Trust in Recommender Systems

John O'Donovan & Barry Smyth
Adaptive Information Cluster
Department of Computer Science
University College Dublin
Belfield, Dublin 4
Ireland
{john.odonovan, barry.smyth}@ucd.ie

1. ABSTRACT

Recommender systems have proven to be an important response to the information overload problem, by providing users with more proactive and personalized information services. And collaborative filtering techniques have proven to be an vital component of many such recommender systems as they facilitate the generation of high-quality recommendations by leveraging the preferences of communities of similar users. In this paper we suggest that the traditional emphasis on user similarity may be overstated. We argue that additional factors have an important role to play in guiding recommendation. Specifically we propose that the trustworthiness of users must be an important consideration. We present two computational models of trust and show how they can be readily incorporated into standard collaborative filtering frameworks in a variety of ways. We also show how these trust models can lead to improved predictive accuracy during recommendation.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Human Factors, Reliability

Keywords

Recommender Systems, Collaborative Filtering, Profile Similarity, Trust, Reputation

2. INTRODUCTION

Recommender systems have emerged as an important response to the so-called information overload problem in which users are finding it increasingly difficult to locate the right information at the right time. [19, 3, 20] Recommender systems have been successfully deployed in a variety of guises, often in the form of intelligent virtual assistants in a variety of e-commerce domains. By combining ideas from user profiling, information filtering and machine learning, recommender systems have proven to be effective at delivering the user a more intelligent and proactive information service by making concrete product or service recommendations that are sympathetic to their learned preferences and needs.

In general two recommendation strategies have come to dominate. Content-based recommenders rely on rich content descriptions of the items (products or services for example) that are being recommended [13]. For instance, a content-based movie recommender will typically rely on information such as genre, actors, director, producer etc. and match this against the learned preferences of the user in order to select a set of promising movie recommendations. Obviously this places a significant knowledge-engineering burden on the designers of content-based recommenders since the required domain knowledge may not be readily available or straightforward to maintain. As an alternative the collaborative filtering (CF) recommendation strategy provides a possible solution. It is motivated by the observation that in reality we often look to our friends for recommendations. Item knowledge is not required. Instead, collaborative filtering (sometimes called social filtering) relies on the availability of user profiles that capture the past ratings histories of users [3, 16]. Recommendations are generated for a target user by drawing on the ratings history of a set of suitable recommendation partners. These partners are generally chosen because they share similar or highly correlated ratings histories with the target user.

In this paper we are interested in collaborative filtering, in general, and in ways of improving its ability to make accurate recommendations, in particular. We propose to modify the way that recommendation partners are generally selected or weighted during the recommendation process. Specifically, that in addition to profile-profile similarity—the standard basis for partner selection—we argue that the *trustworthiness* of a partner should also be considered. A recommendation partner may have similar ratings to a target user but they may not be a reliable predictor for a given item or set of items. For example, when looking for movie recommendations we will often turn to our friends, on the basis that we have similar movie preferences overall. However, a particular friend may not be reliable when it comes to recommending a particular type of movie. The point is, that partner similarity alone is not ideal. Our recommendation partners should have similar tastes and preferences and they should be trustworthy in the sense that they have a history of making reliable recommendations. We propose a number of computational models of trust based on the past rating

behaviour of individual profiles. These models operate at the profile-level (average trust for the profile overall) and at the profile-item-level (average trust for a particular profile when it comes to making recommendations for a specific item). We describe how this trust information can be incorporated into the recommendation process and demonstrate that it has a positive impact on recommendation quality.

3. BACKGROUND

The semantic web, social networking, virtual communities and, of course recommender systems: these are all examples of research areas where the issue of trust, reputation and reliability is becoming increasingly important, especially as we see work progress from the comfort of the research lab to the hostile real-world.

3.1 Defining Trust

Across most current research, definitions of trust fall into various categories, and a solid definition for it, in many cases can be quite elusive. Marsh's work in [10] goes some way towards formalising trust in a computational sense, taking into account both it's social and technological aspects. More specifically, work in [1] illustrates categories which trust falls into, two of which concern this work: *Context-specific interpersonal trust,* which we are most interested in, is a situation where a user has to trust another user with respect to one specific situation, but not necessarily to another. The second category is *system / impersonal trust* which describes a users trust in a system as a whole. The issue of trust has been gaining an increasing amount of attention in a number of research communities and of course there are many different views of how to measure and use trust.

3.2 Trust & Reputation Modeling on the Semantic Web

For example, on the semantic web, trust and reputation can be expressed using domain knowledge and ontologies that provide a method for modelling the trust relationships that exist between entities and the content of information sources.

Recent research in [6] describes an algorithm for generating locally calculated reputation ratings from a semantic web social network. This work describes TrustMail, an email rating application. Trust scores in this system are calculated through inference and propagation, of the form $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow C)$, where A, B and C are users with interpersonal trust scores. The TrustMail application [6] looks up an email sender in the reputation / trust network, and provides an inline rating for each mail. These trust values can tell a user if a mail is important or unimportant. Trust values in this system can be defined with respect to a certain topic, or on a general level, as discussed in sections 2 and 3 of this paper. A big limitation of the work in [6] and [12] is that they require some explicit trust ratings in order to infer further trust rating. Experimental evidence is presented in [6] which shows that bad nodes in trust propagation network cause rating accuracy to drop drastically.

3.3 Trust-Based Filtering & Recommendation

More directly related to the work in this paper are a number of recent research efforts that focus on the use of trust and reputation models during the recommendation process.

In our research we are interested in automatically inferring trust relationships from ratings based data, and using these relationships to influence the recommendation process. Recently a number of researchers have tackled a related issue by using more directly available trust relationships.

For example, the work of [12] builds a trust model directly from trust data provided by users as part of the popular *epinions.com* service. *Epinions.com* is a web site that allows users to review various items (cars, books, music, etc.). In addition they can assign a trust rating to reviewers based on the degree to which they have found them to be helpful and reliable in the past. [12] argue that this trust data can be extracted and used as part of the recommendation process, especially as a means to relieve the sparsity problem that has hampered traditional collaborative filtering techniques. The sparsity problem refers to the fact that on average two users are unlikely to have rated many of the same items, which means that it will be difficult to calculate their degree of similarity and so limits the range of recommendation partners that can participate in a typical recommendation session. [12] argue that it is possible to compare users according to their degree of connectedness in the trust-graph encoded by *Epinions.com*. The basic idea is to measure the distance between two users in terms of the number of arcs connecting the users in the trust-graph encoded by the *Epinions.com* trust data. They show that it is possible to compare far more users according to this method than by conventional forms of ratings similarity and argue that because of this trust-based comparisons facilitate the identification of more comprehensive communities of recommendation partners. However, it must be pointed out that while the research data presented does demonstrate that the trust data makes it possible to compare far more users to each other it has not been shown that this method of comparison maintains recommendation accuracy.

Similar work on the *Epinions.com* data in [11] introduces a trust-aware recommendation architecture which again relies on a web of trust for defining a value for how much a user can trust every other user in the system. This system is successful in lowering the mean error on predictive accuracy for cold start users, ie: user who have not rated sufficiently many items for the standard CF techniques to generate accurate predictions. Trust data is used to increase the overlap between user profiles in the system, and therefore the number of comparable users. Work in [11] also defines a trade-off situation between recommendation coverage and accuracy in the system. This work however lacks an empirical comparison between a standard CF technique, such as Resnick's user based algorithm, and the trust-based technique. Future work in [11] mentions utilising both *local* and *global* trust metrics, which we discuss later in this paper in the form of Item and Profile level trust. The same research group are involved in *Moleskiing*, [2], a trust-aware decenteralised ski recommender, which uses trust propagation in a similar manner.

The work of [14] contemplates the availability of large numbers of virtual recommendation agents as part of a distributed agent-based recommender paradigm. Their main innovation is to consider other agents as personal entities which are more or less reliable or trustworthy and, crucially, that trust values can be computed by pairs of agents on the basis of a conversational exchange in which one agent solicits the opinions of an other with respect to a set of items.

Each agent can then infer a trust value based on the similarity between its own opinions and the opinions of the other. Thus this more emphasises a degree of proactiveness in that agents actively seek out others in order to build their trust model which is then used in the *opinion-based* recommendation model. This approach is advantageous from a hybrid recommender perspective, in that agents can represent individual techniques and they can be combined using opinions based on trust.

4. COMPUTATIONAL MODELS OF TRUST

We distinguish between two types of profiles in the context of a given recommendation session or rating prediction. The *consumer* refers to the profile receiving the item rating, whereas the *producer* refers to the profile that has been selected as a recommendation partner for the consumer and that is participating in the recommendation session. So, to generate a predicted rating for item i for some consumer c, we will typically draw on the services of a number of producer profiles, combining their individual recommendations according to some suitable function, such as Resnick's formula, for example. (see Equation 1)

To date collaborative filtering systems have relied heavily on what might be termed the *similarity assumption*: that similar profiles (similar in terms of their ratings histories) make good recommendation partners.

Our benchmark algorithm uses Resnick's standard prediction formula which is reproduced below as Equation 1; see also [20]. In this formula $c(i)$ is the rating to be predicted for item i in consumer profile c and $p(i)$ is the rating for item i by a producer profile p who has rated i. In addition, $\bar{c}$ and $\bar{p}$ refers to the mean ratings for c and p respectively. The weighting factor $sim(c,p)$ is a measure of the *similarity* between profiles c and p, which is traditionally calculated as Pearson's correlation coefficient.

$$c(i) = \bar{c} + \frac{\sum\limits_{p \epsilon P(i)} (p(i) - \bar{p}) sim(c,p)}{\sum\limits_{p \epsilon P_i} |sim(c,p)|} \qquad (1)$$

We use this benchmark as it allows for ease of comparison with existing systems.

As we have seen above Resnick's prediction formula discounts the contribution of a partner's prediction according to its degree of similarity with the target user so that more similar partners have a large impact on the final ratings prediction.

We propose, however, that profile similarity is just one of a number of possible factors that might be used to influence recommendation and prediction. We believe that the reliability of a partner profile to deliver accurate recommendations in the past is another important factor, one that we refer to as the *trust*. Intuitively, if a profile has made lots of accurate recommendation predictions in the past they can be viewed as more trustworthy that another profile that has made many poor predictions. In this section we define two models of trust and show how they can be readily incorporated into the mechanics of a standard collaborative filtering recommender system.
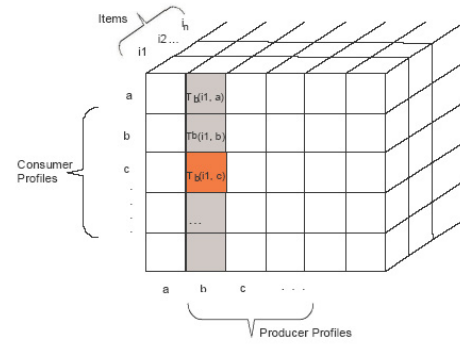

Figure 1: Calculation of Trust Scores from Rating Data

4.1 Profile-Level & Item-Level Trust

We say that a ratings prediction for an item, i, by a producer p for a consumer c, is *correct* if the predicted rating, $p(i)$, is within ϵ of c's actual rating $c(i)$; see Equation 2. Of course normally when a producer is involved in the recommendation process they are participating with a number of other recommendation partners and it may not be possible to judge whether the final recommendation is correct of a result of p's contribution. Accordingly, when calculating the correctness of p's recommendation we separately perform the recommendation process by using p as c's sole recommendation partner. For example, in Figure 1, a trust score for item *i1* is generated for producer b by using the information in profile b *only* to generate predictions for each consumer profile. Equation 3 shows how each box in Figure 1 translates to a binary success/fail score depending on whether or not the generated rating is within a distance of ϵ from the actual rating a particular consumer has for that item. In a real-time recommender system, trust values for producers could be easily created on the fly, by a comparison between our predicted rating (based only on one producer profile) and the actual rating which a user enters.

$$Correct(i,p,c) \Leftrightarrow |p(i) - c(i)| < \epsilon \qquad (2)$$

$$T_p(i,c) = Correct(i,p,c) \qquad (3)$$

From this we can define two basic trust metrics based on the relative number of correct recommendations that a given producer has made. The full set of recommendations that a given producer has been involved in, $RecSet(p)$, is given by Equation 4. And the subset of these that are correct, $CorrectSet(p)$ is given by Equation 5. The i values represent items and the c values are predicted ratings.

$$RecSet(p) = \{(c_1, i_1), ..., (c_n, i_n)\} \qquad (4)$$

$$CorrectSet(p) = \{(c_k, i_k) \in RecSet(p) : Correct(i_k, p, c_k)\} \qquad (5)$$

The *profile-level trust*, $Trust^P$ for a producer is the percentage of correct recommendations that this producer has contributed; see Equation 5. For example, if a producer has been involved in 100 recommendations, that is they have

served as a recommendation partner 100 times, and for 40 of these recommendations the producer was capable of predicting a correct rating, the profile level trust score for this user is 0.4.

$$Trust^P(p) = \frac{|CorrectSet(p)|}{|RecSet(p)|} \qquad (6)$$

Obviously, profile-level trust is very coarse grained measure of trust as it applies to the profile as a whole. In reality, we might expect that a given producer profile may be more trustworthy when it comes to predicting ratings for certain items than for others. Accordingly we can define a more fine-grained item-level trust metric, $Trust^I$, as shown in Equation 3, which measures the percentage of recommendations for an item i that were correct.

$$Trust^I(p,i) = \frac{|\{(c_k, i_k) \in CorrectSet(p) : i_k = i\}|}{|\{(c_k, i_k) \in RecSet(p) : i_k = i\}|} \qquad (7)$$

4.2 Trust-Based Recommendation

Now that we can estimate the trust of a profile (or a profile with respect to a specific item) we must consider how to incorporate trust into the recommendation process. The simplest approach is to adopt Resnick's prediction strategy (see Equation 1). We will consider 2 adaptations: *trust-based weighting* and *trust-based filtering*, both of which can be used with either profile-level or item-level trust metrics. We note that there are many ways to incorporate trust values into the recommendation process. We uses Resnick's formula since it is the most widely used.

4.2.1 Trust-Based Weighting

Perhaps the simplest way to incorporate trust in to the recommendation process is to combine trust and similarity to produce a compound weighting that can be used by Resnick's formula; see Equation 8.

$$c(i) = \bar{c} + \frac{\sum_{p \epsilon P(i)} (p(i) - \bar{p})w(c,p,i)}{\sum_{p \epsilon P(i)} |w(c,p,i)|} \qquad (8)$$

$$w(c,p,i) = \frac{2(sim(c,p))(trust^I(p,i))}{sim(c,p) + trust^I(p,i)} \qquad (9)$$

For example, when predicting the rating for item i for consumer c we could compute the arithmetic mean of the trust value (profile-level or item-level) and the similarity value for each producer profile. We have chosen a modification on this by using the harmonic mean of trust and similarity; see Equation 9 which combines profile similarity with item-level trust in this case. The advantage of using the harmonic mean is that it is robust to large differences between the inputs so that a high weighting will only be calculated if both trust and similarity scores are high. We chose a harmonic mean method over addition, subtraction and multiplication techniques as it performed best in our preliminary optimisation tests.

4.2.2 Trust-Based Filtering

As an alternative to the trust-based weighting scheme above we can use trust as a means of filtering profiles prior to

recommendation so that only the most trustworthy profiles participate in the prediction process. For example, Equation 10 shows a modified version of Resnick's formula which only allows producer profiles to participate in the recommendation process if their trust values exceed some predefined threshold; see Equation 11 which uses item-level trust ($Trust^I(p,i)$) but can be easily adapted to use profile-level trust. The standard Resnick method is thus only applied to the most trustworthy profiles.

$$c(i) = \bar{c} + \frac{\sum_{p \epsilon P^T(i)} (p(i) - \bar{p})sim(c,p)}{\sum_{p \epsilon P^T(i)} |sim(c,p)|} \qquad (10)$$

$$P_i^T = \{p \epsilon P(i) : Trust^I(p,i) > T\} \qquad (11)$$

4.2.3 Combining Trust-Based Weighting & Filtering

Of course, it is obviously straightforward to combine both of these schemes so that profiles are first filtered according to their trust values and the trust values of these highly trustworthy profiles are combined with profile similarity during prediction. For instance, Equation 12 shows both approaches used in combination using item-level trust.

$$c(i) = \bar{c} + \frac{\sum_{p \epsilon P^T(i)} (p(i) - \bar{p})w(c,p,i)}{\sum_{p \epsilon P^T(i)} |w(c,p,i)|} \qquad (12)$$

5. EVALUATION

So far we have argued that profile similarity alone may not be enough to guarantee high quality predictions and recommendations in collaborative filtering systems. We have highlighted trust as an additional factor to consider in weighting the relative contributions of profiles during ratings prediction. In our discussion section we consider the all important practical benefits of incorporating models of trust into the recommendation process. Specifically, we describe a set of experiments conducted to better understand how trust might improve recommendation accuracy and prediction error relative to more traditional collaborative filtering approaches.

5.1 Setup

In this experiment we use the standard MovieLens dataset [20] which contains 943 profiles of movie ratings. Profile sizes vary from 18 to 706 with an average size of 105. We divide these profiles into two groups: 80% are used as the producer profiles and the remaining 20% are used as the consumer (test) profiles. For all of our evaluation experiments, training and test profiles are independent profile sets.

Before evaluating the accuracy of our new trust-based prediction techniques we must first build up the trust values for the producer profiles as described in the next section. It is worth noting that ordinarily these trust values would be built on-the-fly during the normal operation of the recommender system, but for the purpose of this experiment we have chosen to construct them separately, but without reference to the test profiles. Having built the trust values we

evaluate the effectiveness of our new techniques by generating rating predictions for each item in each consumer profile by using the producer profiles as recommendation partners. We do this using the following different recommendation strategies:

1. *Std* - The standard Resnick prediction method.

2. *WProfile* - Trust-based weighting using profile-level trust.

3. *WItem* - Trust-based weighting using item-level trust.

4. *FProfile* - Trust-based filtering using profile-level trust and with the mean profile-level trust across the producers used as a threshold..

5. *FItem* - Trust-based filtering using item-level trust and with the mean item-level trust value across the profiles used as a threshold.

6. *CProfile* - Combined trust-based filtering & weighting using profile-level trust.

7. *CItem* - Combined trust-based filtering & weighting using item-level trust.

5.2 Building Trust

Ordinarily our proposed trust-based recommendation strategies contemplate the calculation of relevant trust-values on-the-fly as part of the normal recommendation process or during the training phase for new users. However, for the purpose of this study we must calculate the trust values in advance. We do this by running a standard *leave-one-out* training session over the producer profiles. In short, each producer temporarily serves as a consumer profile and we generate rating predictions for each of its items by using Resnick's prediction formula with each remaining producer as a lone recommendation partner; that is, each producer is used in isolation to make a prediction. By comparing the predicted rating to the known actual rating we can determine whether or not a given producer has made a correct recommendation — in the sense that the predicted rating is within a set threshold of the actual rating – and so build up the profile-level and item-level trust scores across the producer profiles.

This approach is used to build both profile-level trust values and item-level trust values. To get a sense of the type of trust values generated we present histograms of the profile-level and item-level values for the producer profiles in Figures 2 and 3. In each case we find that the trust values are normally distributed but they differ in the degree of variation that is evident. Not surprisingly there is greater variability in the more numerous item-level trust values, which extend from as low as 0.5 to as high as 1. This variation is lost in the averaging process that is used to build the profile-level trust values from these item-level data. Most of the profile-level trust values range from about 0.3 to about 0.8. For example, in Figure 3 approximately 13% of profiles have trust values less that 0.4 and 25% of profiles have trust values greater than 0.7. By comparison less than 4% of the profile-level trust values are less than 0.4 and less than 6% are greater than 0.7. The error parameter ϵ from equation 2 was set to be 1.8 for our tests as this gave a good distribution of trust values.
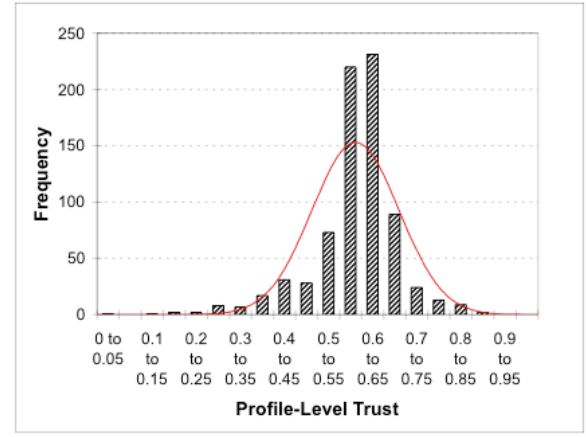

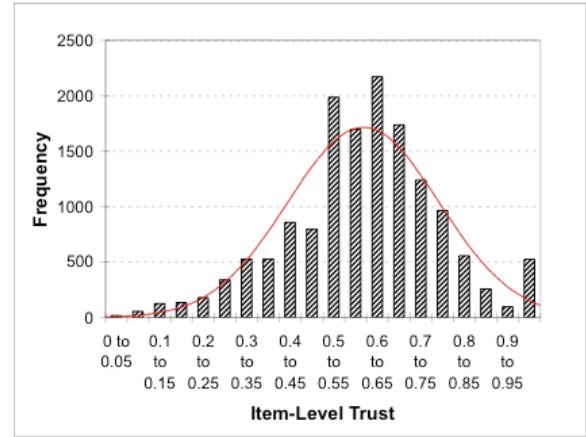**Figure 2: The distribution of profile-level trust values among the producer profiles.**

Figure 3: The distribution of item-level trust values among the producer profiles.

If there was little variation in trust then we would not expect of trust-based prediction strategies to differ significantly from standard Resnick, but of course since there is much variation, especially in the item-level values, then we do expect significant differences between the predictions made by Resnick and the predictions made by our alternative strategies. Of course whether the trust-based predictions are demonstrably better remains to be seen.

5.3 Recommendation Error

Ultimately we are interested in exploring how the use of trust estimates can make recommendation and ratings predictions more reliable and accurate. In this experiment we focus on the mean recommendation error generated by each of the recommendation strategies over the items contained within the consumer profiles. That is, for each consumer profile, we temporarily remove each of its rated items and use the producer profiles to generate a predicted rating for this target item according to one of the 7 recommendation strategies proposed above. The rating error is calculated with reference to the item's known rating and an average error is calculated for each strategy.

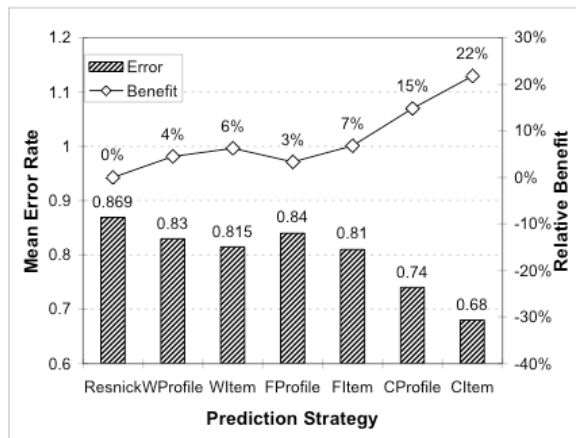The results are presented in Figure 4 as a bar-chart of

Figure 4: The average prediction error and relative benefit (compared to Resnick) of each of the trust-based recommendation strategies.
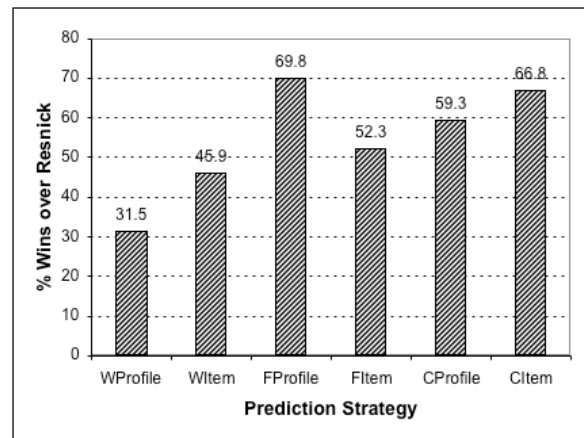

Figure 5: The percentages of predictions where each of the trust-based techniques achieves a lower error prediction than the benchmark Resnick technique.

average error values for each of the 7 strategies. In addition, the line graph represents the relative error reduction enjoyed by each strategy, compared to the Resnick benchmark. A number of patterns emerge with respect to the errors. Firstly, the trust-based methods all produce lower errors than the Resnick approach (and all of these reductions are statistically significant at the 95% confidence level) with the best performer being the combined item-level trust approach ($CItem$) with an average error of 0.68, a 22% reduction in the Resnick error.

We also find that in general the item-level trust approaches perform better than the profile-level approaches. For example, $WItem$, $FItem$ and $CItem$ all outperform their corresponding profile-level strategies ($WProfile$, $FProfile$ and $CProfile$). This is to be expected as the item-level trust values provide a far more fine-grained and accurate account of the reliability of a profile during recommendation and prediction. An individual profile may be very trustworthy when it comes to predicting the ratings of some of its items, but less so for others. This distinction is lost in the averaging process that is used to derive single profile-level trust values, which explains the difference in rating errors.

In addition, the combined strategies significantly outperform their corresponding weighting and filtering strategies. Neither the filtering or weighting strategies on their own are sufficient to deliver the major benefits of the combination strategies. But together the combination of filtering out untrustworthy profiles and the use of trust values during the ratings prediction results in a significant reduction in error. For example, the combined item-level strategy achieves a further 16% error reduction compared to the weighted or filter-based item-level strategies, and the combined profile-level strategy achieves a further 11% error reduction compared to the weighted or filter-based profile-level approaches.

5.4 Winners & Losers

So far we have demonstrated that on average, over a large number of predictions, the trust-based predictions techniques achieve a lower overall error than Resnick. It is not clear, however, whether these lower errors arise out of a general improvement by the trust-based techniques over the majority of individual predictions, when compared to Resnick, or

whether they arise because of a small number of very low error predictions that serve to mask less impressive performance at the level of individual predictions. To test this, in this section we look at the percentage of predictions where each of the trust-based methods wins over Resnick, in the sense that they achieve lower error predictions on a prediction by prediction basis.

These results are presented in Figure 5 and they are revealing in a number of respects. For a start, even though the two weighting-based strategies ($WProfile$ & $WItem$) deliver an improved prediction error than Resnick, albeit a marginal improvement, they only win in 31.5% and 45.9% of the prediction trials, respectively. In other words, Resnick delivers a better prediction the majority of times. The filter-based ($FProfile$ & $FItem$) and combination strategies ($CProfile$ & $CItem$) offer much better performance. All of these strategies win on the majority of trials with $FProfile$ and $CItem$ winning in 70% and 67% of predictions, respectively.

Interestingly, the $FProfile$ strategy offers the best overall improvement in terms of its percentage wins over Resnick, even though on average it offers only a 3% mean error reduction compared to Resnick. So even though $FProfile$ delivers a lower error prediction than Resnick nearly 70% of the time, these improvements are relatively minor. In contrast, the $CItem$, which beats Resnick 67% of the time, does so on the basis of a much more impressive overall error reduction of 22%.

6. DISCUSSION

6.1 Trust, Reliability or Competence?

It has been pointed out that *competence* may be a more suitable term than trust for the title of this work. We feel however that there several ways of comprehending this term in the context of recommenders. Competence (and also trust) may imply the overall ability of the system to provide consistently good recommendations to its users. Our intended meaning is more specific than this, in that we are defining the goodness of a users contribution to the computation of recommendations. A proper distinction must be

defined between this metric, and the overall *trust* that a user places in the system. As an alternative *reputation* may be used in the place of trust for the title of this work.

6.2 Acquiring Real-World Feedback

As with all recommenders, in practice our trust based system relies on the fact that users will provide ratings on the items the system recommends. Currently our system works on an experimental dataset divided into training and test sets, where trust values are built on the 80% test set. This setup is for evaluation purposes only, and we are developing a realtime recommender system upon which we can demonstrate the manner in which we dynamically generate trust values as users provide ratings. Feedback of some sort must occur in all recommenders, for example, the Grouplens news recommender [20], uses the amount of time a user spends reading an article as a non-invasive method of acquiring feedback. We are looking at similar ways to elicit feedback from the interactions a user has with the system. In *PTV* [4] [19], feedback is taken explicitly from users in the form of rating recommendations. The *físchlár* video recommender system [5], [22] elicits implicit feedback by checking if recommended items are recorded or played. On *Amazon.com* a fundamental *purchased or not* technique is used to compute user satisfaction with recommendations. There is generally a trade off between non-invasive and high quality methods of acquiring user feedback, for example in the Grouplens system the read-time measurement is flawed and misleading if the user leaves the pc, or is simply not reading what is on the screen. These problems are more thoroughly discussed in [15]

6.3 Trust & CF Robustness

The trust models defined in this paper can not only be used to increase recommendation accuracy in a recommender, they can be utilised to increase the overall robustness of CF systems. The work of O'Mahony et. al [18, 17] and Leiven [9] outlines recommender systems from the viewpoints of *accuracy*, *efficiency*, and *stability*. [18] defines several attack strategies that can adversely skew the recommendations generated by a K-NN CF system. They show empirically in [17] that a CF system needs to attend to each of these factors in order to succeed well. There are many motivations for users attempting to mislead recommender systems, including profit, and malice, as outlined in [8].

We propose that our item-level and, probably more importantly profile-level trust values will enable a system to automatically detect malicious users, since they have provided consistently bad recommendations, so that by employing our trust weighting mechanism, see Equation 9, we can render their contribution to future recommendations ineffective. In a future paper we will explore avenues of trust-aided CF robustness, including techniques to recognise when a malicious user provides 'liked' ratings to a consumer. Further work on robustness in CF systems was carried out by Kushmerick in [7]

6.4 Trust & Recommendation Explanation

Ongoing work by Sinah and Swearingen outlines the importance of transparency in recommender system interfaces, [21]. In a study they performed on 5 music recommender systems, they have shown that both mean *liking* and mean *confidence* are greatly increased in a system that is more transparent. It stems from psychology that people are generally more comfortable with what they are familiar with and understand, and the *black box* [21] approach of most recommender systems seems to completely ignore this important rule.

Our trust models can be used as part of a more broad *recommendation explanation*. By using trust scores we are able to say to a user (in, for example, a car recommender) "You have been recommended a Toyota Carina; This recommendation has been generated by users A, B and C, and these users have successfully recommended Toyota Carinas X, Y and Z times in the past, and furthermore, P, Q, and R% respectively of their overall recommendations have been successful in the past." We believe that this recommendation accountability is a very influential factor in increasing the faith a user places in the recommendation.

7. CONCLUSIONS

Traditionally collaborative filtering systems have relied heavily on similarities between the ratings profiles of users as a way to differentially rate the prediction contributions of different profiles. In this paper we have argued that profile similarity on its own may not be sufficient, that other factors might also have an important role to play. Specifically we have introduced the notion of trust in reference to the degree to which one might trust a specific profile when it comes to making a specific rating prediction.

We have developed two different trust models, one that operates at the level of the profile and one at the level of the items within a profile. In both of these models trust is estimated by monitoring the accuracy of a profile at making predictions over an extended period of time. Trust then is the percentage of correct predictions that a profile has made in general (profile-level trust) or with respect to a particular item (item-level trust). We have described a number of ways in which these different types of trust values might be incorporated into a standard collaborative filtering algorithm and evaluated each against a tried-and-test benchmark approach and on a standard data-set. In each case we have found the use of trust values to have a positive impact on overall prediction error rates with the best performing strategy reducing the average prediction error by 22% compared to the benchmark.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] Alfarez Abdul-Rahman and Stephen Hailes. A distributed trust model. In *New Security Paradigms 1997*, pages 48–60, 1997.

[2] Paolo Avesani, Paolo Massa, and Roberto Tiella. Moleskiing: a trust-aware decentralized recommender system. *1st Workshop on Friend of a Friend, Social Networking and the Semantic Web. Galway, Ireland*, 2004.

[3] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Gregory F. Cooper and Serafín Moral, editors, *Proceedings of the 14th Conference on Uncer-*

tainty in *Artificial Intelligence (UAI-98)*, pages 43–52, San Francisco, July 24–26 1998. Morgan Kaufmann.

[4] Paul Cotter and Barry Smyth. Ptv: Intelligent personalised tv guides. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 957–964. AAAI Press / The MIT Press, 2000.

[5] Alan Smeaton et al. The físchlár digital video system: a digital library of broadcast tv programmes. *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries.*

[6] Jennifer Golbeck and James Hendler. Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In *Proceedings of EKAW'04*, pages LNAI 2416, p. 278 ff., 2004.

[7] N. Kushmerick. Robustness analyses of instance-based collaborative recommendation. In *Proceedings of the European Conference on Machine Learning, Helsinki, Finland.*, volume 2430, pages 232–244. Lecture Notes in Computer Science Springer-Verlag Heidelberg, 2002.

[8] Shyong K. Lam and John Riedl. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*, pages 393–402. ACM Press, 2004.

[9] Raph Levien. Attack resistant trust metrics. *Ph.D Thesis, UC Berkeley.*

[10] S. Marsh. Formalising trust as a computational concept. *Ph.D. Thesis. Department of Mathematics and Computer Science, University of Stirling,1994.*

[11] Paolo Massa and Paolo Avesani. Trust-aware collaborative filtering for recommender systems. *To Appear in: Proceedings of International Conference on Cooperative Information Systems, Agia Napa, Cyprus, 25 Oct - 29 Oct 2004.*

[12] Paolo Massa and Bobby Bhattacharjee. Using trust in recommender systems: an experimental analysis. *Proceedings of 2nd International Conference on Trust Managment, Oxford, England*, 2004.

[13] P. Melville, R. Mooney, and R. Nagarajan. Content-boosted collaborative filtering. In *P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering. In Proceedings of the 2001.*, 2001.

[14] Miquel Montaner, Beatriz Lopez, and Josep Lluis de la Rosa. Developing trust in recommender agents. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 304–305. ACM Press, 2002.

[15] D. Oard and J. Kim. Implicit feedback for recommender systems, 1998.

[16] John O'Donovan and John Dunnion. A framework for evaluation of collaborative recommendation algorithms in an adaptive recommender system. In *Proceedings of the International Conference on Computational Linguistics (CICLing-04), Seoul, Korea*, pages 502–506. Springer-Verlag, 2004.

[17] Michael O'Mahony, Neil Hurley, Nicholas Kushmerick, and Guenole Silvestre. Collaborative recommendation: A robustness analysis, url: citeseer.ist.psu.edu/508439.html.

[18] Michael P. O'Mahony, Neil Hurley, and Guenole C. M. Silvestre. An attack on collaborative filtering. In *Proceedings of the 13th International Conference on Database and Expert Systems Applications*, pages 494–503. Springer-Verlag, 2002.

[19] Derry O'Sullivan, David C. Wilson, and Barry Smyth. Improving case-based recommendation: A collaborative filtering approach. In *Proceedings of the Sixth European Conference on Case Based Reasoning.*, pages LNAI 2416, p. 278 ff., 2002.

[20] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM CSCW'94 Conference on Computer-Supported Cooperative Work*, Sharing Information and Creating Meaning, pages 175–186, 1994.

[21] Rashmi Sinha and Kirsten Swearingen. The role of transparency in recommender systems. In *CHI '02 extended abstracts on Human factors in computing systems*, pages 830–831. ACM Press, 2002.

[22] B. Smyth, D. Wilson, and D. O'Sullivan. Improving the quality of the personalised electronic programme guide. In *In Proceedings of the TV'02 the 2nd Workshop on Personalisation in Future TV, May 2002.*, pages 42–55, 2002.