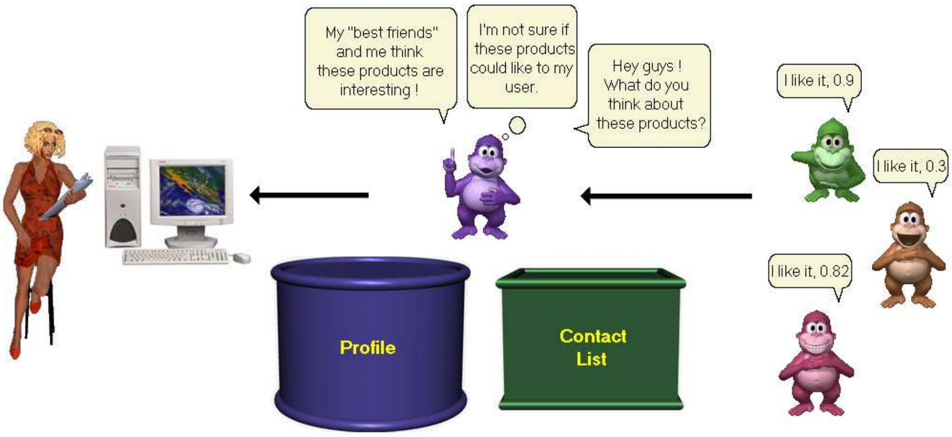

Fig. 1. Information Filtering based on Opinion

4 Social Trust Model for Recommender Agents

The opinion-based filtering method is based on a social model of trust that we describe following the main dimensions of recommender agents identified in [11]: user profile representation, initial profile generation, profile exploitation, relevance feedback, and profile adaptation.

4.1 User Profile Representation

The process of filtering information is based on user profiles which are somewhat hypothesis of unknown target concepts of user preferences. Recommender systems build and exploit these profiles. The construction of accurate profiles is a key task since the success of the system will depend to a large extent on the ability to represent the user's actual interests. Our model considers a user profile representation based on past experiences and a list of agents which the agent trusts. It is described as follows:

Given a set of agents: $A = \{a_1, a_2, \ldots, a_r\}$ and a set of products: $P = \{p_1, p_2, \ldots, p_s\}$. Each product is characterized by a set of objective attributes such as name, price, etc. Thus

$$p_i = \{at_{i_1}, at_{i_2}, \ldots, at_{i_n}\}$$

being At the set of all possible attributes.

Each agent can be interested in one product. Such interest can either be expressed by the user (explicit attributes) or be captured automatically by the system as a result of the user interactivity (implicit attributes). Explicit interests provide more confidence in the recommendation process. However, they are not always available. Implicit interests are useful to decide upon interesting items for the user. In our model we distinguish both kinds of user interactions: explicit from implicit, and therefore it is a hybrid approach. We name the set of explicit interest as

$$Int^e = \{int^e_1, int^e_2, \ldots, int^e_m\}$$

and the set of implicit interest as:

$$Int^i = \{int^i_1, int^i_2, \ldots, int^i_l\}$$

Both int^e_j and int^i_j are defined in [0,1].

Each agent has experiences in several products. An experience keeps information about the objective attributes of a given product, as well as subjective information regarding the interest of the user in that product.
Thus,

$$E_i =< p_i, Int^e_i, Int^i_i, \delta_i >$$

Where $p_i \subset P$ is the set of objective attributes of the product, $Int^e_i \subset Int^e$ is the set of explicit interest, $Int^i_i \subset Int^i$ is the set of implicit interest, and δ_i is a temporal parameter in [0,1] that indicates the relevance of the experience. Initially δ is set to 1, and it is updated according to the evolution of the agent. For the sake of simplicity we will not deal with this parameter in this paper; see [9] for further information.
Experience of agent a_i in product p_j is $E_{i,j}$, and the set of all possible experiences is denoted as $\mathcal{E}$.

For example, in the restaurant domain products and interests are represented as:

$$\begin{aligned} A_t = \{&name, address, phone\ number, cuisine,\\ &approximate\ price, capacity, web\ page\} \end{aligned}$$

$$\begin{aligned} Int^e = \{&general\ evaluation,\\ &quality/price\ relation,\\ &quantity\ of\ food\} \end{aligned}$$

$$\begin{aligned} Int^i = \{&web\ page\ visits\ rate,\\ &retrieved\ queries\ rate,\\ &rate\ of\ time\ spent\ on\ the\ web\ page\} \end{aligned}$$

A single experience of the user in a restaurant recommended by its agent is:

$$
\begin{aligned}
E =< \{"Mallorca\ Restaurant",\\
"2228\ East\ Carson\ St,\ Pittsburgh, PA",\\
"(412)4881818", "Spanish", "\$70", 300,\\
"www.mallorcarestaurant.com"\},\\
\{0.83, 0.76, 0.91\},\\
\{0.72, 0.36, 0.81\},\\
0.83 >
\end{aligned}
$$

Each agent a_i has a list of contact neighborhood agents on which it relies:

$$C_i = \{(a_{i_1}, t_{i,i_1}), (a_{i_2}, t_{i,i_2}), \ldots, (a_{i_n}, t_{i,i_k})\}$$

where $a_{i_j} \in A$ and t_{i,i_j} is a numerical value between [0,1] that represents the truth value the agent a_i has on agent a_{i_j}.

The set of all experiences of a given user and the set of selected agents that the agent trusts constitute the user profile:

$$Prof_i =< \mathcal{E}_i, C_i >$$

where $\mathcal{E}_i \subset \mathcal{E}$.

4.2 Initial Profile Generation

In order to start recommending to a user, the agent needs to fill in the user profile. Initial experiences are generated through the use of a training set. That is, the user is prompted to a set of products and he/she must fill in information regarding his/her interest in the products. We have chosen this technique because, as we will prove later, a training set provides the opportunity to calculate an initial trust for agents in the contact list. Other advantages and disadvantages of this kind of experience generation have been broadly discussed elsewhere, as for example in [11].

The training set consists of a collection of selected products $P^t \subset P$. For each product in the training set, the agent asks the user about the explicit interest and also gathers information related to implicit interests. Thus, the agent has an initial set of experiences.

The next step of the initial profile generation is to obtain *friend* agents for the contact list. Initially the list is empty. However, we assume that there is a server that provides the list of the currently available agents in the world that the agent runs. Such an assumption is reasonable taking into account that most of the multi-agent system platforms currently developed and FIPA [4] compliant provide such a service.

Then we elaborate the initial trust of agents in the world using a procedure that we have called *playing agents* following [15]. The querying agent asks other agents in the world (enquired agents), one by one about, an item of the training set. We can apply this procedure because each agent has been generated from the same training set, so they are able to provide answers about items belonging to such set. Then, the agent asks the enquired agent about the items that the user "loves" or "hates" (see Figure 2).

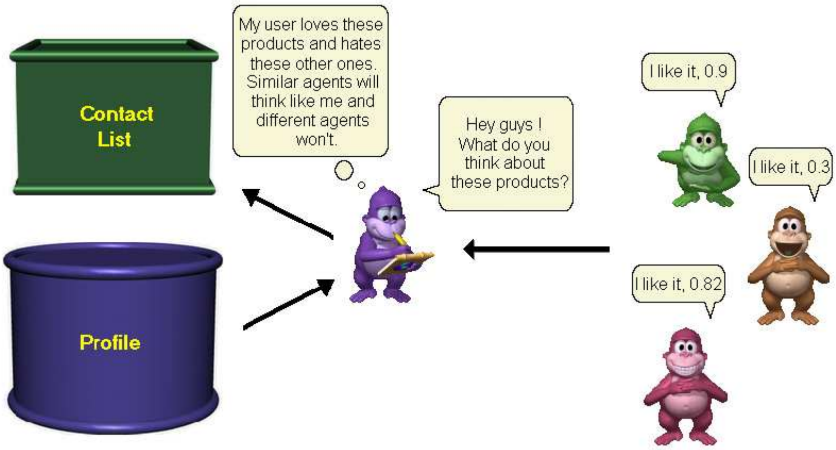

Fig. 2. "Playing Agents"

Note that the answer provided by the enquired agents does not consist of the set of their interest regarding the item asked about, since it would violate its privacy. The implementation of the *playing agents* procedure emphasizes interactiveness in the open world, but the information exchanged (the interest value) hides detailed information about the other users and preserves its personal data. Hence, the answer consists of a quantitative value, between 0 and 1, that represents the degree of interest the agent has in the product (0-hates, 1-loves). This *interest value* of an agent a_i in a product p_j, $v_{i,j}$ is calculated as follows:

$$v_{i,j} = \delta_j * g(f^e(Int_j^e), f^i(Int_j^i)) \tag{1}$$

where f^e is the function that combines the explicit interest of agent a_i in product p_j, f^i is the function that combines the implicit attributes, g is the function that combines the results of f^e and f^i, and finally δ_j is the temporal parameter related to the relevance of the product explained above. Aggregation techniques like [17] can be used for implementing f^e and f^i. For example, the Ordered Weighted Average (OWA) operator [20] is suitable because we are dealing with different preferences of the user and such preferences can be ordered according to their relative importance. The OWA operator is defined as follows:

$$f = \sum_{j=1}^{|p_i|} w_j * Int_j \qquad (2)$$

Where:

- $|p_i|$ is the cardinality of the product, that is, the number of attributes that characterizes it; and
- $\{\sigma(1), ..., \sigma(|p_i|)\}$ is a permutation of the values 1,...n so that $v_{i,\sigma(j-1)} \geq v_{i,\sigma(j)}$ $\forall j = 2, ...|p_i|$; in addition, the weights w_j are provided by an expert and must belong to [0,1] and $\sum_j w_j = 1$.

Finally, function g is a weighted arithmetic average (WA) that gives more importance to explicit attributes (objective ones) than to implicit ones (subjective):

$$g(e, i) = \lambda_e * e + \lambda_i * i \qquad (3)$$

For instance, we use $\lambda_e = 0.7$ and $\lambda_i = 0.3$.

Applied to the previous example on the experience of the user in a restaurant, we have the following values:

- First, the weights of the interests attributes are:

Explicit Attributes			Implicit Attributes		
j	$v_{i,\sigma(j)}$	w_j	j	$v_{i,\sigma(j)}$	w_j
1	0.91	0.5	1	0.81	0.5
2	0.83	0.33	2	0.72	0.33
3	0.76	0.27	3	0.36	0.27

- Second, the application of the OWA operator to aggregate the different explicit attributes and then the different implicit attributes is calculated as follows:

$$f_j^e = 0.5 * 0.91 + 0.33 * 0.83 + 0.27 * 0.76 = 0.93$$

$$f_j^i = 0.5 * 0.81 + 0.33 * 0.72 + 0.27 * 0.36 = 0.74$$

- Then, the application of the WA operator to aggregate explicit and implicit attributes is calculated as follows:

$$g(f_j^e, f_j^i) = 0.7 * 0.93 + 0.3 * 0.74 = 0.87$$

- Finally, the interest value is computed:

$$v_{i,j} = 0.83 * 0.87 = 0.72$$

	p_1	p_2	$\ldots$	$p_{\lvert P_t \rvert}$
a_{e_1}	$v_{e_1,1}$	$v_{e_1,2}$	$\ldots$	$v_{e_1,\lvert P_t \rvert}$
a_{e_2}	$v_{e_2,1}$	$v_{e_2,2}$	$\ldots$	$v_{e_2,\lvert P_t \rvert}$
$\vdots$				
a_{e_n}	$v_{e_n,1}$	$v_{e_n,2}$	$\ldots$	$v_{e_n,\lvert P_t \rvert}$

Table 1. Interest values gathered by the querying agent

The current querying agent, a_q, gathers a total of $|P^t|$ interest values of each enquired agent a_{e_i}, one for each product in the training set.

Then, the trust that agent a_q has in agent a_e, noted as $t_{q,e}$ is computed as follows,

$$t_{q,e} = \frac{\sum_{i=1}^{|P^t|} \delta_{p_i}(1 - |v_{q,i} - v_{e,i}|)}{\sum_{i=1}^{|P^t|} \delta_{q_i}} \tag{4}$$

This function computes the similarity between both agents, a_q and a_e, weighted by the relevance of the products (δ_{p_i}) according to a_q's interests (the querying agent). The result of the function is a normalized value in [0,1].

The agent only keeps the agents that have similar interests in the contact list. This is achieved by means of a fixed length contact list: only the n closest agents will be kept in the list. The *playing agents* procedure is repeated periodically in order to update the contact list according to the evolution of the user interests. Moreover, the trust value of each agent is updated as a result of a recommendation, as explained in section 4.5. In this sense, acquaintance among agents is improved over time.

Finally, we want to add that the number of agents in the collaborative world is also a matter of constraint in the *playing agents* procedure. That is, it will be very time-costly if any agent, in order to build a contact list, starts a *playing agents* procedure with all the agents in the world. For example, in a platform where agents recommend restaurants from Girona, up to 75.000 agents, one for each citizen, could be considered in the *playing agents* procedure. To reduce the number of agents to be queried, in each *playing agents* execution only a subset of all available agents is considered.

4.3 Profile Exploitation for Recommendation

The agent that recommends items to a user can receive a new product from its environment, or it can also proactively look for new products (for example, asking a server).

When an agent receives a new product, p_{new}, the agent computes the degree of similarity between the new product and the previous ones, according to the similarity measure based on the Clark's distance:

For all experiences E_p in the user profile,

$$sim(p_q, p_new) = \sqrt[2]{\sum_{i=1}^{|p_q|} \frac{|at_{q,i} - at_{new,i}|^2}{|at_{q,i} + at_{new,i}|^2}} \quad (5)$$

where $at_{p,i}$ is the i attribute of the product in the experience E_p and $at_{new,i}$ is the i attribute of the new product. Clark's distance is defined in [0,1] and has been proved useful in several domains. Then:

- If there is some product above a given threshold τ^+, the system recommends it. This process coincides with content filtering.
- If the best similar product is under a threshold τ^-, that means that the user has no interest in it and therefore the agent does not recommend it to the user.
- If the similarity of the products is in $[\tau^-, \tau^+]$ then the agent turns to the opinion filtering method to provide a recommendation.

The opinion filtering method consists of the following steps:

1. Ask the trustworthy agents in the contact list for their opinion on product p_{new}. For each enquired agent a_{e_i} a product value $v_{e_i,new}$ is calculated following equation 1.

	p_{new}
a_{e_1}	$v_{e_1,p_{new}}$
a_{e_2}	$v_{e_2,p_{new}}$
$\vdots$	
a_{e_n}	$v_{e_n,p_{new}}$

Table 2. Product interest values showed by the different enquired agents.

2. Compute a global value for the new product, r_{new} based on the opinion of all the queried agents. Since we are dealing with several sources of information an appropriate combination function is the weighted average (WA) where weights are the trust values of the agents. So,

$$r_{new} = \frac{\sum_i^{|C_q|} t_{q,i} * v_{e_i,new}}{\sum_i^n t_{q,i}} \quad (6)$$

where $t_{q,i}$ is the trust value that agent a_q has on the queried agent a_{e_i}; and $|C_q|$ is he cardinality of the contact list of the querying agent a_q.

If r_{new} goes above the τ^+ threshold, then the new product is recommended to the user.

It is important to note that if the enquired agents provide the interest values of the product, that is, $int_1^e, ...int_m^e$, and $int_1^i, ...int_l^i$, instead of an aggregated value, $v_{i,new}$, the information gathered by the querying agent will be richer and a more accurate decision can be made. For example, we can use Multicriteria Decision Making techniques (MCDM, [18]) based on the preferences of the querying agent. However, such information can be considered confidential in some environments. So in our approach privacy prevails over accuracy.

4.4 Relevance Feedback

To maintain the user profile, systems need relevant information regarding feedback of the recommendations given to the user. The most common way to obtain relevance feedback from the user is by means of the information given explicitly by the user and the information observed implicitly from the user's interaction with the web. In our model, this relevance feedback information is captured and kept in the Int^e and Int^i sets, included in each experience of users' profiles.

4.5 Profile Adaptation

Objective attributes of products often change, as for example the price. The user can also change his/her interest since human interests change as time goes by. Therefore, the same user can characterizze the same product with a different interest at different times. Then, the update of the user profile is required. In our model we have taken a lazy approach: we do not maintain the interest value of the product explicitly represented in the user profile. We compute it upon demand. Thus, the update process regarding product changes is costless, since it only consists in keeping either the new attribute of the product or the new interest of the user.

The key issue in adaptation is the relevance feedback from previous recommendations. If agents provide a recommendation based on the opinions of our "trustworthy" agents such trust should be updated according to the outcomes. Updating trust and trust dynamics is out of the scope of this paper and is explained in [10].

5 Related Work

There are very few approaches to trust in the collaborative world applied to the information filtering field. Knowledge Pump is an information technology system for connecting and supporting electronic repositories and networked communities [6]. Glance et al. introduce a technique that they call community-centered collaborative filtering (CCCF). In CCCF, the collaborative filter is bootstrapped by the partial view of the social network constructed from a user-input list of "advisors" (people whose opinion users particularly trust). The set of advisors is generated through statistical algorithms that mine the usage data automatically. The main difference from our model is the computation of the trust value since Glance bases it on the person-person correlation. So transparency of user data is required through agents, while in our system privacy prevails. The collaborative filter weighted higher the opinions of his/her most trusted contacts when predicting the user's opinion on items.

In other fields, such as electronic commerce, we can find other trust models that fit the particularities of the domains. For example, Schillo et al. present a formalization and an algorithm for trust so that agents can autonomously deal with deception and identify trustworthy parties in open systems [14]. They demonstrate with results that their approach helps each single agent to establish a model of trustworthiness of other agents. With only few iterations, agents learn who to trust and who to exclude from future interactions. They also show that agents form groups and play among themselves to profit from mutual support. Before that, they implemented a relevant computational method in the Social Interaction FrameWork (SIF) [13] in which an agent evaluated the reputation of another agent on the basis of direct observation and through other witnesses. The idea of using the opinion of other agents to build a reputation is also applied by Yu and Singh [21]. Their agents build and manage the trust representations not only taking into account the previous experiences of their users, but also communicating with other agents (belonging to other users). They aim at avoiding interaction with undesirable participants and formalizing the generation and propagation of the reputation in electronic communities.

6 Conclusions

The opinion-based filtering method dealing with an open environment such as Internet is a new approach that seems suitable for recommender agents. Like in the real world, agents rely on certain agents and mistrust others to achieve a purpose. If we provide agents with a technology to evaluate and trust other agents, agents can exploit the collaborative world with a better performance. The model presented in this paper is along this line. We have currently designed and developed a first prototype to test feasibility of the project [19]. Next, we plan to test the model and its advantages and disadvantages through experimentation.

From our point of view, the opinion-based filtering method can be considered as an evolution of the collaborative filtering methods due to the agent's world. If we consider that the hybrid approaches between content-based and collaborative filtering provide better results [11], we can consider this approach as an evolution of the information filtering methods in general (see Figure 3).

As future work, it is also important to show the cost of trust compared to traditional information filtering methods. We are also considering an extension of our model that would take into account the representation of user's interests through fuzzy values, in an attempt to make a more suitable measure. Moreover, we are currently analyzing the applicability of algorithms that automatically generate the different weights needed to apply aggregation measures, like the ones defined in [16]. These algorithms will provide flexibility to our model.

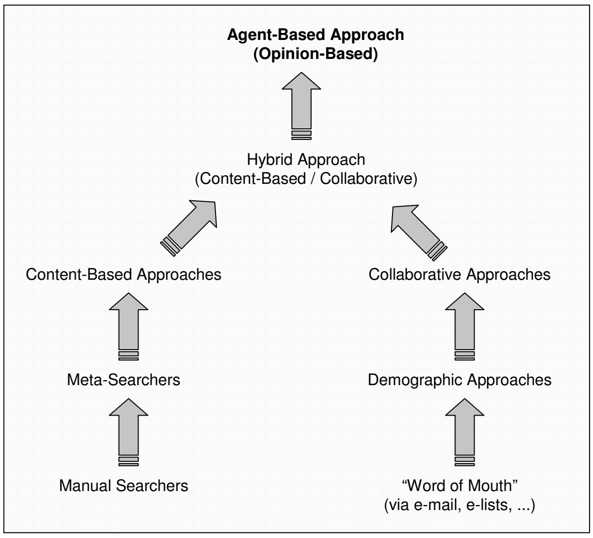

Fig. 3. Evolution of Information Filtering Methods

Acknowledgments

This research has been developed within the DAF-DAF Project supported by the CICYT grant DPI2001-2094-C03-01.

References

1. C. Castelfranchi. Information agents: The social nature of information and the role of trust, 2001. Invited Contribution at Cooperative Information Agents V (CIA'01). Modena (Italy).
2. C. Castelfranchi and R. Falcone. Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In *Demazeau, Y. (ed.), Proceedings of the Third International Conference on Multi-Agent Systems*, pages 72–79. IEEE Computer Society, Los Alamitos, 1998.
3. G. Elofson. Developing trust with intelligent agents: An exploratory study. In *Proceedings of the First International Workshop on Trust*, pages 125–139, 1998.
4. FIPA. http://www.fipa.org/specifications/index.html, 2001.
5. D. Gambetta. Can we trust trust? In *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Gambetta, D (editor). Basil Blackwell. Oxford, 1990.
6. N. Glance, D. Arregui, and M. Dardenne. Knowledge pump: Supporting the flow and use of knowledge. In *Information Technology for Knowledge Management*, pages 35–45. Eds. U. Borghoff and R. Pareschi, New York: Springer-Verlag, 1998.
7. M. Klusch. Information agent technology for the internet: A survey. In *Journal on Data and Knowledge Engineering, Special Issue on Intelligent Information Integration*, volume 36:6. D. Fensel (Ed.), Elsevier Science, 2001.
8. S. P. March. Formalising trust as a computational concept. In *Phd Thesis, Department of Computing Science and Mathematics, University of Stirling*, 1994.

9. M. Montaner. Personalized agents based on case-based reasoning and trust in the collaborative world. In *Thesis Proposal, University of Girona*, 2001.
10. M. Montaner, B. López, and J. L. de la Rosa. Developing trust in recommender agents. *Accepted as Poster at the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02). Palazzo Re Enzo (Italy)*, 2002.
11. M. Montaner, B. López, and J. L. de la Rosa. A taxonomy of recommender agents on the internet. In *Submitted to Artificial Intelligence Review*, 2002.
12. J. Sabater and C. Sierra. Regret: A reputation model for gregarious societies. In *Research Report. Institut d'Investigació i Intel.ligència Artificial*, 2000.
13. M. Schillo and P. Funk. Who can you trust: Dealing with deception. In *Proceedings of the Workshop Deception, Fraud and Trust in Agent Societies at the Autonomous Agents Conference*, pages 95–106, 1999.
14. M. Schillo, P. Funk, and M. Rovatsos. Using trust for detecting deceitful agents in artificial societites. In *Applied Artificial Intelligence, Special Issue on Trust, Deception and Fraud in Agent Societies*, 2000.
15. L. Steels and P. Vogt. Grounding adaptive language games in robotic agents. In *Proceedings of the Fourth European Conference on Artificial Life*, pages 473–484, 1997.
16. V. Torra. the learning of weights in some aggregation operators: The weighted mean and the owa operators. volume 6, pages 249–265. Mathware and Soft Computing, 1999.
17. V. Torra. On the integration of numerical information: from the arithmetic mean to fuzzy integrals. Torra V. (Ed). Information fusion in data mining. Physiin-Verlag. (Forthcoming), 2001.
18. A. Valls. Development of a method for multiple criteria decision making based on negation fuctions. chapter 2: State of the art. Thesis proposal, Artificial Intelligence Program, UPC, 2000.
19. R. Vilà and M. Montaner. Implementació d'un sistema multiagent distribuit format per agents personals que recomanen restaurants aplicant raonament basat en casos i tècniques de trust, 2002. Projecte Fi de Carrera en Enginyeria Informàtica, Universitat de Girona.
20. R. R. Yager. On ordered weighted averaging aggregation operators in multi-criteria decision making. volume 18, pages 183–190. IEEE Transactions on SMC, 1988.
21. B. Yu and M. P. Singh. A social mechanism of reputation management in electronic communities. In *cooperative information agents, CIA-2000*, pages 154–165, 2000.