# Coping with Information Overload through Trust-Based Networks

Frank E. Walter, Stefano Battiston, Frank Schweitzer

Chair of Systems Design, ETH Zurich Kreuzplatz 5, 8032 Zurich, Switzerland
fewalter|sbattiston|fschweitzer@ethz.ch

## 1 Introduction

### 1.1 Motivation

Over the recent decade, the Internet has conquered people's homes and life: they pursue an increasing amount of activities on the World Wide Web and this has fundamentally impacted the lifestyle of society. For example, people use their computers for communication with others, to buy and sell products on-line, to search for information, and to carry out many more tasks. Along this development, so far unknown ways of marketing, trading and information sharing are booming. This situation is made possible by a set of related emerging technologies centred around the Internet – just to mention a few: collaborative work and information sharing environments, peer-to-peer networks , and rating, recommendation, and reputation systems. At the economic level, the impact of these technologies is already very high and it is expected to grow even more in the future. The Internet has become a social network, "linking people, organisations, and knowledge" [2] and it has taken the role of a platform on which people pursue an increasing amount of tasks that they have usually only done in the real-world. An approach looking at these emerging technologies and their effects from a complex systems perspective can, as we will show in this chapter, be very useful.

### 1.2 Emerging Technologies

In the following, we will look at the particular technologies already mentioned – collaborative work and information sharing environments, peer-to-peer

* The model discussed in this chapter is based on our paper of a trust-based recommendation system on a social network, see [1]. For more formal and detailed descriptions of the model, the analysis, and the simulations, please also refer to this paper. For further materials on our research in this area, please see our website www.sg.ethz.ch/research.

networks, and rating, recommendation, and reputation systems – in more detail. We will demonstrate that collaborative work and information sharing environments are tools to create vast amounts of globally available information; in addition, peer-to-peer networks help to quickly spread this information over large distances. This leads to the situation that people are confronted with an *information overload*; one possible solution to this problem lies, as we will demonstrate, in rating, recommendation, and reputation systems.

### Collaborative Work and Information Sharing Environments

Collaborative work and information sharing environments have created platforms where people are able to share knowledge, tastes, bookmarks etc. An example of such a system would be `wikipedia.org`, a free on-line encyclopedia which can be accessed and edited by anyone on the web. Over the recent years, Wikipedia has grown manifold and now is considered a real challenge for the established encyclopedias available both on-line or as books. Wikipedia is an example of a whole range of websites that act as the platform for people making information available to others – there are many more, for example `delicious.com`, a repository for the bookmarks of people, `citeulike.org`, where people can make their bibliographies and literature lists available, `ohmynews.com`, which is an online newspaper with the motto "every citizen a reporter" and where anyone can contribute articles, and many, many more.

### Peer-to-Peer Networks

At the same time, peer-to-peer networks have become very popular because they enable users to share information, typically digital content. Peer-to-peer networks are inherently distributed in the sense that they do not require a central server which coordinates clients but rather that nodes self-organise and adapt to change. This makes it very difficult to attack peer-to-peer networks (i.e., this includes attempts to take them off the network). Furthermore, they reflect the structure of social networks in the real life. The simplicity to duplicate and share digital content combined with the ineffective implementations of digital rights management platforms has caused some to suggest the revision of the notion of intellectual property. Several approaches can be thought of in the framework of these emerging technologies: for instance, a rating system of the digital content would allow to compensate authors based on the aggregate rating of the items that they offer. Nonetheless, the core feature of peer-to-peer networks is that they provide a medium to spread information without boundaries in space and time.

### Information Overload

Now, the technologies mentioned so far – collaborative work and information sharing environments as well as peer-to-peer networks – confront people with an *information overload*: they are facing too much data to be able to effectively filter out the pieces of information that are most appropriate for them. The exponential growth of the Internet [3] implies that the amount of information accessible to people grows at a tremendous rate. Historically, people have – in various situations – already had to cope with information overload and they have intuitively applied a number of *social mechanisms* that help them deal with such situations. However, many of these, including the notion of trust, do not yet have an appropriate *digital mapping* [4]. Finding suitable representations for such concepts is a topic of on-going research [6, 5, 7, 8, 9] across disciplines.

### Rating, Recommendation, and Reputation Systems

The problem of information overload has been in the focus of recent research in computer science and a number of solutions have been suggested. The use of search engines [10] is one approach, but so far, they lack personalisation and usually return the same result for everyone, even though any two people may have vastly different preferences and thus be interested in different aspects of the search results. A different proposed approach are rating, recommendation, and reputation systems [13, 11, 12]:

- *Rating systems* allow users to post their rating on items, which are then ranked according to the aggregate rating in the system. An example would be `ciao.com`, a website which allows to do product and price comparisons. The obvious drawback of such systems in which the aggregate rating is made the benchmark is that users with preferences deviating from the average will find the rating unsatisfactory for them.
- *Recommendation systems* based on collaborative filtering suggest users items based on the similarity of their preferences to other users. For example, on `amazon.com` users are often presented the message that “people who bought [a particular] book also bought these other books” followed by a list of related books. This kind of recommendation system works quite well for low-involvement items such as books, movies or alike. Many scientific teams are working on the data mining aspect, but the few works based on complex systems theory seem particularly promising [26], [25]. Furthermore, the combination of collaborative filtering with trust is one the hot topics in computer science in the near future [28], and again a complex systems approach is proving to be quite successful [32], [1]. In such recommendation systems, the fact that information is processed in a centralised way raises scalability issues. However, more importantly, if ratings concerned high-involvement services, such as health care, insurance, or financial services, centralisation also raises confidentiality issues. As we will see in the following, these limitations can be overcome with trust-based networks.

- *Reputation systems* are used more and more in trading. Possibly the most prominent example is `ebay.com`, the Internet auction platform where both buyers and sellers have an associated reputation value which reflects their reliability, quality-of-service, and trustworthiness. Such notions of reputation are gaining visibility – even to the point that people post their `ebay.com` reputation value on their curriculum vitae when looking for a job. However, there are several unsolved game theoretic drawbacks to such systems, for instance the incentive to give good ratings in order to avoid retaliation.

Figure 1 illustrates the use cases of such recommendation systems along the example of `amazon.com`. In the example, a user is searching for a travel guide to Switzerland. The recommendation system is used to establish a ranking of potential books to be bought and to facilitate the decision making of the user.

### 1.3 Applications to Business and Society

As we have seen from the examples, these concepts have formed the basis for recently founded businesses all over the world. This demonstrates their high impact at the global economic level. Moreover, the current trend is that the sector is continuously expanding with the foundation of new start-ups. However, the impact is not limited to the business world, but also affects society. For the first time in history, a large-scale real-time self-organisation of citizens in previously unknown forms is possible. For instance, now it is more straightforward for consumers to reach and share ratings of products independently of the producers. It is also feasible, for example, that groups of consumers form buying groups that negotiate with firms the delivery of products or services with specific features. Market diversity , which, today, is a producer-driven process, could become a consumer-driven process, a major change of perspective. In particular, the market share for sustainable products and services could increase significantly. In particular, the application of recommendation systems and akin is not limited to targeted marketing. On the contrary, there is an unprecedented potential for empowering citizens to make more informed choices in their daily life in a vast range of domains, from grocery purchases to political support.

### 1.4 Role of Complex Systems Theory

The important aspect from the perspective of complex systems theory is that these developments give rise to large-scale collective dynamics. While computer science research in this field mainly focusses on aspects such as protocols, algorithms, security, and infrastructure, the theoretical understanding of the large-scale emerging properties is poor. Research from a complex systems perspective can and should give important contributions to better understand these developments with respect to collective dynamics.

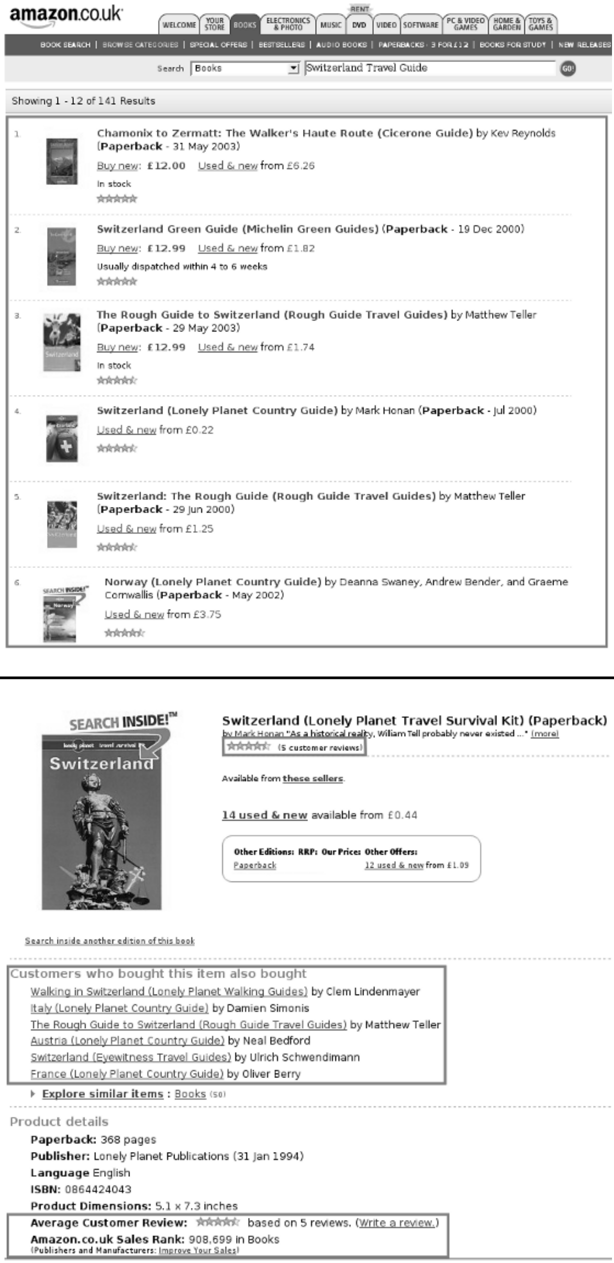


**Fig. 1.** Amazon, an example of a recommendation system. In the example on top, recommendations are used to *rank particular items in a category*, e.g. books that claim to be travel guides to Switzerland, and in the example at bottom, recommendations are used to *make choices based on ratings* that they provide, e.g. whether to buy/not to buy a particular book. Note the erroneous result on Norway in the list

### 1.5 Trust-based Networks

The complex systems approach offers a promising way to cope with all these mentioned challenges: *trust-based networks*. A trust-based network can be defined as an information processing system in which interconnected agents (citizens, firms, organisations) share knowledge in their domains of interest. Each agent has a set of neighbours – e.g., friends, partners, and collaborators – with which it decides to share lists of products, services, people, experts etc. together with ratings on these. Trust between neighbours is built up dynamically, based on the satisfaction experienced from the recommendations received by these neighbours.

Soon, paths of trust build up in the network, and each agent is able to reach and rely on – filtered – information, even if coming from another agent far away in the network. This emerging property has some reminiscence to the building of optimal paths in ant colonies [27, 29]. Some recent works have proven the overwhelming superiority of such trust-based recommendation systems over those based on the frequency the recommendations [32]. From the point of view of scalability, trust-based networks are inherently distributed in their nature and do not require centralised information.

A trust-based network can be regarded as an IT support tool for decision making shaped around the natural behaviour of individuals in society. Today's search engines allow the user to find a range of information/products/services from a centralised source, corresponding to a set of keywords. Through a trust-based network, an agent can, instead, search relevant items from specialised, distributed sources and evaluate the trustworthiness of the items with respect to its own preferences.

Subsequently, we present an example of a trust-based network by illustrating a model of a trust-based recommendation system. This system, in an automated and distributed fashion, filters information for agents based on the agents' social network and trust relationships.

The model that we are going to present enables a quantitative study of the problem and also provides a sketch for a solution in terms of a real Internet application/web service. The idea at the core of the model is that agents

- leverage their social network to *reach information*; and
- make use of trust relationships to *filter information*.

We describe the model and the results obtained through multi-agent simulations. To some extent, it is also possible to make analytical predictions of the performance of the system as a function of the preferences of the agents and the structure of the social network. In the following, we will refrain to go into all the details of the model and stay at the level of an informal treatise; for more formal and detailed descriptions of the model, the analysis, and the simulations, please also refer to [1].

The remainder of the chapter is organised as follows: in the following section, we put our work into the context of the related work. Subsequently, we describe an illustrative example of a situation in which a user could benefit from the use of a trust-based network. Then, we present our model of a trust-based recommendation system on a social network. This is followed by a summary of the results from computer simulations and analytical approximations as well as their interpretations. Subsequently, we outline an application of the model. Finally, we illustrate a number of extensions.

## 2 Related Work

Recent research in computer science has dealt with recommendation systems [11]. Such systems mostly fall into two classes: content-based methods suggest items by matching agent profiles with characteristics of products and services, while collaborative filtering methods measure the similarity of preferences between agents and recommend what similar agents have already chosen [38]. Interestingly, some of the achievements in this field come from the community of complex systems research [26, 25]. Often, recommendation systems are centralised and, moreover, they are offered by entities which are not independent of the products or services that they provide recommendations on, which may constitute a bias or conflict-of-interest.

Additionally, the diffusion of information technologies in business and social activities results in intricate networks of electronic relationships. In particular, economic activities via electronic transactions require the presence of a system of trust and distrust in order to ensure the fulfilment of contracts [4, 9]. However, trust plays a crucial role not only by supporting the security of contracts between agents, but also because agents rely on the expertise of other trusted agents in their decision-making.

Along these lines, some recent works have suggested to combine distributed recommendation systems with trust and reputation mechanisms [13, 12, 31, 39, 28]. Because of the fact that both building expertise and testing items available on the market are costly activities, individuals in the real world attempt to reduce such costs through the use of their social/professional networks.

Such complex networks, in particular their structure and function, are the subject of an extensive and growing body of research across disciplines [17]. In particular, it has been shown that the structure of social networks plays an important role in decision making processes [22, 23, 24].

In this chapter, we combine these three approaches – recommendation systems, trust, and social networks – along the lines of [1].

## 3 Illustrative Example

The situation we want to model could be illustrated by the following scenario: a person needs to buy a bottle of Swiss wine to accompany an evening with cheese fondue and, just having moved to the country, does not know which one to choose. Therefore, the person contacts its friends and asks them for advice. The friends either have a piece of advice or they pass the question on to their own friends. Let us assume that there are several brands of wine to choose from: $\{a, b, c, ...\}$. After some time, the person receives a number of recommendations, say 6 in number, for specific brands to choose from. For instance, there could be

- 3 recommendations that suggest brand $a$,
- another 2 that suggest brand $b$, and
- 1 that suggests brand $c$.

How is it possible to make the best use of the recommendations? One might choose brand $a$ because it is the most frequently recommended, but it may also be that brand $c$ has been recommended by a friend of a friend who is known to be an expert in wines. Now, there is a trade-off – should one rely on the opinion of the majority or on the opinion of an expert? For an person with average preferences, the opinion of the majority, i.e. the "average opinion" might do well. However, if the preferences of the person deviate from the average in its community, following the advice of an expert may be much more useful.

Let us assume, for the moment, that the person decides for brand $a$ because it is the most frequently recommended choice. However, upon consumption, it discovers that this brand does not match its taste at all. Now, it may make sense that, at the next time when the person goes shopping for wine, it gives less importance to the recommendations of those agents that recommended brand $a$ and that it may even try brand $b$ or $c$. By following such a strategy, the person would, over time, learn which other people give reliable advice with respect to a particular context and which do not.

Note that in order for this system to work, all people concerned need to have identical definitions of the concept "wine": whenever they exchange recommendations on wine, they know that they all are talking of the same concept.

However, consider that the person now also requires a recommendation on which brand of cheese to buy for the fondue. By the same procedure as for the wine, it obtains a number of recommendations, some from the same people that also made recommendations for the wine. Should the experiences made with the former recommendations on wine influence the decision of which recommendation on cheese to follow? Certainly this must not necessarily be the case: for example, the expert on wines may give good recommendations on wine, but since he is not at all experienced in cheese, his recommendations on cheese may be completely useless. In other words, there may be some contexts in which people may follow recommendations by certain friends and other contexts in which they may not follow the recommendations by the same friends.

What people intuitively do in real life is to keep a mental mapping of the level of trust that they have towards the advice of friends in a particular context. However, this is a difficult task when the market offers thousands of product and service categories as well as dozens of brands in each category. Certainly, the recent developments in the field of information technology make it both desirable and possible to automate this process by means of a computer-assisted recommendation system.

## 4 Model Description

In the following, we describe an example of a trust-based network by illustrating a model of a trust-based recommendation system. The model deals with agents which have to decide for a particular item that they do not yet know based on recommendations of other agents. When facing to purchase an item, agents query their neighbourhood for recommendations on the item to purchase. Neighbours in turn pass on a query to their neighbours in case that they cannot provide a reply themselves. In this way, the network replies to a query of an individual by offering a set of recommendations. One way to deal with these recommendations would be to choose the most frequently recommended item. However, because of the heterogeneity of preferences of agents, this may not be the most efficient strategy in terms of utility. Thus, we explore means to incorporate knowledge of trustworthiness of recommendations into the system. In the following, we investigate under which conditions and to what extent the presence of a trust system enhances the performance of a recommendation system on a social network.

### 4.1 Agents, Objects, and Profiles

We consider a set $S_A$ of $N_A$ agents $a_1, a_2, a_3, ..., a_{N_A}$. The agents are connected in a *social network* such as, for example, a social network of people and their friends [15, 16, 17] that are recommending books to each other. Hence, each agent has a set of links to a number of other agents (which we call its neighbours). In reality, social networks between agents to evolve over time; in other words, relationships form, sustain, and also break up. In this chapter, we mainly focus on a static network while dynamic networks will be investigated more thoroughly in further work. At this stage, we assume the network to be described by a random graph [34].

Furthermore, there exists a set $S_O$ of $N_O$ objects, denoted $o_1, o_2, o_3, ..., o_{N_O}$. These objects represent items, agents, products, buyers, sellers, etc. – anything that may be subject to the recommendations, for example, books. We further assume that objects are put into one or more of $N_C$ categories from $S_C$, denoted $c_1, c_2, ..., c_{N_C}$, where these categories are defined by the system and cannot be modified (i.e. added, removed, or redefined) by the agents. In a scenario where the recommendation system is on books, categories could be

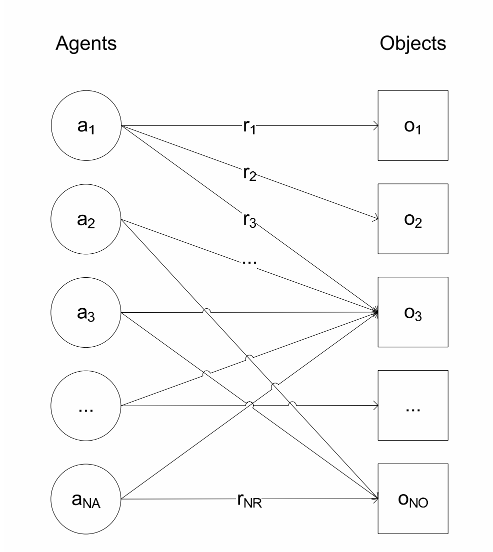


**Fig. 2.** Agents rating Objects: this is a bipartite graph with the agents on the left hand side and the objects on the right hand side, the ratings being the connections. The set of all possible ratings of an agent constitutes its respective profile [1]

books on 'epicurean philosophy', 'Swiss folklore', or 'medieval archery'. We denote the fact that an object $o_i$ is in category $c_j$ by stating $o_i \in c_j$.

Each agent $a_i$ is associated to one certain preference profile which is one of $N_P$ preference profiles in the system, where $S_P = \{p_1, p_2, p_3, ..., p_{N_P}\}$. In the following, we will use the terms 'preference profile', 'profile', and 'preferences' interchangeably. Such a profile $p_i$ is a mapping which associates to each object $o_j \in S_O$ a particular corresponding rating $r_j \in [-1, 1]$, $p_i : S_O \rightarrow [-1, 1]$. This is illustrated in Figure 2. In the current version of the model, we only consider discrete ratings where $-1$ signifies an agents' dislike of an object, 1 signifies an agents' favour towards an object. In a future version of the model, this assumption can be relaxed; we chose to initially focus on a discrete rating scheme because most of the ones found on the Internet are of such type. We assume that agents only have knowledge in selected categories and, in particular, they do only know their own ratings on objects of other categories subsequent to having used these objects. Thus, each agent is and remains an expert only on a set of initially assigned selected categories.

### 4.2 Trust Relationships

In this model, we also consider trust relationships between agents: each agent $a_i$ keeps track of a trust value $T_{a_i,a_j} \in [0, 1]$ to each of its neighbour agents $a_j$. These values are initialised to $T_{a_i,a_j} = 0.5$. It is important to stress that trust relationships only exist between neighbours in the social network; if two agents are not directly connected, they also cannot possibly have a trust relationship with each other. However, two such agents may indirectly be connected to each other through a path in the network. For example, agent $a_i$ could be connected to agent $a_j$ through agents $a_k$ and $a_l$, should $a_k$ and $a_l$, $a_i$ and $a_k$, as well as $a_l$ and $a_j$ be neighbours. We can then compute a trust value along the path $\mathrm{path}(a_i, a_j)$ from $a_i$ to $a_j$ – in the example, $\mathrm{path}(a_i, a_j) = \{(a_i, a_k), (a_k, a_l), (a_l, a_j)\}$ – as follows:

$$T_{a_i,\ldots,a_j} = \prod_{(a_k,a_l)\in \mathrm{path}(a_i,a_j)} T_{a_k,a_l} \tag{1}$$

i.e. the trust value along a path is the product of the trust values of the links on that path. Figure 3 illustrates a part of such a social network of agents and a chain of trust relationships between two agents.

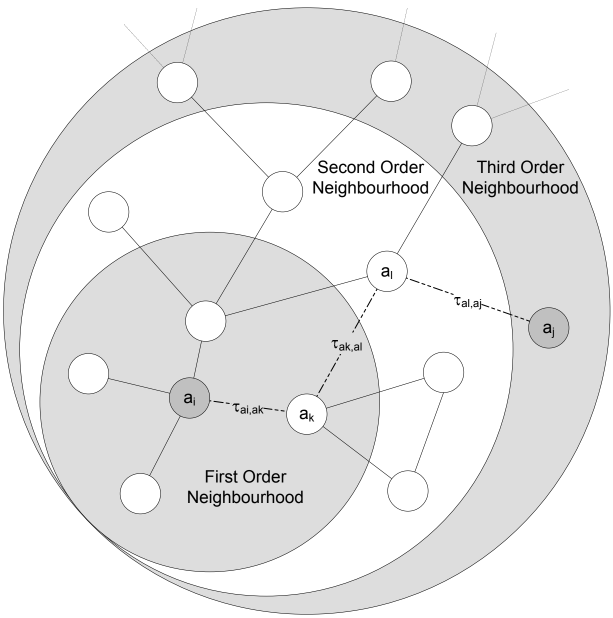


**Fig. 3.** Social Network of Agents and their Trust Relationships: a section of the social network around agent $a_i$, indicating a chain of trust relationships to agent $a_j$ and ordering the neighbours according to their distance in hops ('orders of neighbourhood') [1]

### 4.3 Temporal Structure, Search for Recommendations

The model assumes a *discrete linear bounded model of time*. In essence, there are two possible types of search for a recommendation:

1. *Ranking within a category (RWC)*: agents query for a particular category and search recommendations for several objects in this category in order to decide for one of the recommended objects in the response from the network – typically the best one.
2. *Specific rating for an object (SRO)*: agents query for a particular object and search recommendations on this very object in order to decide for or against using it, based on the response from the network.

Of these, the RWC is a superset of the SRO; a system which can provide a RWC can trivially be extended to provide SRO, too. Hence, in the following, we focus on the former rather than the latter.

At each time step $t$, each agent $a_i$ (in random order) selects a category $c_j$ (again, in random order, with the constraint that the agent is not an expert on the category) and searches for recommendations on the network. In informal notation, the protocol for the agent's search proceeds as follows:

1. Agent $a_i$ prepares a query$(a_i, c_j)$ for category $c_j$ and then transmits it to its neighbours.
2. Each neighbour $a_k$ receives query$(a_i, c_j)$ and either
   a) returns a response$(a_i, a_k, (o_j, r_j), T_{a_i,\ldots,a_k})$, if it knows a rating $r_j$ for a particular object $o_j$ in $c_j$ that it can recommend, i.e. if $p_k(o_j) = r_j > 0$:
   b) or, passes query$(a_i, c_j)$ on to its own neighbours if it does not know a rating $r_j$ for the particular category $c_j$.

It is assumed that agents keep track of the queries they have seen. Now there are two strategies to guarantee that the algorithm terminates: either,

- agents do not process queries that they have already seen again ("incomplete search", IS); or,
- agents pass on queries only once, but, if they have an appropriate recommendation, can return responses more than once ("complete search", CS).

In essence, both are a form of *breadth-first search* on the social network of agents, but with different properties: the former returns, for each possible recommendation, only one possible path in the network from the querying to the responding agent; the latter, however, returns, for each possible recommendation, each of the possible paths in the network from the querying to the responding agent.

As we will see later, this is a crucial difference for the decision making of agents. For a given recommendation, there might be several paths between the querying and the responding agent. The IS returns a recommendation along one of these paths, while the CS returns a set of recommendations along all possible paths. Some paths between two agents have high trust, some have low trust. The IS may return a recommendation along a low-trust path even though there exists a high-trust path, thus providing an agent with insufficient information for proper decision making. Of course, there is also a pitfall with the CS – it is computationally much more expensive.

### 4.4 Decision Making

As a result of a query, each agent $a_i$ possesses a set of responses from other agents $a_k$. It now faces the issue of making a decision on the ratings provided. The agent needs to decide, based on the recommendations in the response, what would be the appropriate choice of all the objects recommended. We denote query$(a_i, o_j) = Q$ and a response$(a_i, a_k, (o_j, r_j), T_{a_i,\ldots,a_k}) \in R$ where $R$ is the set of all responses. The values of trust along the path provide a ranking of the recommendations. There are many ways of choosing based on such rankings; we would like to introduce an exploratory behaviour of agents and an established way of doing so consists in choosing randomly among all recommendations with probabilities assigned by a logit function [30]. For this purpose, it is convenient to first map trust into an intermediate variable $\hat{T}$, ranging in $[-\infty, \infty]$:

$$\hat{T}_{a_i,\ldots,a_k} = \frac{1}{2}\ln\left(\frac{1+2(T_{a_i,\ldots,a_k}-0.5)}{1-2(T_{a_i,\ldots,a_k}-0.5)}\right) \in [-\infty, \infty] \tag{2}$$

$$P(\text{response}(a_i, a_k, (o_j, r_j), T_{a_i,\ldots,a_k})) = \frac{\exp(\beta\hat{T}_{a_i,\ldots,a_k})}{\sum_R \exp(\beta\hat{T}_{a_i,\ldots,a_l})} \in [0,1] \tag{3}$$

where $\beta$ is a parameter controlling the exploratory behaviour of agents. For $\beta = 0$, the probability of choosing each response will be the same (i.e. this is equivalent to a random choice), but for $\beta > 0$, responses with higher associated values of $Ta_i, ..., a_k$ have higher probabilities. To decide for one of the objects, the agent chooses randomly between all recommendations according to these probabilities. This process is illustrated in Figure 4.

For benchmarking the trust-based approach of selecting recommendations, we consider an alternative decision making strategy, namely a *frequency-based approach* without any trust relationships being considered at all. In this approach, an agent chooses randomly among each of the recommendations with equal probability for each of the recommendations.

### 4.5 Trust Dynamics

In order to enable the agents to learn from their experience with other agents, it is necessary to feedback the experience of following a particular recommendation into the trust relationship. This is done as follows: subsequent to an

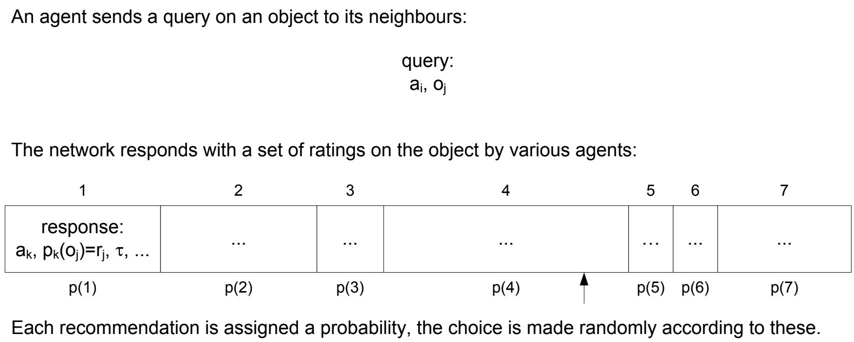


**Fig. 4.** Search for Recommendations and Decision Making: agents send queries, they receive responses, and then decide for one randomly according to probabilities they have assigned to each recommendation [1]

interaction, agent $a_i$ who has acted on a rating through its neighbour, agent $a_j$, updates the value of trust to this neighbour, based on the experience that he made. Let $o_k$ be the chosen object. Then, assuming agent $a_i$ having profile $p_i$, $p_i(o_k) = r_k$ is the experience that $a_i$ has made by following the recommendation transmitted through $a_j$. It is convenient to define the update of $T(t+1)$ in terms of an intermediate variable $\tilde{T}(t+1)$:

$$\tilde{T}_{a_i,a_j}(t+1) = \begin{cases} \gamma \tilde{T}_{a_i,a_j}(t) + (1-\gamma) r_k \text{ for } r_k \geq 0 \\ (1-\gamma)\tilde{T}_{a_i,a_j}(t) + \gamma r_k \text{ for } r_k < 0 \end{cases} \tag{4}$$

where $\tilde{T}_{a_i,a_j}(0) = 0$ and $\gamma \in [0,1]$. Because $\tilde{T}_{a_i,a_j} \in [-1,1]$, we have to map it back to the interval $[0,1]$:

$$T_{a_i,a_j}(t+1) = \frac{1 + \tilde{T}_{a_i,a_j}(t+1)}{2} \in [0,1] \tag{5}$$

The distinction between $r_k \geq 0$ and $r_k < 0$ creates, for values of $\gamma > 0.5$, a slow-positive and a fast-negative effect which usually is a desired property for the dynamics of trust: trust is supposed to build up slowly, but to be torn down quickly. The trust update is only applied between neighbouring agents – the trust along a pathway between two non-neighbour agents $T_{a_i,\ldots,a_j}$ changes as a result of changes on the links of the path. The performance of the system results from the development of pathways of high trust and thus is a emergent property of local interactions between neighbouring agents.

It is important to note that – in the current version of the model – trust turns out to reflect the similarity of agents. In further extensions of the model, it should reflect other notions such as "agent $a_j$ cooperated with agent $a_i$", "agent $a_j$ gave faithful information to agent $a_i$", or "agent $a_j$ joined a coalition with agent $a_i$". In other words, the metric should be an aggregate of different dimensions of trust, possibly measuring the faithfulness, reliability, availability, and quality of advice from a particular agent.

### 4.6 Utility of Agents, Performance of the System

In order to quantitatively measure the difference of the trust-based approach of selecting recommendations as compared to the frequency-based approach, it is necessary to define measures for the utility of agents as well as for the performance of the system.

We define an instantaneous utility function for an agent $a_i$ following a recommendation from agent $a_j$ on object $o_k$ at time $t$ as follows:

$$u(a_i, t) = r_i \tag{6}$$

where agent $a_i$'s profile determines $p_i(o_k) = r_i$. We consider the performance of the system to be the average of the utilities of the agents in the system:

$$\Phi(t) = \frac{1}{N_A} \sum_{a_i \in S_A} u(a_i, t) \tag{7}$$

This gives us a measure for quantitatively comparing the difference that the trust-based approach makes towards the frequency-based approach, both on the micro-level of an agent and the macro-level of the system. In the following, we will use the instantaneous measures for utilities and performance rather than the cumulative ones (if not indicated otherwise).

## 5 Results and Interpretation

One of the most important results of the model is that the system self-organises in a state with performance near to the optimum. Despite the fact that agents only consider their own utility function and that they do not try to coordinate, long paths of high trust develop in the network. This allows agents to rely on recommendations from agents with similar preferences, even when these are far away in the network. Therefore, the good performance of the system is an emergent property, achieved without explicit coordination.

### 5.1 Key Quantities

Three quantities are particularly important for the performance of the system: the network density, the preference heterogeneity among the agents, and the sparseness of knowledge. The core result is that recommendation systems in trust-based networks outperform frequency-based recommendation systems within a wide range of these three quantities:

- *Network density*: if the network is very sparse, agents receive useful recommendations on only a fraction of the items that they send queries about; the denser the network, the better the performance, but above a critical threshold for the density, the performance stabilises. The proximity of this value to the optimum depends on the other two quantities.
- *Preference heterogeneity*: if the preferences of agents are homogeneous, there is no advantage for filtering the recommendations; however, if the preferences of agents are all different, agents cannot find other agents to act as suitable filters for them. In between, when preferences are heterogeneous, but 'not too much', the system performance can be near to the optimum.
- *Knowledge sparseness*: when knowledge is dense ($N_c$ and/or $N_p$ small), it is easy for an agent to receive recommendations from agents with similar preferences. In the extreme situation in which, for each category there is only one expert with any given preference profile, agents can receive useful recommendations on all categories only if there exists a high-trust path connecting any two agents with the same profile. This is, of course, related to the density of links in the network.

The performance of the system thus depends, non-linearly, on a combination of these three key quantities. Under certain assumptions, the model can be investigated analytically and in a mean-field approach it is possible to make quantitative predictions on how these factors impact the performance. These results are presented in [1]. Here, we illustrate the properties of our recommendation system by describing the results of multi-agent simulations of the model. As a benchmark, we compare the trust-based recommendation system to a frequency-based recommendation system.

### 5.2 Simulation Parameters

For the simulations we have used the following parameters to the model: we consider $N_a = 100$ agents, and the simulations are averaged over $N_r = 100$ runs. The size of each category is the same and we vary $N_c \in \{10, ..., 50\}$ and $N_p \in \{2, 4, 6\}$; $N_o$ is usually adjusted such that there are at least 2 objects in each category. Profiles are distributed such that the sum over a profile is 0 on average – across the profile, categories, and agents. Each agent is an expert on one category. Further, for the social network we assume a random directed graph with a given number of agents, $N_a$, and a given total number of links, $\ell$. The *network density* is then defined as $p = \ell/N_a(N_a - 1)$. Agents are connected randomly with respect to their profile.

### 5.3 Trust and Decision-Making Dynamics

Figure 5 (left) shows that the update rule of trust as described by eq. 4 and eq. 5 produces a slow-positive fast-negative dynamics. Trust between two agents of the same profile evolves to 1 (red line, partially covered by the green one). Trust between two agents of opposite profiles evolves to 0 (blue line). In case that an agent recommends an object that is rated negatively, trust drops quickly and recovers slowly (green line). The probability of choosing a recommendation depends critically on the parameter $\beta$, which controls the exploratory behaviour of agents, as shown in Figure 5 (right).

### 5.4 Performance over Time and Role of Learning

Over time, each agent develops a value of trust towards it neighbours which reflects the similarity of their respective profiles. After some time, paths of high trust develop, connecting agents with similar profiles. As a result, the performance of the system, as defined in eq. 7 increases over time and reaches a stationary value which approaches the optimum. This is shown in Figure 6, where coloured curves correspond to different values of $\gamma$.

We have also simulated a situation in which, prior to the start of the dynamics, there is a learning phase in which the agents explore only the recommendations of their direct neighbours on the categories that these claim to be expert on. This way, the trust dynamics already start from a value deviating from the neutral point of 0.5 and closer to one of the fix points (see eq. 4). In this case, the performance is optimal from the beginning on (black curve). Interestingly, the system evolves, even in the normal dynamics, to the same value that is reached with the learning phase, supporting the idea that the optimal performance is an emergent behaviour of the system.

### 5.5 Impact of Network Density and Search Type

In the model description, we have described two types of search. Figure 7 – the performance of the system plotted against increasing values of density

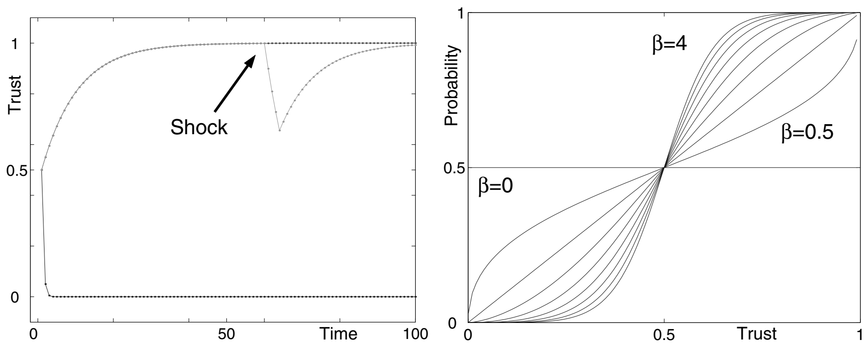


**Fig. 5.** Trust and Decision-Making Dynamics. The left illustrates the slow-positive fast-negative dynamics of trust and the right the impact of the choice of the exploration parameter $\beta$ on the decision making [1]

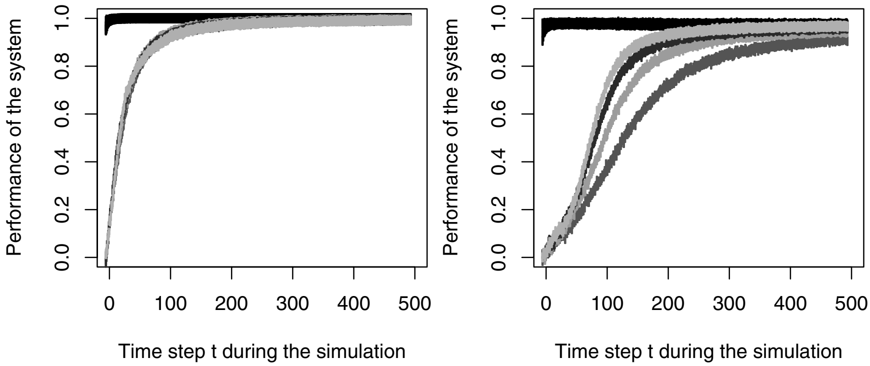


**Fig. 6.** Performance $\Phi$ vs. Time for $N_c = 10$ (**left**) and $N_c = 50$ (**right**). Over time, performance approaches the optimum – with learning (black line), this process is accelerated. Different colours represent different values of $\gamma$ [1]

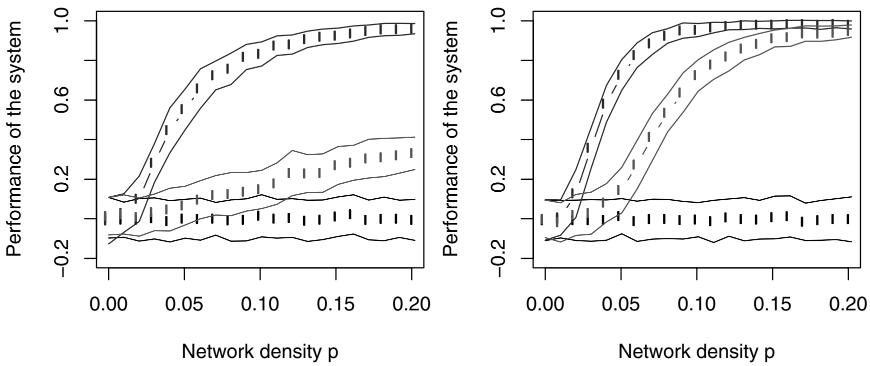


**Fig. 7.** Performance $\Phi$ vs. density for different $N_c$. Incomplete search (**left**) and complete search (**right**). For sparse knowledge, the complete search performs much better than the incomplete search [1]

in the network – shows that the search type becomes important when the knowledge is sparse. We notice a sigmoid shape which would become steeper for systems with larger numbers of agents. We consider different $N_c$, corresponding to levels of sparseness of knowledge (in blue and red, 10 and 50 categories, respectively, $N_p = 2$). With the incomplete search algorithm, the performance deteriorates. With the complete search algorithm, the system reaches the optimal performance even in the case of maximally sparse knowledge (50 categories means that there is only 1 expert from each profile in each category). In both (left) and (right) the black curves correspond to the frequency-based recommendation system used as benchmark. In fact, without trust, the performance is 0 on average, because random choices lead to an equal distribution of “good” and “bad” objects (with respect to profiles).

### 5.6 Preference Heterogeneity and Knowledge Sparseness

We now illustrate the role of preference heterogeneity. We consider first the case in which there are two possible, opposite, profiles in the population, say $p_1$ and $p_2$. We define the fraction of agents characterised by the first profile as $n_1$. In Figure 8 (left), we plot the performance of the system with and without trust (yellow and black, respectively) against increasing values of $n_1$. When $n_1 = 0.5$ there is an equal frequency of both profiles, while when $n_1 = 1$ all agents have the first profile. For the system without trust, the performance increases for increasing $n_1$. In fact, despite that choices are random, agents receive recommendations which are more and more likely to match the preferences of the majority. On the other hand, the minority of agents with the profile $p_2$ are more and more likely to choose wrong recommendations, but their contribution to the performance of the system decreases. The simulation results are in good agreement with the predictions obtained in an analytical approximation (red and blue), see [1].

For the system with trust the performance is almost unchanged by the frequency. This very strong result has the following explanation: The social network is a random graph in which agents have randomly assigned profiles. Agents assigned to $p_2$ decrease in number, but, as long as the minority, as a whole, remains connected (there is a path connecting any two such agents) they are able to filter the correct recommendations. At some point the further assignment of an agent to $p_1$ causes the minority to become disconnected and to make worse choices. In the simulations, this happens when $n_1 = 0.9$ and $n_2 = 0.1$. Another way of investigating the role of heterogeneity of preferences is to consider an increasing number of profiles in the population, each with the same frequency. In the extreme case in which, for each category there is only

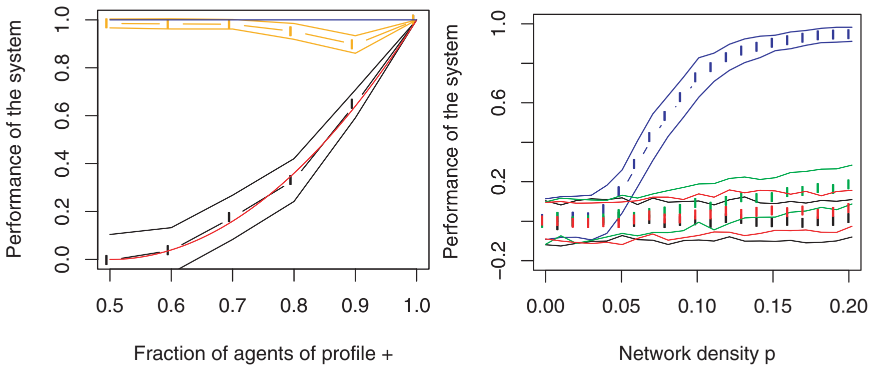


**Fig. 8.** Effect of heterogeneity on performance. Performance as a function of the heterogeneity of preferences (**left**) and with different $N_p$ (**right**). The trust-based approach performs well also in very homogeneous systems; in the extreme case of very heterogeneous systems, performance drops [1]

one expert with any given preference profile, the performance, at constant values of network density, drops dramatically, as shown in Figure 8 (right).

## 6 Application Scenario

We consider a portal on which users can register with their *name* and a *brief profile* containing personal or contact information. Similar to many other social networking services, each user maintains a list of other users which he knows or is a friend of – a list commonly known as the "buddy list" of a user. The system provides a *search facility* in which a user can search for people on the platform by their name, address, or other details, as to make it straightforward for people to find other people they know in the real world. The set of users in the system and the connections between them constitutes a *social network.*

Furthermore, the system maintains a *list of objects*, an object being a unique representation for a user, product, buyer, seller, etc. Each object has a *name* but also several *keywords* and a brief *description* as to enable users to search for objects not only by their name. Each user now maintains a list of objects that it has an opinion on and associates a rating with each of these objects. A rating consists of

- A value of *'like'*, *'neutral'*, or *'dislike'*, corresponding to numeric values in the set $R = \{1, 0, -1\}$.
- Optionally, a *brief textual description* with an explanation of the rating in human-understandable format.

This scheme (please note that it is similar to the one used by `ebay.com`) has several benefits, the main ones being the following:

- It is *simple.* Complicated schemes – for examples, ones requiring users to make more fine-grained ratings – suffer from the fact that no two users will interpret the metrics used in exactly the same manner, thus leading to inaccuracies being amplified by the finer granularity of ratings. Additionally, such schemes tend to be too confusing to use.
- It is *two-fold* in the sense that the numeric value can be processed automatically by algorithmic means but the textual string can still be used by humans in case that they would like to obtain more information on a particular rating.

Consider a university setting and let the users of the system be students, researchers, and professors. The social network is built by acquaintance and thus spans among the people in different groups of a university, but also across groups and universities between people that know each other through collaboration, projects, or conferences. Furthermore, let the objects in the system be publications. Each publication has a unique identifier, information about the title, authors, and similar information, as well as a set of keywords and possibly an abstract. Each user maintains a list of publications he knows – a subset of all publications known to the system – and with each of these, associates a rating and possibly a brief textual description with more information.

The purpose of the recommendation system is to provide users with a unique gateway to more information on the objects listed in the system. Users can search for objects based on the name, description, or keywords. They then see a ranking of objects matching their search and upon selecting a particular object, they are displayed

- information on that object,
- an *aggregate rating* derived from the ratings of users in their social network and weighted by the trust relationships to these users
  and possibly
- a *representative subset of the ratings* (numeric values as well as textual descriptions) used in construction of the aggregate rating.

Based on the recommendation provided by the system, users can then decide to use a particular object. When they do so, they experience this object and thus are able to provide a rating themselves. The system detects and records such ratings and uses them as *feedback* on the trustworthiness of ratings by other users. Over time, the system learns which users provide particularly useful/useless recommendations for which other users, and uses this knowledge to adjust the computation of aggregate ratings for individual users. Along these lines, returning to the example scenario in the university setting, the system allows users to

- Search for publications based on title, authors, keywords, and so on.
- Obtain a recommendation for any such publication; the recommendation is based on the ratings of other users in a user's social network and the trust relationships existing to these other users.
- Obtain a ranking of publications for a particular set of tags, e.g. publications in a particular field or by a particular author.

The benefit of the system is that users are able to select, from a possibly huge set of publications, those that may be relevant for them based on what the people they trust find relevant. Thus, the system provides a filtering technique for people to cope with information overload.

Any such recommendation system can implemented in an according way that it can be accessed through a web interface but also through a web service which seamlessly and transparently makes it available to all sorts of mobile devices such as notebook computers, handheld devices, or mobile phones. This might be more suitable for scenarios different from the example in a university setting, such as recommendation systems for restaurants and bars, products in supermarkets, and so on – returning to the example of Swiss wine and cheese, imagine the scenario of a person querying for recommendations in a supermarket and receiving responses with ratings through a PDA in a matter of seconds, while standing between two aisles in the supermarket.

# 7 Extensions to the Model

At this point, let us return to the model itself. So far, we have made the assumptions that

- agents are self-interested in the sense of bounded rationality, but do not act randomly, selfishly, or maliciously and that
- the social network of agents is fixed and does not change over time – no agents join or leave the networks and no links are rewired, added, or dropped.

In reality, both of these assumptions need to be relaxed, so in further work, we plan to investigate the behaviour of the system in an evolving network as well as its robustness in the presence of agents which act randomly, selfishly, and maliciously.

## 7.1 Evolving Social Network

Considering a static social network between agents does not appropriately depict reality; usually, *social networks evolve over time* with links being created and deleted at each time step. People tend to establish contacts to new people and lose contact to old acquaintances. Both of these actions lead to an evolution of the underlying social network.

Thus, to stay close to reality, we have to analyse the model having an underlying dynamic social network with the possibility of the

- *Creation of links* between agents which have mutually benefited from each other's recommendations for a particular number of times.
- *Deletion of links* unilaterally or bilaterally between agents that believe the other agent to give useless recommendations.

It is conceivable that the evolution of the social network has a crucial impact on the performance of the system: over time, agents learn which other agents are trustworthy as well as which are not and adjust their links accordingly. A priori, it is not clear whether this leads to better performance (possibly because agents have similar agents as their immediate neighbours that they can rely on) or worse performance (possibly because agents focus too much on their immediate neighbours to see that there are more opinions than these). It might also be interesting to analyse to what extent the global and local properties of the underlying social network change.

It is reasonable to assume that a person is more likely to keep a link towards a neighbour the more he/she trusts the neighbour and vice versa.

We do not model the decision-making process explicitly but we capture this tendency with a stochastic rule as follows:

$$P(\text{rewire}) = 1 - T_{a_i,a_j}, \;\; P(\text{keep}) = T_{a_i,a_j} \tag{8}$$

i.e. $P(\text{rewire}) + P(\text{keep}) = 1$. Equation 8 implies that the probability to randomly rewire the link from agent $a_i$ to $a_j$ is high if the trust from $a_i$ to $a_j$ is low. Thus, trustworthy links are kept while untrustworthy links are replaced.

Figure 9 shows how snapshots of the evolution of a sample network of agents at different stages for different values of $\beta$ (more and less explorative agents) look like when applying this mechanism. This illustrates the dilemma between exploration and exploitation faced by the agents. For $\beta = 0$, agents choose randomly, thus performing worse, but they explore many the other agents repetitively and their trust relationships converge to the steady state of the trust dynamics. Then, over time, links with low trust are rewired and links with high trust are kept. This leads to the emergence of two disconnected clusters. Eventually, subsequent to the formation of clusters, such agents will

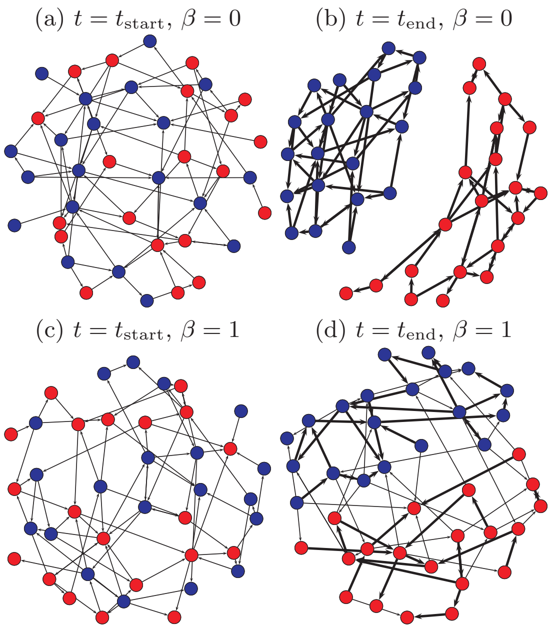


**Fig. 9.** Snapshots of the evolution of a network of 40 agents in 2 profiles and 80 links at time $t = t_{\text{start}}$ and $t = t_{\text{end}}$ for $\beta = 0$ and $\beta = 1$, respectively. When $\beta = 0$ (very explorative agents, see eq. 3), disconnected clusters of agents with the same profile form, when $\beta = 1$ (less explorative agents, see eq. 3), interconnected clusters of agents with the same profiles form. For $\beta > 0$, agents develop stronger ties to agents of the same profile than to agents of different profiles [1]

perform well, as any recommendation will come from an agent of the same profile. For $\beta = 1$, agents choose according to the strength of trust relationships, thus performing better, and they are able to exploit their knowledge. However, they exploit stronger links while not even exploring weaker ones. This results in clustering, but with interconnections between clusters. As networks in reality are evolving, it is important to study the impact of such behaviour on the system in more detail.

### 7.2 Robustness against Attacks

The model also allows us to focus on the robustness of the recommendation system against attacks. This is a very important aspect because of the fact that in real-life systems there will be users that try to cheat the system as soon as money is involved – which would be the case even in the illustrative example of Swiss wine and cheese. The financial incentive for some of the agents in the system may have a level high enough to, for example, lead to the following: wine and cheese manufacturers may be tempted to improve recommendations for their products so as to increase their revenues, upset customers may try to defame products that they made bad experiences with as an act of retaliation, and so on.

To illustrate and further stress this point, consider a similar example from the field of search engines: Google, currently the most widely used search engine builds its search engine rankings according to the page rank algorithm. The basic idea is that the more links point to a page, the higher up in the search ranking this page will be placed. Of course, as Google has a vast market share in the search engine domain, it is of utmost importance for the manufacturers of a certain product or the providers of a certain service that they rank among the top 5 of the search engine results for certain keywords. There is a strong incentive for manipulation of the search engine results by means of increasing the number of links to particular pages in the context of certain keywords. This can be done, for example, by setting up large numbers of artificial web pages with hardly any content except a number of keywords which all cross-link to a desired web page and thus increase the number of links to this page with respect to the keywords. This has become known as "Google bombing" and is an ongoing issue that all search engines have to deal with.

Thus, in the construction of a real-life recommendation system, cheaters and attackers have to be considered. For example, it would be possible to consider three different additional types of agents:

- *Random agents* are agents that, instead of giving correct recommendations, give a random recommendation. This is not necessarily due to selfish or malicious intentions, but may just as well result from a pure lack of knowledge. In a sense, having such agents in the system mimics the effect of noise on communication channels.

- *Selfish agents* are agents that do not return recommendations except in the case that they have already received responses through the agent that initiated the query. Obviously, if all the agents in a system are selfish, the system is in a deadlock state where no one gives anyone else recommendations.
- *Malicious agents* are agents that intentionally give recommendations that do not correspond to their own beliefs – i.e., they recommend what they would not use themselves, and vice versa. An ideal recommendation system should be able to cope with such agents.

In each of the cases, we are interested in the performance of the recommendation system with respect to differing fractions of such random, selfish, and malicious agents in the system: Does the presence of random/selfish/malicious agents impact the performance of the recommendation system? Is there a critical value of the fraction of random/selfish/malicious agents for which the recommendation system becomes unusable/usable?

It may also be interesting to look at more sophisticated agents, e.g. ones that alternate between these types of behaviour, or agents which form networks with other agents to influence the system in a particular way. Understanding the aspect of the robustness against attacks is crucial for real-life systems.

## 8 Conclusions

In this chapter, we have presented trust-based networks as an application of complex systems theory to cope with information overload on the Internet. By combining recommendation systems, trust, and social networks, it is possible to build a system in which agents use their trust relationships to filter the information that they have to process, and their social network to reach knowledge that is located far from them. The emergent property of the system is that it self-organises in a state with performance near to the optimum without explicit coordination of the agents. In this chapter, we have given one example of a real-world application, but we believe that the system is applicable to a vast variety of domains ranging from low-involvement products such as books or groceries to high-involvement services such as insurance or health-care.

## Acknowledgements

We thank Markus M. Geipel, Patrick Groeber and João F. M. Rodrigues for their valuable comments. The network graphs were layouted with an elastic-band-layouter developed by Markus M. Geipel. For details, please refer to `www.sg.ethz.ch/research/graphlayout`.

## References


1. Walter, F.E., Battiston, S. and Schweitzer, F.: A Model of a Trust-based Recommendation System on a Social Network. Journal of Autonomous Agents and Multi-Agent Systems (forthcoming, 2007). Available on Arxiv.org, http://arxiv.org/abs/nlin.AO/0611054 (2006)
2. Wellmann, B.: Computer networks as social networks. Science **293** (2001) 2031–2034
3. Huberman, B.A., and Adamic, L.A.: Growth dynamics of the World-Wide Web. Nature **401** (1999) 131
4. Marsh, S.: Formalising Trust as a Computational Concept. Phd. Thesis - University of Stirling (1994)
5. Mui, L., Mohtashemi, M. and Halberstadt, A.: A Computational Model of Trust and Reputation for E-Businesses. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences IEEE Press (2002)
6. Castelfranchi, C. and Falcone, R.: Social trust: a cognitive approach. In: Castelfranchi, C., and Tan, H.-J., eds.: Trust and deception in virtual societies. Kluwer Academic Publishers (2001) 55–90
7. Gray, E., Seigneur, J.-M., Chen, Y. and Jensen, C.D.: Trust Propagation in Small Worlds. In: Nixon, P. and Terzis, S., eds.: Proceedings of the First International Conference on Trust Management, Lecture Notes in Computer Science Springer **2692** (2003) 239–254
8. Guha, R., Kumar, R., Raghavan, p. and Tomkins, A.: Propagation of Trust and Distrust. In WWW '04: Proceedings of the 13th International Conference on the World Wide Web ACM Press (2004) 403–412
9. Sabater, J. and Sierra, C.: Review on Computational Trust and Reputation Models. Artificial Intelligence Review **24** (2005) 33–60
10. Brin, S. and Page, L.: The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems **30** (1998) 107–117
11. Montaner, M., López B., De La Rosa, J.-L. A Taxonomy of Recommender Agents on the Internet. Artificial Intelligence Review **19** (2003) 285–330
12. Massa, P. and Bhattacharjee, B.: Using Trust in Recommender Systems: An Experimental Analysis In: Jensen, C.D., Poslad, S. and Dimitrakos, T., eds.: Proceedings of the Second International Conference on Trust Management (ITRUST 2004), Lecture Notes in Computer Science Springer **2692** (2004) 221–235
13. Massa, P. and Bhattacharjee, B.: Using Trust in Recommender Systems: An Experimental Analysis In: Gini, M., Ishida, T., Castelfranchi, C., and Lewis Johnson, W., eds.: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02) ACM Press (2002) 304–305
14. Erdos, P. and Rényi, A.: On random graphs. Publicationes Mathematicae Debrecen **6** (1959) 290–291
15. Watts, D.J. and Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393** (1998) 440–442
16. Barabasi, A.-L. and Albert, R.: Emergence of Scaling in Random Networks. Science **286** (1999) 509–512
17. Newman, M.: The Structure and Function of Complex Networks. SIAM Review **45** (2003)

18. Kleinberg, J. and Lawrence, S.: The structure of the Web. Science **294** (2001) 1849–1850
19. Amaral, L.A.N., Scala, A., Barthélémy, M. and Stanley, H.E.: Classes of small-world networks. Proceedings of the National Academy of Sciences **97** (2000) 11149–11152
20. Golder, S. and Huberman, B.A.: The Structure of Collaborative Tagging Systems. http://arxiv.org/cs/0508082 (2005)
21. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition **5** (1993) 199–220
22. Battiston, S., Bonabeau, E. and Weisbuch, G.: Decision making dynamics in corporate boards. Physica A **322** (2003) 567
23. Battiston, S., Weisbuch, G. and Bonabeau, E.: Spread of decisions in the corporate board network. Advances in Complex Systems **6** (2003)
24. Battiston, S., Weisbuch, G. and Bonabeau, E.: Statistical properties of board and director networks. European Journal of Physics B **38** (2004)
25. Laureti, P. and Moret, L. and Zhang, Y. -C. and Yu, Y. -K.: Information Filtering via Iterative Refinement. Europhysics Letters **75** (2006) 1006
26. Laureti, P., Slanina, F.,Yu, Y.-K. and Zhang, Y.-C. Buyer Feedback as a Filtering Mechanism for Reputable Sellers. Physica A **316** (2002) 413
27. Dorigo, M., Maniezzo, V. and Colorni, A.: The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics **26** (1996) 29–41
28. Ziegler, C.-N. and Golbeck, J.: Investigating Correlations of Trust and Interest Similarity. Decision Support Systems (2006)
29. Schweitzer, F. and Lao, K. and Family, F.: Active random walkers simulate trunk trail formation by ants. BioSystems **41** (1997) 153–166
30. Weisbuch, G., Kirman, A. and Herreiner, D.: Market Organisation and Trading Relationships. The Economic Journal **110** (1998) 411–436
31. Palau, J., Montaner, M., López, B. and de la Rosa, J.-L: Collaboration Analysis in Recommender Systems Using Social Networks. In: Klusch, M., Ossowski, S. Kashyap, V. and Unland R., eds.: Proceedings of the 8th International Workshop on Cooperative Information Agents (CIA 2004), Lecture Notes in Computer Science Springer **3191** (2004) 137–151
32. Battiston, S., Walter, F.E. and Schweitzer, F.: Impact of Trust on the Performance of a Recommendation System in a Social Network In: Proceedings of the Workshop on Trust at the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06) (2006)
33. Abdul-Rahman, A. and Hailes, S.: Supporting Trust in Virtual Communities. In: Proceedings of the 33th Annual Hawaii International Conference on System Sciences (2000)
34. Bollobas, B.: Random Graphs. Academic Press (1985)
35. Luhmann, N.: Trust and Power John Wiley & Sons (1979)
36. Sztompka, P.: Trust: A Socialogical Theory Cambridge University Press (1999)
37. Gambetta, D.: Trust: Making and Breaking Cooperative Relations Electronic Edition, Oxford University (2000)
38. Sarwar, B., Karypis, G., Konstan, J., and Riedl, J.: Analysis of Recommendation Algorithms for E-Commerce. In: EC '00: Proceedings of the 2nd ACM conference on Electronic Commerce ACM Press (2000) 158–167

39. Golbeck, J.: Generating Predictive Movie Recommendations from Trust in Social Networks. In: Proceedings of the 4th International Conference on Trust Management (2006) 93–104