Trust-aware Bootstrapping of Recommender Systems

Paolo Massa and **Paolo Avesani** [1]

Abstract. Recommender Systems (RS) suggest to users items they might like such as movies or songs. However they are not able to generate recommendations for users who just registered, in fact bootstrapping Recommender Systems for new users is still an open challenge. While traditional RSs exploit only ratings provided by users about items, Trust-aware Recommender Systems let the user express also trust statements, i.e. their subjective opinions about the usefulness of other users. In this paper we analyze the relative benefits of asking new users either few ratings about items or few trust statements about other users for the purpose of bootstrapping a RS ability to generate recommendations. We run experiments on a large real world dataset derived from the online Web community of Epinions.com. The results clearly indicate that while traditional RS algorithms exploiting ratings on items fail for new users, asking few trust statements to a new user is instead a very effective strategy able to quickly let the RS generate many accurate items recommendations.

1 Introduction

Information overload makes Recommender Systems (RS) [3] a tool that cannot be renounced. Nevertheless the bootstrapping of a Recommender System is still an open challenge.

Bootstrapping, known also as cold start problem, is threefold: it can be concerned with a new system, a new item or a new user. The first scenario refers to situations where a Recommender System has been just launched and can't rely on the collaborative contribution of a community of users [2]. The second scenario is represented by the extension of the catalog of contents: usually opinions on recently introduced items, for example new movies, are not available [9]. Third, we have a cold start problem when a new user subscribe to a Recommender System [6]. In the following we will focus our attention on this third kind of bootstrapping challenge.

When a new user joins for the first time a Recommender System the system doesn't know anything about her. A poor or an empty profile prevents the system to deliver personalized recommendations. The main drawback is the latent period required by the system to acquire enough knowledge about the new user. Proactive strategies, based on user preference elicitation, may shorten this period but there is the risk of annoying the user. The bottleneck for a quick bootstrapping is therefore the elicitation of user preferences: it has to be enough rich to enable the Recommender System and at the same time enough quick to not bother the user and to drive her away from the system.

In this paper we hence concentrate on the issue of bootstrapping Recommender Systems based on Collaborative Filtering (CF) for new users. And we propose to tackle this problem by exploiting elicitation of explicit trust between users. As in CF the user provides examples of items she likes, in the same way the user can provide examples of users she trusts when operating in a trust-aware framework [6]. The intuitive strategy is to exploit the notion of trust that allows the users to refers to those "reviewers whose reviews and ratings they have consistently found to be valuable"[2]. According to this strategy the early profile elicitation will be oriented to acquire ratings on other users rather than ratings on items.

The working hypothesis is that inviting users to elicit opinions on users (trust statements) rather than opinions on items allows to shorten the bootstrapping of RSs for cold start users. The benefits can be summarized as follows: (1) the number of trust statements needed from a new user for bootstrapping a Recommender System is much less than the number of rating on items; (2) while exploiting the few early ratings provided by a new user does not enable to generate recommendations, exploiting just few early trust statements allows to significantly increase the number of possible recommendations; (3) the accuracy of generated recommendations increases as well exploiting trust statements rather than ratings on items.

The main contribution of this paper is the empirical proof of our hypotheses on a real world dataset, derived from the large Web community of Epinions (http://epinions.com). The straightforward impact of this work is a new guideline for Recommender Systems design: a new user has to be invited to elicit few other users she trusts rather than to express her opinions on a pool of items.

In the following we briefly summarize the issues that arise when a new user approaches a Recommender System, afterwards we introduce the basic notions of trust network and trust metric. Section 4 illustrates the hypotheses of this work, while Section 5 and Section 6 are devoted to the empirical analysis and the discussion of results respectively.

2 Motivation

Collaborative Filtering (CF) [3] is the most used technique for Recommender Systems. CF relies on the opinions provided by the users in the form of ratings to items, such as movies, songs or restaurants. A CF algorithm predicts the rating a certain user might give to a certain item she has not yet rated and experienced; the RS can then, for example, suggest to that user the items not yet rated that received the highest predicted rating. CF does not consider the content of the items, such as the genre of a movie, but only the ratings provided by the community of users and hence it can work unchanged on every domain. The functioning of a Collaborative Filtering Recommender System can be divided in two independent steps: (1) neighbours formation and (2) ratings prediction. In order to create items recommendations for the active user, first the CF algorithm tries to find some like-minded users that have tastes similar to the active user (*step 1*). Like-minded users are called neighbours and CF computes a similarity coefficient for each of them. *Step 2* consists into predicting the

[1] ITC-IRST, Trento, Italy

[2] This formulation of trust is that proposed to Epinions.com users.

rating the active user would give to a certain item as the weighted average of the ratings given by her neighbours to that item, where the weights are the similarity coefficients of the neighbours. The rationale is that the ratings of users that are more similar to the active user are considered more.

The typical instantiation of step 1 is based on the computation of the Pearson correlation coefficient (Formula 1), that has shown to provide the best results [3]. It computes $w_{a,u}$, the value of similarity between the active user a and another user u, as a comparison of the ratings they provided. $r_{a,i}$ is the rating given by user a to item i and $\overline{r}_a$ is the average of the ratings given by user a.

$$w_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i} - \overline{r}_a)(r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i=1}^{m}(r_{a,i} - \overline{r}_a)^2 \sum_{i=1}^{m}(r_{u,i} - \overline{r}_u)^2}} \qquad (1)$$

Note that m is the number of items rated by both user a and u and in fact, in order to compare two users, there must be some overlapping in the items they have rated. Lack of overlapping means the similarity weight $w_{a,u}$ cannot be computed. Moreover, if 2 users only have one item rated by both, then the coefficient is not meaningful, being either 1 or -1 based on the differences of the rating with respect to the average rating. Hence, for a user, it is possible to compute the correlation coefficient only in users who share at least 2 co-rated items and we will see in the following how these are usually a small portion. Additionally, a similarity weight computed based on few item ratings is a noisy and unreliable value.

It is crucial to note that a failure in step 1 produces a failure in step 2. In fact if the number of identified neighbours for the active user is null or very small, it is unlikely that they have rated the item whose rating the RS tries to predict and hence a recommendation is not possible. The computability of similarity weights is a paramount problem for new users: since they have rated 0 items, it is not possible to find neighbours and hence it is not possible to predict their ratings and generate personalized recommendations for them. This is an intrinsic weaknesses of the Collaborative Filtering model: new users suffer from the cold start problem. Our goal is to propose a way for bootstrapping RSs for newly registered users and for exploiting as soon and as much as possible the early preferences elicited by the users.

The main idea of Trust-aware Recommender Systems [6] is to change what a RS asks to its users: from rating items to rating other users. Rating other users means expressing how much the active user trusts them for their ability to provide useful ratings to items. We call this expression a trust statement and we will precise this concept in the next section along with an analysis of the differences between rating items and rating other users. Let us briefly note that letting users express trust in other users is a feature that is becoming more and more utilized in current Web communities [5]. For example on Epinions (http://epinions.com), users can assign ratings to items but they can also express which users they trust ("reviewers whose reviews and ratings that user has consistently found to be valuable") and which users they distrust.

In this paper we explore whether the user activity of rating items can be replaced by and/or integrated with the user activity of rating other users, i.e. of expressing trust statements. The ultimate goal is to reduce the elicitation effort for the users and to allow Recommender Systems to create recommendations for the users as soon as possible. This is especially relevant for newly registered users: unless they receive good and tailored items recommendations since the very beginning, they have an incentive for leaving the system and never contribute again.

3 Trust Metrics in Recommender Systems

Trust is a concept that is starting to receive increasing attention by the research community and be used in many current online systems [5]. For the purpose of this paper we define trust as "the explicit opinion expressed by a user about another user regarding the perceived quality of a certain characteristic of this user". In Recommender Systems, the characteristic that is under evaluation is the ability to provide useful ratings, so that a source user should trust a target user if she believes that the target user's ratings are useful to her. When referring to the information provided by an user, we also call it "trust statement". Since the users of a system express trust statements about other users, it is possible to aggregate all the trust statements for building the overall trust network [7]. Note that the trust network is weighted (if the users can express different numeric scores for their trust in other users) and directed (since trust statements are not necessarily symmetric).

In the context of Recommender Systems, the traditional information expressed by users is ratings given to items. Trust statements are instead ratings given to users and the goal of this paper is to analyze differences between them and if trust statements are more effective for bootstrapping a RS for a new user. The most relevant difference between ratings to items and ratings to users is that the second ones can be propagated. In fact, assuming user a does not know user b (i.e. she has not expressed a trust statement in her), it is possible to predict the degree of trust a might pose in b exploiting trust propagation over the trust network. Trust metrics [11] are computational algorithms with this precise goal. The basic assumption of trust metrics is that if user a trusts user b at a certain level and user b trusts user c at a certain level, something can be predicted about how much a should trust c. This reflects the intuition that friends of friends are more likely to become friends than random strangers and that it is common to rely on opinions of known people when forming a first opinion about unknown people.

While the topic of trust metrics is very recent, it is receiving an increasing attention. Let us briefly introduce how PageRank [8], one of the algorithms powering the search engine Google (http://google.com) can be considered a trust metric, since it performs trust propagation over the link network in order to compute which Web pages are more authoritative. Other trust metrics have been recently proposed in the context of Semantic Web [1], Recommender Systems [10, 6] and Peer-to-Peer networks [4]. Trust Metrics can be classified into Local and Global [11, 7]. Global trust metrics produce a value of reputation for a precise user that is the same from the point of view of every other user while local trust metrics provide personalized views. However it is out of the scope of this paper to provide a survey of the proposed trust metrics and the rest of the Section is devoted to briefly explain the trust metrics we have used in our experiments, MoleTrust, described in [7]. It is a local trust metric and hence it must be run once from the point of view of every user and not just once for all the community as with global trust metrics. MoleTrust predicts the trust value a *source user* should place into a *target user* by walking the trust network starting from the source user and by propagating trust along trust statements (the directed edges of the trust network). Intuitively the trust score of an unknown user depends on the trust statements she received and the trust scores of the users who issued them.

It can be divided in two stages. At the first stage the task is to remove cycles in the trust network and hence to transform it into a directed acyclic graph (DAG). The problem created by cycles is that, during the graph walk, they require visiting a node many times adjusting progressively the temporary trust value until this value converges. In order to have a time-efficient algorithm, it is preferable to visit every single node no more than once and, in doing this, to compute her definitive trust value. In this way, the running time is linear with the number of nodes. Moreover trust is propagated only to users at a certain distance so that, by controlling this trust propagation horizon parameter, it is possible to reduce even more the computational time. After the first stage, the trust network is a DAG with trust flowing away from source user and never flowing back.

The second stage is responsible for predicting the trust scores, representing how much source user should trust every single other reached user. For predicting the trust score of a user, MoleTrust analyzes all the incoming trust edges (representing the trust statements remaining from step 1) and accepts only the ones coming from users with a predicted trust score greater or equal than a certain threshold (in our experiments, set to 0.6). The predicted trust score of a user is the average of all the accepted incoming trust statement values, weighted by the trust score of the user who has issued the trust statement. A more detailed explanation of MoleTrust can be found in [7].

4 Working Hypotheses

Trust-aware Recommender Systems [6] extend the type of information collected from users enabling the elicitation of opinions both on items and other users. Therefore the trust statements don't replace the ratings but simply enrich the notion of user profile. In our work we are not interested to prove whether elicitation of user trust should be preferred to opinions on items because ratings are the kernel of Recommender Systems. The focus of our research is the investigation of informative power of different kind of opinions at the early stage of user registration. The main question is to understand whether for a new user it is more effective to express opinions on an item she likes or to elicit a user she trusts.

The investigation of this question is designed taking into account the general framework of recommender systems illustrated in Section 2. The basic idea is to arrange an alternative implementation of neighbours computation (see step 1 of architecture). The assessment of user similarity based on the similarity of ratings is replaced with the use of trust information. The alternative way to measure user relevance weights is therefore derived from the explicit trust statements and the estimated trust values computed with propagation metrics.

In the following we analyze the relative benefits of the elicitation of cumulative ratings rather than trust statements at the early stage of interaction for novel users.

Since new users start with 0 ratings on items and 0 trust statements on other users, it is of paramount importance for the RS to come to know some information about the user as soon as possible. We assume that a user would like to receive a useful service without having to spend too much time and effort in providing information to the Recommender System. For this reason it is very important to reduce as much as possible the initial bootstrapping period in which the RS is not able to provide recommendations but simply asks information from the user. Here we compare two opposite strategies of asking few trust statements and of asking few ratings on items and their relative benefits in letting the RS provide recommendations to the new user.

Our first working hypothesis is that with few trust statements, a RS is able to find a large number of neighbours, i.e. performs well in step 1 of the RS framework we previously described. The reason behind this hypothesis is trust propagation over the trust network that can be performed by a trust metric. In fact, propagating trust starting just with few trust statements emanating from the new user should allow to reach most of the other users and hence to consider them as possible neighbours. On the other hand few ratings on items expressed by the new user in general don't allow to compare the new user with all the other users. The reasons are data sparsity and the fact that overlapping between rated items is required for Pearson correlation coefficient to be computable. Moreover, even when there is such an overlapping, a similarity coefficient based just on very few items rated by the two users tends to be noisy and unreliable. As a consequence, step 1 fails and the number of identified neighbours is null or tiny at best.

Our second hypothesis is that the larger number of identified neighbours translates into a larger number of items for which recommendation predictions are possible, i.e. the coverage of the algorithm is greater with Trust-aware Recommender Systems.

While the number of neighbours might be greater, it might be the case that the identified neighbours are not good quality neighbours and that the recommendations created with a weighted sum of their ratings are not accurate. Our third hypothesis is that recommendations accuracy for new users when a recommendation is possible is comparable for the two different methods.

The overall assumption underlying our experiments is that the best way to bootstrap a Recommender System for a new user is by exploiting trust, i.e. the first information asked to a new user should be to identify few other users she trusts and not to rate few items.

5 Experimental Settings

We tested the previously introduced hypotheses against a real world dataset derived from the large Web community of Epinions. Epinions is a consumers opinion site in which users can review items (such as cars, books, movies, software, etc) and also assign them numeric ratings in the range 1 (min) to 5 (max). Users can also express their *Web of Trust*, i.e. "reviewers whose reviews and ratings they have consistently found to be valuable" and their *Block list*, i.e. "a list of authors whose reviews they find consistently offensive, inaccurate, or not valuable". We crawled these data directly from the Epinions Web site. Our dataset consists of approximatively $50,000$ users who rated a total of almost $140,000$ different items at least once. The total number of reviews is around $660,000$. The total number of issued trust statements is about $490,000$. Details about the distributions of ratings and trust statements can be found in [6]. Note that the Block list is not shown on the site and kept private and hence it is not available in our dataset. Table 1 presents the percentage of cold start users, users who provided few ratings and few trust statements. Note how the largest portion of Epinions users are cold start users, for example, more than half of the users (53%) provided less than 5 ratings. It is important to underline that these are real world distributions representing a large community of real users and their elicitation patterns.

	0	1	2	3	4	5	6	7
#ratings	18.52	15.70	7.99	5.92	4.70	3.89	3.33	2.93%
#trust	31.10	19.14	9.46	6.10	4.38	3.43	2.64	1.98%

Table 1. Percentage of Epinions users who expressed x ratings and x trust statements.

In order to test our hypotheses we run two different algorithms and we compared them. The first algorithm is a standard Collaborating Filtering one [3] taking as input the ratings provided by users. In step

1 it computes the similarity weights between the active user and all the other users in order to find neighbours using Pearson correlation coefficient as defined in Formula 1. Then in step 2 it predicts the rating that active user would give to a certain item as a weighted sum of the ratings given by her neighbours to that item, where the weights are the similarity coefficients computed in step 1.

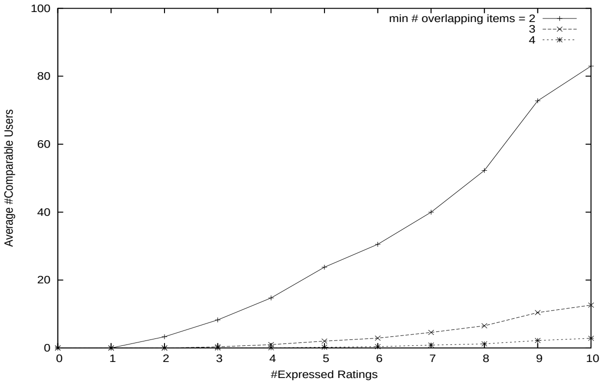

Figure 1. Average number of comparable users computing Pearson correlation coefficient with different minimum number of overlapping items. Users are aggregated based on the number of ratings they gave (x axis).

The second algorithm is a trust-aware one [6] taking as input the trust statements provided by users. Step 1 finds neighbours and their weights by using MoleTrust trust metric that propagates trust over the trust network. Step 2 is precisely the same as a standard CF algorithm. In essence, the only difference is in how the two algorithms find neighbours and which information they take as input. We compare the two algorithms when they utilize a similar amount of information bits, for example the performances of a CF algorithm on users who provided 3 ratings are going to be compared with the performances of a trust-aware algorithm on users who provided 3 trust statements.

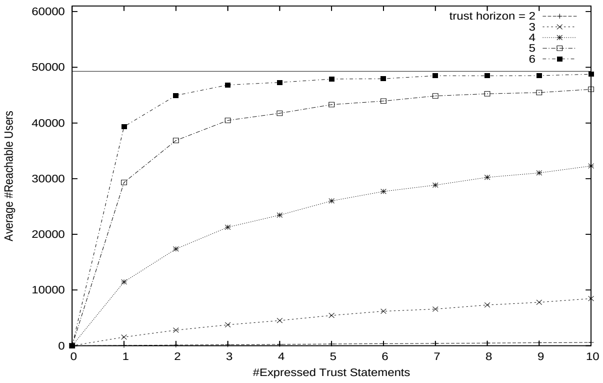

Figure 2. Average number of reachable users propagating trust up to different trust propagation horizons. Users are aggregated based on the number of trust statements they expressed (x axis). The horizontal line represents the maximum number of reachable users (49289).

In order to test the first hypothesis, we analyze the number of users for which a weight can be computed using the two algorithms. For the standard CF algorithm, Figure 1 reports the average number of comparable users, with different required quantities of overlapping items. Note that the y axis is much smaller than the ideal maximum (the number of users minus 1) that is 49289. For users with less than 5 ratings, even accepting similarity weights computed only on 2 overlapping items, the number of comparable users is less than 20! For the trust-aware RS algorithm, we analyze the benefit of propagating trust up to different trust propagation horizons (see Figure 2). Of course, with larger horizons more users are reachable and can be considered as neighbours but their trust score predictions become less and less reliable. Note however that just by propagating trust up to distance 3 or 4 it is possible to reach a very large portion of users also for cold start users. This is particularly important when compared to the tiny portions of users comparable with the standard Pearson correlation coefficient (Figure 1). Let us underline once more the striking difference in the y axis of Figures 1 and 2.

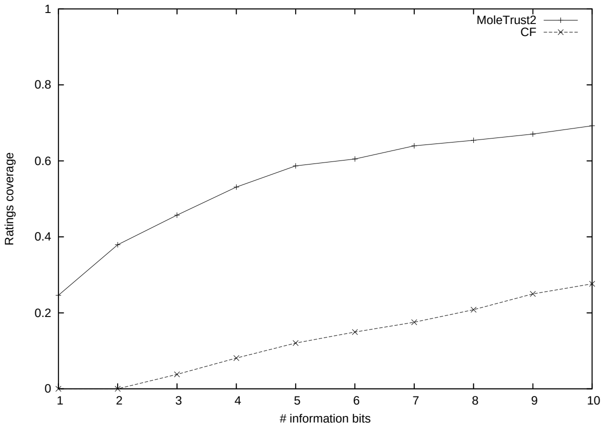

Figure 3. Ratings coverage for MoleTrust2 and CF.

In order to test second and third hypotheses, we analyze both the accuracy and the coverage of the overall algorithms, i.e. the final output of step 2 that is predicted ratings. We use leave-one-out methodology that consists into deleting one rating, trying to predict it with an algorithm, and then comparing the real and the predicted ratings. Coverage refers to the portion of deleted ratings for which a prediction is possible. Accuracy refers to the difference between the real and predicted rating when a prediction is possible, in particular we computed Mean Absolute Error (MAE) [3].

Figure 3 shows the ratings coverage of the different algorithms while Figure 4 reports the MAE representing their accuracy. For the trust-aware Recommender System algorithm, we present here the results obtained propagating trust up to distance 2, and hence the algorithm is called “MoleTrust2”. Note that for users with 2 ratings, CF is not able to produce any recommendations since, after leave-one-out removal, they are actually users with 1 rating and hence it is not possible to compute their similarity with any other user.

6 Discussion of results

Figure 1 and Figure 2 show the relative benefits of exploiting Pearson correlation coefficient on ratings and of exploiting the MoleTrust trust metric on trust statements for the purpose of finding neighbours (step 1 of the algorithms). It shows that, taken a new user and using the similarity coefficient, the number of other users that are comparable is extremely tiny. Note how the ideal value would be the total

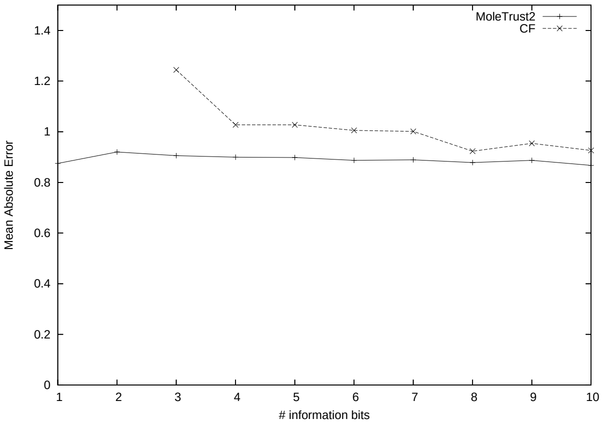

Figure 4. MAE of ratings predictions for MoleTrust2 and CF.

number of users (almost $50,000$) while the y axis of Figure 1 is 100. On the other hand, exploiting trust it is possible to reach a large portion of the users; propagating up to distance 5 for instance allows to reach almost all the $50,000$ users also for cold start users. This is a significant improvement with respect to the use of a similarity coefficient on items ratings and this confirms our first hypothesis. These two figures really gives an idea of the potential of the different input information used in step 1 in order to form the set of neighbours. In fact, exploiting trust as input information is more effective since trust can be propagated over the trust network using a trust metric. On the other hand, the computation of similarity coefficients between ratings requires overlapping of rated items but this is very unlikely to happen because of data sparsity and this is especially an issue for new users.

With respect to the second hypothesis, Figure 3 gives a powerful visual insight about the relative performances of a CF algorithm and a trust-aware one (MoleTrust2). Let us remember that cold start users are really the majority of the users in our realistic Epinions dataset, for example 53% of the users provided less than 5 ratings. So the difference in performances really affects a significant portion of the users. As an example, for users who provided 4 ratings, CF is on average able to predict less than 10% of their ratings. Instead, for users who provided 4 trust statements, trust-aware is able to predict around 50% of their ratings! Note that, because of leave-one-out, for users with n ratings, only $n-1$ ratings are in reality used. However, even shifting the line of CF coverage on the left of one position, the difference in coverage remains huge.

With respect to the third hypothesis, Figure 4 clearly shows how the error produced by MoleTrust2 is smaller than the one produced by CF. Even if the difference is not too large, this is an important point as well.

It is worth underlying that the evidence presented here is based on experiments run on a large, real world dataset. This evidence shows that bootstrapping a RS for a new user is possible with just very few trust statements, even just 1 or 2. From this evidence it is possible to derive a suggestion for designers of Recommender Systems: new users should be asked to find soon at least one other trustworthy user already in the system. She can be for example the user who invited the new user in the system or, as another example, a user very active in the community and likely to be appreciated that is conveniently shown in the RS homepage. Note however that we don't propose to totally replace ratings on items nor we state that ratings on items are not useful and should not be acquired and asked to users. Actually they are the real basic data used in step 2 by both algorithms since the predicted ratings are computed as a weighted average of the ratings provided by users. In this paper, we simply made the case for a bootstrapping strategy for new users powered by trust: the initial short information gathering window for a new user should be guided towards acquiring few trusted users instead of few ratings on items in order for the RS to be able to generate many accurate recommendations soon so that the user is satisfied and keeps using the system, possibly by providing also ratings on items.

7 Conclusions

In this paper we have compared two possible ways of bootstrapping a Recommender System for a new user. The first way is the traditional approach of asking to a new user to rate few items so that the system can start finding other users with similar tastes and generate recommendation for the new user. The alternative way is related to the elicitation of trust: the new user is asked to explicitly indicate few other users she trusts. We have evaluated the two strategies on data derived from the large and real world Web community of Epinions. Our experiments demonstrates that asking ratings to a new user is unlikely to rapidly let the RS to generate personalized recommendation due to data sparsity and the need of overlapping of rated items with possible neighbours. On the other hand a RS able to get just few trust statements from a new user is able to produce a large number of accurate personalized recommendations, because it is able to exploit trust propagation over the trust network by means of a trust metric. A suggestion to Recommender System designers can be derived from the presented evidence: the RS should not ask to a new user to rate some items but instead just to indicate few trustworthy users already in the system.

REFERENCES

[1] J. Golbeck, J. Hendler, and B. Parsia, 'Trust networks on the Semantic Web', in *Proceedings of Cooperative Intelligent Agents*, (2003).

[2] H. Guo, 'Soap: Live recommendations through social agents', in *Fifth DELOS Workshop on Filtering and Collaborative Filtering*, (1997).

[3] J. Herlocker, J. Konstan J., A. Borchers, and J. Riedl, 'An Algorithmic Framework for Performing Collaborative Filtering.', in *Proceedings of the 22nd SIGIR Conference*, (1999).

[4] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, 'The EigenTrust Algorithm for Reputation Management in P2P Networks', in *Proceedings of WWW2003*. ACM, (2003).

[5] P. Massa, 'A survey of trust use and modeling in current real systems', (2006). Under publication in "Trust in E-Services: Technologies, Practices and Challenges", Idea Group, Inc.

[6] P. Massa and P. Avesani, 'Trust-aware collaborative filtering for recommender systems', in *Proc. of Federated Int. Conference On The Move to Meaningful Internet: CoopIS, DOA, ODBASE*, (2004).

[7] P. Massa and P. Avesani, 'Controversial users demand local trust metrics: an experimental study on epinions.com community', in *Proc. of 25th AAAI Conference*, (2005).

[8] L. Page, S. Brin, R. Motwani, and T. Winograd, 'The pagerank citation ranking: Bringing order to the web', Technical report, Stanford, USA, (1998).

[9] A. Schein, A. Popescul, L. Ungar, and D. Pennock, 'Methods and metrics for cold-start recommendations', in *Proceedings of the 25th SIGIR Conference*, (2002).

[10] C. Ziegler and G. Lausen, 'Analyzing correlation between trust and user similarity in online communities', in *Proceedings of the Second International Conference on Trust Management*, (2004).

[11] C. Ziegler and G. Lausen, 'Spreading activation models for trust propagation', in *IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE'04)*, (2004).