Trust-enhanced Recommender Systems for efficient on-line collaboration

Georgios Pitsilis [1]

Abstract. Trust has been explored by many researchers in the past as a solution for assisting the process of recommendation production. In this work we are examining the feasibility of building networks of trusted users using the existing evidence that would be provided by a standard recommender system. As there is lack of models today that could help in finding the relationship between trust and similarity we build our own that uses a set of empirical equations to map similarity metrics into Subjective Logic trust. In this paper we perform evaluation of the proposed model as being a part of a complete recommender system. Finally, we present the interesting results from this evaluation that shows the performance and benefits of our trust modeling technique as well as its impact on the user community as it evolves over time.

Keywords: Recommender Systems, Subjective Logic, Trust Evaluation.

1 Introduction

Recommender Systems incorporate a specific type of information filtering that has the purpose of presenting information items that are likely of interest to some user. They are widely used in e-commerce sites like *Amazon*[1] and *ebay*[2] with the aim of helping users to choose products they might like. The contribution of recommender systems comes in two forms, either as predicted ratings of services that a user wants to know about, or as lists of services that users might find of interest.

The best known technique that is used in Recommender systems is *Collaborative Filtering* CF [3]. The idea behind CF if the formation of a graph of virtual relationships that may exist between the users and is done by applying statistical techniques upon the preferences of users. The correlation of user ratings is expressed with a metric called *Similarity* and it can be calculated using the mathematical formula of *Correlation Coefficient.*

Georgios Pitsilis
Center of Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, O.S. Bragstads plass 2E, NO-7491, Trondheim, Norway
e-mail: pitsilis@q2s.ntnu.no

[1] This work has been carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme.

However, *Recommender Systems* and particularly CF are not perfect and because of the sparse datasets used they appear to have weaknesses such as provision of low quality predictions (known as the *false negatives* and *false positives* problems [4]), as well as low coverage. With coverage we refer to the number of accurate predictions that can be offered to users. A prediction is expressed as a level of likeliness of a user for some particular product.
Also, the architectural characteristics of CF are known to be vulnerable to attacks from malicious and libelous users. It is the case today that recommendation provision is done by centralized entities which require access to all users ratings and preferences in order to do the appropriate correlations. Given that such data should be considered as confidential information of people it is required that such services be run by trusted authorities. Ideally such services could be run in such a way that users or entities that act on behalf of them would be able to perform the appropriate correlations and work out predictions even though they may have not been provided with full access to this information. Also, performing these correlations requires computing power which increases exponentially with the number of users and ratings [5], and that imposes a scalability problem.
Trust has been a research concept in the past as a potential solution to overcome many of the problems of recommender systems [6][7][8][9]. In our approach it is used for extending the neighboring base of users that take part in the collaborative filtering system achieving in this way the benefit of increased number of predictions that can be performed. In addition, this improvement is found to be very supportive for new users, who despite their little contribution in terms of recommendations, they can exploit the benefits of their participation early on.
This work is an extension to previous research that has been done in the past aiming to model trust for collaborative filtering systems in which trust is derived directly from user ratings [10]. In this paper we are attempting an evaluation of those modeling approaches using data from a real Recommender System. More particularly, in this work is demonstrated the benefits that users receive in terms of: a) *Accuracy* of predictions for items that users have not experienced yet and b) *Rapidity* that such information becomes available in the system for a given number of user ratings that have been gathered up to a given time. A low value is indicative of the existence of a problem which in recommender systems is known as *Cold Start Problem* [11].

2 Motivation

Recommender Systems (RS) are widely used nowadays and in simple terms their purpose is to suggest items, usually products, to those who might be interested in them. The main idea is to get the users that participate in such system correlated based on the opinions they have expressed in the past, with the aim to work out predictions of ratings for services or products for any interested user. The techniques used in the contemporary RS are basically based on the idea of predicted ratings being computed on ratings provided by k like-minded individuals.

RS often exist as services embedded into commercial web sites but also exist as services for supporting and providing data to researchers that are particularly interested in investigating problems of this area. *Movielens* [12], *BooksCrossing* [13] and *Netflix* [14] have been built with the sole purpose of supporting research activities. Technologies that have been applied to RS include *Nearest-neighbor* (which includes *Collaborative filtering*), *Bayesian networks* [15] and *Clustering* [16]. Bayesian networks create a decision tree based on a set of user ratings. Despite their speed in providing recommendations they are not practical for environments in which user preferences are updated regularly. In Clustering, users are grouped by their similarity in preferences and predictions are made regarding the participation of a user in some cluster.

The basic idea behind CF is to make predictions of scores based on the heuristic that people who agreed (or disagreed) in the past are likely to agree (disagree) again. Even though such a heuristic can be sufficient to correlate numerous users with each other, systems that have employed this method still appear to be highly sparse, due to the fact that people are often unwilling to provide their feedback. As a result, systems are ineffective at making accurate predictions all the time. By *Sparsity* we mean a lack of shared experiences required for a CF system to work. The Cold start problem [4], is related to *Sparsity* and it is due to the low number of ratings that new users contribute to the system. As a result new users become isolated and hence cannot receive good quality recommendations. Apart from the Cold Start problem conventional RS face other problems such as their *Vulnerability to Attacks*.

Establishing other type of relationships including *Trust*, that could be developed between the users, especially new ones, might be helpful for increasing their contribution to the *CF* system. In this work we attempt to go one step beyond and investigate how some Trust modeling technique could outperform the traditional *CF*. More importantly we are interested to know how this benefit could become available as early as the community is still being formed and that is when the system needs it mostly. As the Cold Start problem emerges mainly during the system initialization, some demonstration of a potential solution is necessary to be accompanied by the appropriate evidence that show how the system performs over time. Therefore, it has been attempted an extensive evaluation to that captures the development of the user community.

3 Background Research

Trust has been proposed as a solution to alleviate the weaknesses of the standard collaborative filtering technique and various trust-based approaches of k-nearest neighbor algorithm have been introduced by many researchers in the past. The work done by Lathia et. al. [7] is focused mostly on finding the k-trusted neighbors rather than the k-similar ones to forming groups of collaborative users. Massa et. al. [6] considers the problem of receiving poor results as the inability of the re-

commender system to exploit other sources of the information such as the Web-of-Trust and he proposes a way of finding trustworthy recommenders via a friend-of-friend finder scheme.
For reference we mention work that has been done by other researchers in the area of *CF* to tolerate similar problems that we intended to do. In [17] Sun, Kong, Ye attempt a comparison between Person's approach, Singular Value Decomposition and Scale and Translation Invariant. In that work Pearson's approach seems to behave better during the startup phase and hence it renders more suitable for tackling the Cold Start problem. Quercia, Hailes, Carpa [18] have investigated a solution for computing trust in collaboration systems but in their proposed model the recommender's trustworthiness is not taken into account in the calculation of derived trust. In [19] there is a technique for improving collaborative filtering based on some idea of removing global effects and in estimating the interpolation weights for each weighting factor in the Collaborative Filtering. As a result to these the estimation accuracy is improved. The work in [20] describes a framework for building hybrid CF systems which combine content and collaboration. The interesting bit of this work is the idea of setting weights on the contribution of similarity by introducing a factor which is based on the number of common items that exist in a relationship. O'Donovan and Smith [21] have introduced the idea of composing a trust value that is analogous to the percentage of accurate predictions of items in which the error is lower than a predefined threshold. In this way, the "neighbors" are filtered to the trusted ones which finally used for building up the recommendation.
To our knowledge, matters like the evolution of user communities in trust-enabled Collaborative Filtering Systems and their effectiveness against problems like the Cold-Start has not been investigated adequately so far. Instead, the main focus has been on the adaptation of trust methods onto *CF* or on the alleviation of problems mentioned above using solutions that do not involve Trust.

4 Description of the Proposed Idea

Our concept is based on the idea of extending the neighboring base of users by supporting the existing similarity relationships by trust relationships that can be transitively propagated throughout the network of users. In contrast to other known research approaches we used an algebra called *Subjective Logic* [22] for calculating the derived propagated trust along chains of users of known trustworthiness. In contrast to previous studies and as we are keener in capturing the benefits of using trust while a user community is being developing, in this experiment we have measured the values of the various properties of the virtual community at standard time intervals.
As it is more important to know the impact of the modeling technique as it is seen from the actual user's point of view, we attempt evaluation of a whole recommendation production cycle on which the proposed modeling has been applied on. In this way, it can be estimated the contribution of each individual mechanism (trust

modeling, trust propagation etc.) to the prediction error of produced recommendations.

Next, we describe the concept of the recommendation production system we are proposing and it roughly can be considered as an extension to the existing recommendation production mechanism that is used in the CF systems today. In order to make easier to the reader to realize how our system can fit into an existing RS mechanism we provide a high level view that illustrates the individual operations.

In standard CF systems the correlation of user ratings is done on a nearest-neighbor basis which requires that correlated parties must have at least a minimum number of common experiences. According to this, only knowledge within a radius of one hop from a referenced node is exploitable. As will be seen later, knowledge that happens to exist at longer distances can also be made exploitable via the trust network, provided that the required mappings and transformations from trust metric to similarity can be performed.

We can imagine the entire view of the system being similar to a graph in which users are represented by vertices and the similarity relationships between them by edges. In a simplified scenario we are supposing that users B and C have experienced some product k that user A is interested in knowing how much she might like it. In addition, users A and B could be potentially related via a similarity relationship that might exist and would normally be captured in an ordinary CF system. Moreover, we assume that user C is not related with A, as not enough evidence can be gathered to build up a similarity relationship. Considering the fact that similarity can be calculated for nodes are at distance of 1 hop from each other, similarity between nodes that are at longer distances could be derived from the trust between them, which in that case is called *indirect trust*. If we assume that trust can be transitive in long chains [23], the indirect trust can be calculated by applying the appropriate algebra on the direct trust of all entities that reside between the two nodes in the graph. Finally the derived indirect trust can then be converted into equivalent similarity. In this case, a possible lack of similarity relationships can be replaced adequately by similarities derived from Trust.

The requirement for trust to become transitive in long chains obeys that a common purpose exists along the chain. According to this, only the last trust relationship should be considered with trust for a certain purpose (functional trust) and all the other trust relationships in the chain should be with respect to the ability to recommend for the given purpose (recommender trust).

In more complex scenarios there might be multiple Trust paths going further than two hops away from the originator and finally ending to the target user which when combined together they provide a single Trust value. That Trust value which denotes how much A would trust C is again replaced by an equivalent similarity value which next is applied to Resnick's formula [24] for computing the predicted rating.

In figure 1 below it is shown how the process of recommendation production is carried out in our system. We distinguish two main sub-paths in the rating prediction, namely the *direct* and the *indirect* one. The direct one exists only if there is direct relationship between the user C, that has already rated some item which user A is interested in, and is depicted as "Similarity A,C_1" . That path can co-exist along with the indirect one. The indirect one requires that both A and C have established a similarity relationship with third entities B of which the direct trust for A and C can be computed. Then the hypothetical trust between A and C is derived and converted into hypothetical similarity A,C2. Finally, the rating prediction that employs Resnick's formula may use data derived from both paths. As an effect of this, a rating prediction can be based on a number of direct and indirect paths.

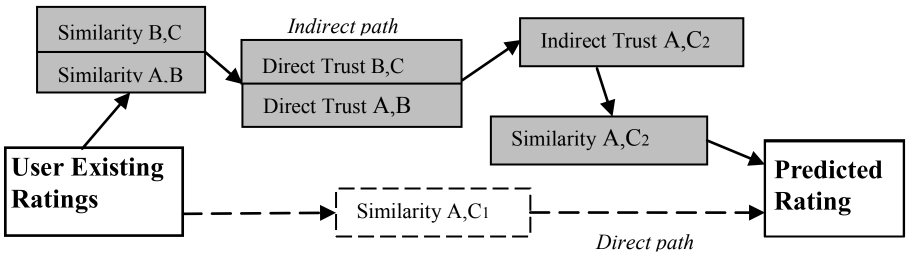

Figure 1. The high level view of the system.

There is a variety of models today for computing Trust in long chains of trustees [25][26], with various advantages and disadvantages. In our evaluation as mentioned earlier we chose *Subjective Logic* algebra to compute indirect trust using the opinions from user ratings. With *opinion* we refer to a metric of Uncertain Probabilities theory [27] which expresses the belief. Because there is always imperfect knowledge, as opinions are based on observations, lack of knowledge should be considered when assessing them. Subjective Logic framework deals with the absence of both trust and distrust by introducing the *uncertainty* property in opinions. This framework uses a simple intuitive representation of uncertain probabilities by using a three dimensional metric that comprises belief (b), disbelief (d) and uncertainty (u). which constitute an opinion. For building up opinions requires that evidence come in such a form that opinions of (b,d,u) can be derived from and thus be better manageable due to the quite flexible calculus that the opinion space provides. Unfortunately, evidence usually is available in forms that are essentially more understandable to humans. A previous work [10] we proposed for transforming data from a recommender system into the suitable format for *Subjective Logic* was necessary as this algebra can be applied onto data that come in the form of opinions. We didn't find suitable to representing opinions based on the Beta Distribution Probability Function [28] as it requires data in the evidence space to be provided strictly in binary form. More particularly, according to Beta distribution function modeling data represent two possible outcomes of a process X or $\overline{X}$ and its behavior is described by the number of X and $\overline{X}$ that derive from the set of observations.

As data in recommender systems are in different forms we came up with an alternative modeling solution the formulas of which we describe in the next section. In the current work we measure the impact of this modeling approach on the development of the Trust network.

As the Cold Start problem is considered as a time related issue we are mostly interested in knowing how the use of the Trust network can be exploited best so that the new users in the system can receive the benefit of their participation whenever they need it. As there are more than one candidate formulas for modeling trust from existing evidence the purpose of trying them all was two-fold. First, to identify if the use of trust in general can tolerate problems as mentioned before, and second to find the best candidate.

5 Experimental Evaluation

The main challenge we faced in this work was concerned with the demonstration of the evolution of the trust network which would identify the best trust modeling candidate formula for a system of which experiences grow as the time develops. In order to achieve that, it is required to be known the time at which every single recommendation has been submitted to the system, or at least the order in which all the examined recommendations have been submitted. To fulfill this requirement we used a publicly available dataset from a Movie recommender system which was provided with the time information for every rating.

As mentioned earlier data from Recommender Systems are usually available in forms not suitable to be processed by a trust algebra. In this respect, we came up with various modeling approaches for converting the data into beliefs [10] and we finally concluded that we should perform analysis of the behavior of only 3 candidate formulas. The available data sample we used for the experiment was taken from a real collaborative filtering system for data captured during a period of 812 days. More specifically we used data from *Movielens* recommender system [12] from which we built up virtual sample communities of 100 users large. This size was chosen as being optimal for demonstrating the performance of our concept and at the same time keeping the evaluation time within satisfactory limits. Computation time is found to be an issue in such experiments, for instance we mention that in a 2.5 GHz single core CPU the computation time of a single time instance for all 5 samples of 100 users finally exceeds two days.

The 3 alternative formulas shown below were those used for shaping the belief property (*b*) to be used by *Subjective Logic*, from evidence such as *user Similarity*.(shown as CC). The first formula in total was applied for 3 different values of *k*.

$$b = \frac{1}{2}(1-u)\left(1+CC^{K}\right) \qquad \textbf{(1)}$$

Equation (1) is used for values of k=3,1/3 and 1. The k=1 denotes linear transformation. CC is the similarity value (known as *Correlation Coefficient*).

$$b = \frac{1}{2} + \left(\frac{\arcsin(CC)}{\pi} \right) \cdot (1-u) \qquad \textbf{(2)}$$

$$b = \frac{1}{2}\left(\sin(CC \cdot \frac{\pi}{2}) + 1 \right) \cdot (1-u) \qquad \textbf{(3)}$$

For calculating the uncertainty we used the simplified formula: $u = (n+1)^{-1}$, in which n denotes the number of common experiences in a trust relationship between two parties A and B. For more information about the formulas used see [10]. For being able to compare the performance of the Trust enabled approach against the standard CF technique we also tried the standard CF algorithms onto the experimental data.
We performed a series of tests in which the prediction accuracy was expressed in MAE between the real rating and its predicted value (noted as recommendation) using the leave-one-out technique. Moreover, due to the unstable behavior of Pearson's similarity we considered that a similarity relationship between two users exists only if there are at least 10 common experiences.
With regard to measuring the evolution of the system we performed the experiments in time frames which differ from each other in the number of ratings that were considered for calculating a recommendation. As the number of recommendations performed at every time stage is more important to be shown than the timestamp information we considered as the best solution to present the adjacent *sparsity* value. With sparsity we refer to the percentage of empty cells in the matrix of users by items. Non-empty cell denotes existence of rating for this item from a particular user. The algorithm used for the evaluation of the total system is presented in fig.2. We call *iTrust(i,j)* the indirect trust between entities *i* and *j*.
As it has been studied in the past [8] there is no reason for searching for trusted neighbors at distances beyond than 2 hops away from the querying node as there is no significant benefit with regard to the cost of searching. Therefore in our experiments the searches were constrained to propagate up to a max distance of 2 hops.
The sample we used comprised 73871 ratings of 500 users divided into 5 sets of 100 users. In order to study the evolution of the measured properties the ratings of each of 100 users were divided into 13 subsets based on their attached Unix type timestamp information. In this way, each subset of ratings would contain roughly the same quantity of scores that have been submitted within the same time slot.

In order to study the advantages of our system against the cold start problem we introduced the following two metrics called *System Coverage Gain* and *User Coverage Gain,* the former being a system-centric metric and the latter user-centric. The purpose of introducing them is solely to demonstrate the actual benefit that users receive when they make use of the trust graph. The metrics are computed at every timestamp TS as the system evolves. We define each metric as:

```
Let K be the set of all users
Let t the examined timestamp
Let R bet the set of all ratings
Let Rt ⊂ R be the set ratings submitted before time t
Let Ru ⊂ Rt be the set of the ratings of some user u
Let Ki ⊂K : |Ru| ≥ 10                                  * for users who have at least 10 ratings < t
For all i ∈ Ki do                                       * for all users that belong to this category
  Let Ei ⊂ Ru                                           * The set of ratings of i
  For all w ∈ Ei do                                     * for all items of user i
   Let M ⊂ Ki : ∀ p ∈ M, Ep ⊂ Rt, |Ei ∩ Ep| ≥ 10 ∧ w ∈ Ep
                                                        * users that have experienced w
    Pset ← {}                                           * set of users used for rating prediction
    For all j ∈ M do                                    * all other users
      SimN ← {}, TrN ← {}                               * zero counters
      S ← CC(i, j)                                      * Pearson's similarity *
      If ( S = null )
            T ← iTrust(i, j)                            * Derived Indirect trust *
            TrN ← TrN + {i}
            If ( T ≠ null )   S ← f(T)                  * similarity derived from trust
      Else
            SimN ← SimN + {i}
      EndIF
      If (( S ≠ null ) ∨ ( T ≠ null ))  Pset ← Pset + {i}   * include user i in rating prediction
   EndFor j
   p(w) ← predict(w, Pset)                              * predict value over Pset
   MAE ← |p(w) − w|                                     * Absolute Mean Error value *
  EndFor w
EndFor i
Average(MAE)
TrustContr = |TrN| / (|TrN| + |SimN|)                   * trust contribution
```

Figure 2. The evaluation algorithm.

a) System Coverage Gain

This metric is characteristic to the benefits that new users receive during their early stages in using the system and SCG represents the relative benefit of the Trust-enabled method over the standard CF. In order to calculate this at every timestamp it was necessary that all rating predictions that had been produced in that particular timestamp were summed up for both the standard CF algorithm used and the Trust-enabled CF. The formula used to calculate the System Coverage Gain is:

$$SCG = \left[\frac{prd(TR) - prd(CF)}{prd(CF)} \right]_{TS}$$

TS indicates the particular timestamp the SCG is computed for. *prd(TR)* is the number of predictions made by all users up to time TS if used the trust-enabled method. *prd(CF)* is the number of requests that would have been made by all users up to time TS if the standard CF method had been used. In this metric all ratings are considered equally the same, no matter if they come from new users or from users who have been using the system for quite a long time. Therefore SCG should be considered as the degree of opportunities that the system provides to users for making predictions for items they are interest in. In simple words, SCG expresses how the cold start problem is seen from a general point of view.

b) User Coverage Gain

Contrary to *System Coverage Gain* this metric demonstrates the benefit as it is seen from the point of view of a new user. This metric is found partially useful as this category of users is the one mostly affected by the cold start problem. On some specific timestamp TS all users who haven't supplied a single rating in the system are marked and the number of ratings they have provided is counted. The benefit every user receives on average at time TS is equal to the total number of ratings supplied by new users normalized by the actual number of new users who have encountered their first experience at that time slot. For being able to show the advantage of our technique over the standard CF, it is necessary that the above metric is been calculated for predictions achieved for both the Trust-Enabled system and a hypothetical system that employs the standard CF. The formula that is used to calculate the *User Coverage Gain* is:

$$UCG = \left[\frac{\frac{prd(TR)}{users(TR)} - \frac{prd(CF)}{users(CF)}}{\frac{prd(CF)}{users(CF)}} \right]_{TS}$$

Where *prd(TR)* is the number of ratings that have been supplied by new users at the time slot that ends at TS and have made use of the trust graph for performing these recommendations. Likewise, *prd(CF)* is the number of ratings that have been supplied by new users during timestamp TS and used the standard CF method. Also, *users(TR)* and *users(CF)* are the sizes of the populations of new users that have made use of the Trust-enabled technique and the standard CF respectively at time TS. In principal, all the TR-related metrics should receive higher values than the corresponded CF as the use of trust graph almost always increases the possibility for more recommendations to be produced.

6 Discussion of the Results

Figure 3 presents how the prediction error evolves as the time develops. In that diagram, as in all results diagrams, the time is represented by its adjacent sparsity value and it is shown across the horizontal axis. As can be seen all modeling approaches have worse performance than the standard CF almost at all timestamps. The only exception is at the first timestamp where almost all modeling approaches appear to give better results than CF. In that sense the Trust-enabled System looks less prone to the cold-start problem and thus can provide a slight benefit to the new users as the system is being built up. There is a likelihood that this benefit is being maintained for longer than the system initialization phase. For instance, in one of the 5 datasets the prediction error for all trust modeling alternatives retained lower figure than in the standard CF for the first 3 consecutive timestamps. Moreover, the penalty in accuracy for using the trust-enabled system instead of the standard CF is not very significant as its error is never higher than 2% than the error of CF. The other interesting observation is the exceptionally worse behavior of the type 2 modeling approach compared to any other approach. There is though a converging behavior with the other 4 approaches towards the end of the simulation. In conclusion, the use of trust graph does not incur a significant penalty in terms of accuracy, and yet more important, performs well during the beginning of the community formation.

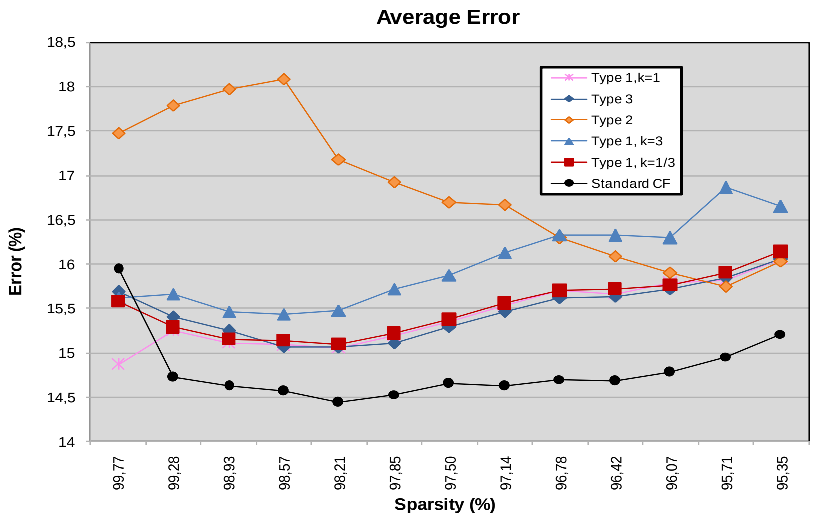

Figure 3. Error of trust-enabled system

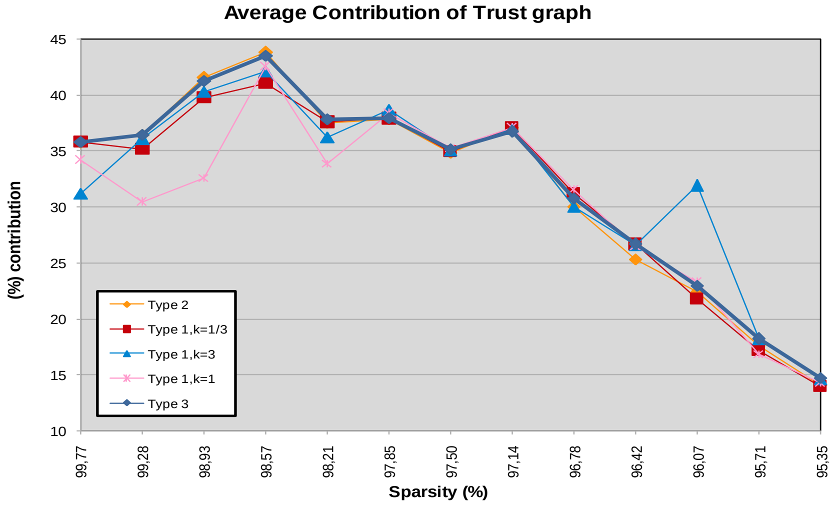

Figure 4. Contribution of trust graph.

Figure 4 shows the *Contribution of Trust Graph* for every recommendation produced. More specifically for every recommendation both the number of trusted neighbors and the total number of neighbors (trusted and similar) which have experienced the recommended item in the past are counted. We define *Contribution of Trust Graph* as the ratio of the above two values. As can be seen from the diagram, in contrast to *Prediction Error*, this metric follows a decreasing trend, but more importantly, its maximum value appears during the beginning of the community formation when sparsity is still high.
A careful examination of the correlation values of all case studies, presented in table 1, reveals the existence of high positive correlation between the *Contribution of Trust Graph* and the *Prediction Error* when the type 2 modeling formula is used. More specifically in all 5 examined user communities the correlation value appears on average to be as high as CC=0.84. That means, the more use of the trust graph is done the worse results are being received. The logical conclusion is that this modeling approach should be rejected as in the long term it does not provide any significant benefit over the other alternative approaches and according to the above evidence it is inappropriate for our work.

Table 1. Correlation values between Trust contribution and Prediction Error

	Transformation Formula				
Sample	*Type 3*	*Type 2*	*Type1, k=1/3*	*Type 1, k=3*	*Type 1, k=1*
1	-0,3662	**0,7224**	-0,4275	-0,5121	-0,3389
2	-0,2633	**0,7393**	-0,3044	-0,0235	-0,1579
3	-0,5857	**0,6760**	-0,6649	-0,6114	-0,0931
4	-0,7263	**0,9576**	-0,9023	-0,7201	-0,6816
5	0,1958	**0,6888**	0,0432	0,2432	0,0291

In all other modeling approaches the evidence shows no stochastic relationship between the error and the trust graph contribution (except for some exceptional cases) and hence there is no reason for not using them. In some exceptional dataset though (sample 4) strong negative correlation was found in the above two metrics when any modeling approach other than type 2 is used. A good interpretation of this could be: the quality of predictions is benefited by the use of trust graph. As this occurred only in one of all data sets used in the experiment we conclude that the evidence is not strong enough to support the positive claim. Hence, in practice the quality of predictions can be not affected by the decreasing use of the trust graph as the time progresses.

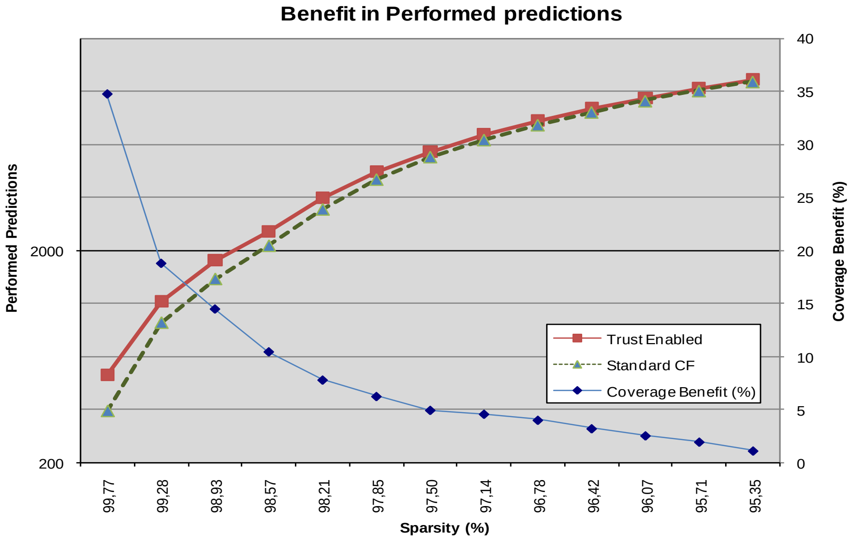

Figure 5. The Gain and the benefit in terms of performed recommendations.

In the following results we present comparisons between only one of the 3 alternative modeling techniques, which for short will be called *trust-enabled*, and the standard Collaborative Filtering technique (no use of trust graph). That is because all trust modeling techniques achieve almost the same levels of the examined metric. This comparison between the trust-enabled and the standard CF is shown pictorially in fig. 5 as figures of produced recommendations. The figure is in logarithmic scale. In continuous line is shown the recommendations that can be produced when the trust-enabled system is used and in dashed line the recommendations that can be produced if applying the standard CF. Note that in the graph the total number of recommendations shown is that which has been provided by all users together, new and experienced ones. Hence, it renders useful as a system-centric metric.

For clarity we also present in the same figure the *Benefit* in terms of produced recommendations when the trust system is used. As can be seen, the high *Benefit* during the early stages of the system development (first timestamp) is followed by a sharp fall until the half of the evaluation time, and finally falls slowly until the end. The very low *Benefit* (lower than 5%) that is received in the second half of the simulation time is the result of saturation that occurs as more and more relationships are being established. Hence, the decreasing trend in that figure can be justified as been a consequence of the increasing number of submitted recommendations. That renders the use of trust system unnecessary as distant users are becoming reachable via similarity relationships. Finally, the *Benefit* minimizes when the similarity graph and the trust graph perfectly match. That suggests undoubtedly the use trust of graph should be made during the early stages of the system development as it is then more useful for the users.

The next two diagrams are referred to the actual gain in terms of produced recommendations. More specifically in 6 is illustrated the total number of recommendations produced by new users only, for all 5 sample communities, at each timestamp. With "new users" we refer to those who joined the system at that specific timestamp. For comparison we display the results for both the standard method and the trust-enabled system. The importance of this graph is that it focuses on the new users only and thus makes distinguishable the benefit received by users who have used the system for the first time. In contrast to the results shown in fig. 5. the user-centric view in fig. 6 shows that even though the trust-enable technique is again beneficial for new users, the actual number of recommendations produced by new users falls sharply after reaching a peak at the middle of the simulation.

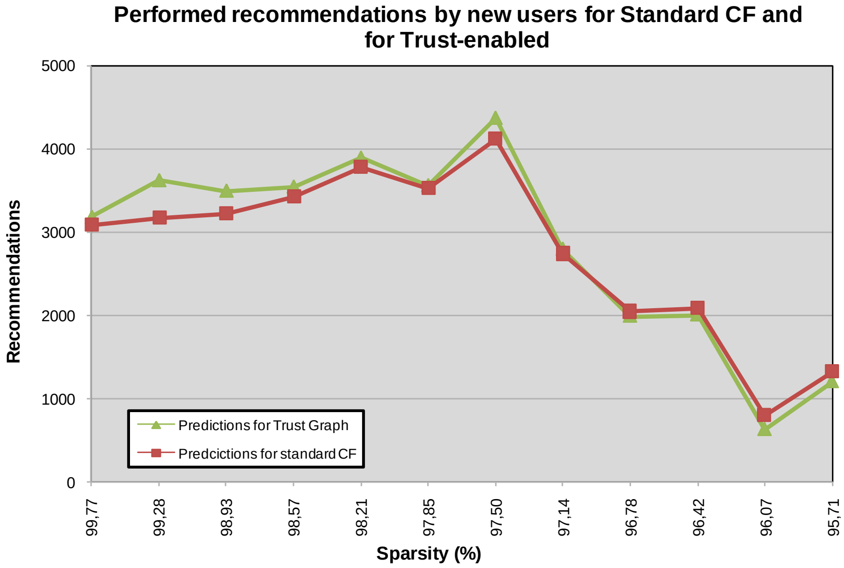

Figure 6. Recommendations produced by new users.

Another interesting observation is that during the first half of the simulation the number of recommendations, when the trust-enabled method is used, is always higher than the recommendations produced via the standard CF, which indicates quicker and more efficient user discovery in the proposed system rather than in the standard CF. However, after reaching the peak in the diagram the standard CF outperforms the trust-enabled method in terms of new recommendations. That is because the proposed system is more efficient in terms of speed at which the new users can make use of the system and thus submit recommendations. As a result, the recommendation discovery process is progressing faster than in the standard CF.

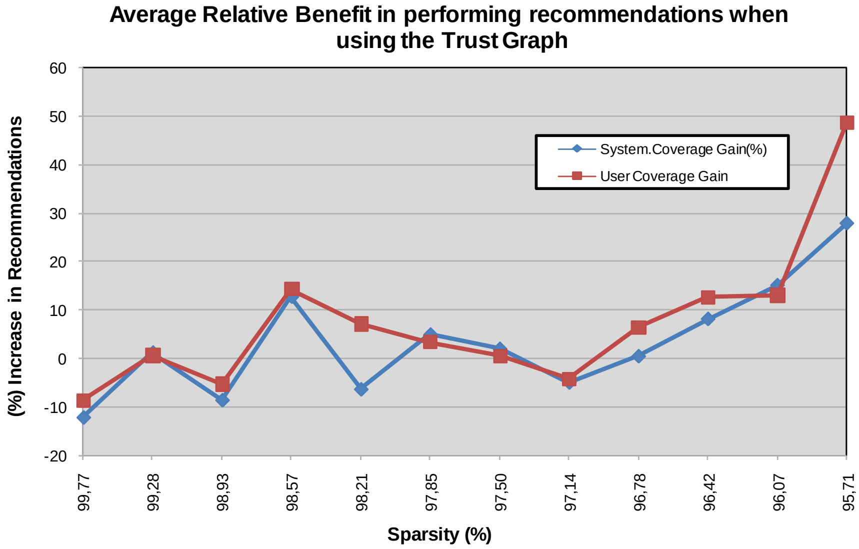

Figure 7. Progress of SCG and UCG.

Finally, fig. 7 shows the benefit in using the trust enabled CF expressed in metrics of *System Coverage Gain* and *User Coverage Gain* as they develop over time. The figure of UCG can be interpreted as: the benefit of new users in fact increase as the system develops as these users who join the system later actually receive more benefit than those who join early. As can be seen, both *User Coverage Gain* and *System Coverage Gain* follow nearly the same trend throughout the experiment but UCG almost at all time frames appears to be higher. Our justification is, the users on average receive more benefit than the system can observe. The negative values received for both UCG and SCG at the beginning indicate a momentary advantage of the standard CF over the trust-enabled one. In relation to diagram 6, the surprising observation is that the decreasing rate of new users' recommendations in the second half of the simulation time does not affect the average benefit they receive.

7 Conclusions and Future Work

We presented evaluation of a trust modeling technique with the purpose to investigate if the deployment of trust-oriented approaches could help in the alleviation of time-dependent problems. We used ratings taken from a real recommender system and we introduced metrics for expressing the benefit as seen both from the user and the system point of view.

The short experiments we performed confirm that new users do receive benefits by using the trust system, as they become more capable of performing more predictions than before. More specifically, compared to the standard CF, the method appears to provide higher potential to the actual user during the startup of the community as the prediction accuracy is maintained at very good levels. In the positive aspects we can include the faster system development as well as the fact that the quality of predictions is not affected by the decreasing use of the trust graph as the time progresses.

The increasing trend of the prediction error in the trust-enabled system as the time develops strongly suggests the use of the trust system during the startup phase, no matter which modeling approach will be chosen. As regard to the question which trust modeling formula is best for converting evidence into user opinions, the tests show there is no single formula that behaves optimally at all time instances.

As far as the benefit of the evolution test concerned, the experiment described in this paper helped very much in revealing the above findings, as static tests we performed in the past, applied on the final time instance of the same data set, had driven to very general conclusions.

The performance of CF algorithms are known to be subjective to the datasets are operate on. Therefore, the value of this work is restricted to the type of dataset used in the experiment. It would be a great advantage if tests with more datasets could be performed that would either confirm our conclusions or reveal new properties that could improve the way performance develops over time. For example, it could be investigated any likely dependency between the time step at which the traditional method outperforms the trust-enabled approach and the quality of the submitted recommendations. The objective of this would be to extend the time period during which the new users receive benefit.

For deploying such a trust-enabled system into a peer-to-peer infrastructure, in which users can join the trust community consciously, it is crucial that their participation is maintained for the longest possible period of time. Achieving this objective means getting Trust-enabled Recommender systems to work more efficiently and thus enhancing users' collaboration.

References

1. Amazon, Electronic Commerce Company, http://www.amazon.com
2. Internet Company for online Auctions and shopping, http://www.ebay.co.uk
3. J. Herlocker, J. Konstan, L. Terveen, J. Riedl, Evaluating collaborative filtering Recommender Systems, In ACM Transactions on Information Systems, vol. 22, pp. 5-53, ACM Press (2004)
4. D. Maltz, K. Ehrlish. Pointing the Way: Active Collaborative filtering, In proc. of CHI-95,pp.202-209,New York, ACM Press (1995)
5. B. M. Sarwar, Sparsity, Scalability, and Distribution in Recommender Systems, Doctoral Thesis, UMI Order Number: AAI9994525., University of Minnesota (2001)

6. P.Massa, P.Avesani, Trust-aware Collaborative Filtering for recommender Systems, In proc. of CoopIS/DOA/ODBASE (1), pp.492-508 (2004)
7. N. Lathia, S. Hailes, L. Carpa, Trust-Based Collaborative Filtering, in proc IFIPTM, Springer, Vol. 263, pp.119-134, Trondheim, Norway (2008)
8. G. Pitsilis, L. F. Marshall, Trust as a key to improving recommendation systems, in Proc 3rd International Conf. iTrust2005, Paris, France, Springer, pp.210-223 (2005)
9. M. Papagelis, D. Plexousakis, T. Koutsouras, Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences, In proc. of iTrust, Springer , pp.224-239, Paris (2005)
10. G. Pitsilis, L. F. Marshall, Modeling Trust for Recommender Systems Using Similarity Metrics, in proc. IFIPTM, Springer, Vol. 263, pp.103-118, Trondheim, Norway (2008)
11. A. I. Schein, A. Popescul, L. H. Ungar, D. M. Pennock, Methods and Metrics for Cold-Start Recommendations. In Proc. 25th Annual International Conference on Research and Development in Information Retrieval, ACM, pp.253–260, New York (2002)
12. B. Miller, I. Albert, S. K. Lam, J.A.Konstan, J. Riedl, MovieLens unplugged: experiences with an occasionally connected recommender system, In Proc. of the 8th international conference on Intelligent user interfaces,Accepted poster, pp.263-266, Miami, Florida, USA (2003)
13. C. N. Ziegler, S. M. McNee, J. A. Konstan, G. Lausen, Improving Recommendation Lists Through Topic Diversification, in Proc. 14th International Conference on WWW,pp.22-32, ACM Press, NewYork (2005)
14. Netflix Inc., http://www.netflixprize.com
15. S. Zhan, L. Liu, An Online Personalized Recommendation Model based on Bayesian Networks, in proc IFIP, Vol 255, Research and Practical Issues of Enterprise Information Systems II, Vol. 2, Springer, pp. 1575-1584 (2007)
16. J. S. Breese, D. Heckerman, C. Kadie, Emperical Analysis of Predictive Algorithms for Collaborative Filtering, In Proc. of the 14th Conference on Uncertainty in Artificial Intelligence, Morgan-Kaufmann, pp.43-52 (1998)
17. X. Sun, F. Kong, S. Ye, A Comparison of Several Algorithms for Collaborative Filtering in Startup Stage, In proc Networking Sensing and Control, IEEE, pp.25-28 (2005)
18. D. Quercia, S. Heiles, L. Carpa, B-trust: Bayesian Trust Framework for Pervasive Computing. In proc 4th International Conf. iTrust 2006. Lecture Notes in Computer Science (Vol.3986/2006). Springer, pp. 298-312 (2006)
19. R. Bell, Y. Koren, Improved Neighborhood-based Collaborative Filtering, In proc IEEE International Conference on Data Mining, pp.7–14 (2007)
20. P. Melville, R. J. Mooney, R. Nagarajan, Content-Boosted Collaborative Filtering for Improved Recommendations, In proc of Eighteenth national conf. of Artificial Intelligence, pp.187-192, ISBN:0-262-51129-0 (2002)
21. J. O'Donovan, B. Smyth , Trust in recommender Systems, in Proc. 10th International Conf. on Intelligent User Interfaces, San Diego-USA (2005)
22. A. Jøsang, A Logic for Uncertain probabilities, International Journal of Uncertainty, fuzziness & Knowledge based systems,Vol.9, No.3 (2001)
23. B. Cristianson, W. Harbison, Why isn't Trust Transitive?, In Proc. of the International Workshop on Security Protocols, pp.171-p176, Springer, (1997)
24. P. Resnick, H. R. Varian, Recommender Systems, Communications of the ACM. 40(3), pp. 56-58 (1997)
25. A. Jøsang, E. Gray, M. Kinateder, Analyzing topologies of Transitive Trust, In Proc. of the Workshop of Formal Aspects of Security and Trust, (FAST 2003), Pisa (2003)
26. A. Rahman, S. Heiles, Supporting trust in Virtual Communities, In Proc. International conference on System Sciences, Hawaii (2000)
27. G. Shafer, A Mathematical Theory of Evidence, Princeton University Press (1976)
28. A. Jøsang, R. Ismail. The Beta Reputation System, In Proc of the 15th Conference on Electronic Commerce, Bled, Slovenia (2002)