Personalizing Applications through Integration of Inferred Trust Values in Semantic Web-Based Social Networks

Jennifer Golbeck
Department of Computer Science
University of Maryland, College Park
College Park, Maryland, USA
golbeck@cs.umd.edu

Abstract. Social Network data, represented using the FOAF Vocabulary, is some of the most prevalent data on the Semantic Web. In this work, we look particularly at trust relationships in web-based social networks and their implications for software personalization. We present a network analysis as the foundation for TidalTrust, and algorithm for inferring trust relationships, and then illustrate how the results can be used to improve application interfaces.

1 Introduction and Background

The Friend Of A Friend (FOAF) project is one of the most popular efforts on the Semantic Web. The vocabulary for describing people and their social network connections is already used to represent information about over 8,000,000 people, and is a form of output being used by many large web-based social networks. This huge source of distributed data offers opportunities for performing social network analysis on real, evolving networks, and the web-based nature means that the publicly accessible data can be computed against and integrated into applications to help benefit the user.

In addition to the core vocabulary, FOAF has been extended in several ways to enhance the information about interpersonal relationships. For example, the FOAF Relationship Module[1] offers dozens of relationship types, such as "sibling of", "would like to know", and "spouse of". This work utilizes the FOAF Trust Module[2] which allows users to indicate how much they trust people they know.

With a social network where users have indicated how much they trust others, it is possible to recommend (or infer) how much one user might trust an unknown person by using the trust values on the paths that connect them. By inferring the trustworthiness of an unknown person, the quality of information from that person can be judged. In this paper, we present an algorithm, Tidal Trust, for inferring trust relationships in Semantic Web-based social networks. We describe the social network analysis that leads to these algorithms, and describe their accuracy within two networks. The benefit of this analysis is that the results can be integrated into applications to enhance the users' experiences by acting with respect to their social preferences. We present two applications – FilmTrust and TrustMail – that utilize the Tidal Trust algorithm, and show how the trust information leads to usability enhancements.

[1] http://www.perceive.net/schemas/20021119/relationship/

1.1 Experimental Networks

One facet of web-based social networks that makes them more complex and interesting that networks traditionally studied with social network analysis is the plethora of information about relationships that is available. In this work, we were particularly interested in trust because of the potential that information has for improving applications. Recent work provides two indications that users will prefer the sort of system that relies on trust in social networks. First, users tend to prefer recommendations from people they know and trust [3]. Related work also showed that users prefer recommendations from systems that they trust and understand [4]. Ziegler and Lausen [5] also showed a correlation between trust and user similarity in an empirical study of a real online community, indicating that trusted people may direct users to data relevant to their interests.

To perform the analysis, it was necessary to have data sources. Two separate trust networks have been grown from scratch. The first network is part of the Trust Project at http://trust.mindswap.org/. This network is built up from distributed data maintained on the Semantic Web. Within their FOAF files, users include trust ratings for people they know using the FOAF Trust Module, a simple ontology for expressing trust developed as part of this project. The ontology has vocabulary for rating people on a scale of 1 (low trust) to 10 (high trust). These ratings can be made in general or with respect to a specific topic. In the network built up for study in this research, users assigned general ratings to one another. There are approximately 2,000 people in this network with over 2,500 connections. Figure 1 shows the current structure of this network.

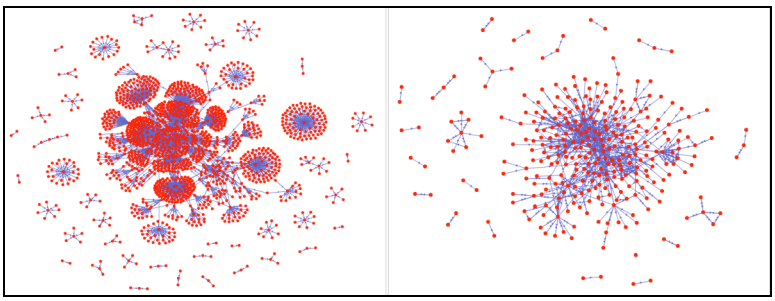

Fig 1. The structure of the social network from the Trust Project (left) and FilmTrust (right).

The second network is part of the FilmTrust project, a website that combines social networks with a movie ratings and reviews site. The site currently comprises 500 members who have rated each others' trustworthiness on the same 1-10 scale. In this network, users rate how much they trust people about movies.

1.2 Related Work

The issue of sharing trust assessments on the semantic web has been addressed in contexts outside of explicit social networks. Gil and Ratnakar addressed the issue of trusting content and information sources [6] on the Semantic Web. Their TRELLIS system derives assessments about information sources based on individual

feedback about the sources. Our work uses this notion of augmenting data on the Semantic Web (social network data in our case) with annotations about its trustworthiness.

Once a trust network has been properly modeled and represented, our attention moves to algorithms for calculating recommendations about trust in the network. The question of trust calculations in social networks has been addressed in several communities with a range of endpoint applications.

The EigenTrust algorithm [7] is used in peer-to-peer systems and calculates trust with a variation on the PageRank algorithm [8], used by Google for rating the relevance of web pages to a search. EigenTrust is designed for a peer-to-peer system while ours is designed for use in humans' social networks, and thus there are differences in the approaches to analyzing trust. In the EigenTrust formulation, trust is a measure of performance, and one would not expect a single peer's performance to differ much from one peer to another. Socially, though, two individuals can have dramatically different opinions about the trustworthiness of the same person. Our algorithms intentionally avoid using a global trust value for each individual to preserve the personal aspects that are foundations of social trust.

Raph Levin's Advogato project [9] also calculates a global reputation for individuals in the network, but from the perspective of designated *seeds* (authoritative nodes). His metric composes certifications between members to determine the trust level of a person, and thus their membership within a group. While the perspective used for making trust calculations is still global in the Advogato algorithm, it is much closer to the methods used in this research. Instead of using a set of global seeds, we let any individual be the starting point for calculations, so each calculated trust rating is given with respect to that person's view of the network.

Richardson et. al.[10] use social networks with trust to calculate the belief a user may have in a statement. This is done by finding paths (either through enumeration or probabilistic methods) from the source to any node which represents an opinion of the statement in question, concatenating trust values along the paths to come up with the recommended belief in the statement for that path, and aggregating those values to come up with a final trust value for the statement. Current social network systems on the Web, however, primarily focus on trust values between one user to another, and thus their aggregation function is not applicable in these systems.

2 Inferring Trust in Social Networks: TidalTrust

Inferring trust relationships within a social network requires an analysis of the properties of trust networks. We start with the assumption that people who are trusted highly will tend to agree with the user more about the trustworthiness of others than people who are less trusted. In this section, we present an analysis of the features of a trust network and their correlation to accuracy. Those results are then used in the development of an algorithm, TidalTrust, for inferring trust relationships

2.1 Correlation of Trust and Accuracy

To investigate the correlation of trust and accuracy, experiments were performed on the Trust Project network. The goal was to ascertain if neighbors with higher trust ratings were more likely to agree with the source about the trustworthiness of a third

person. This was determined repeating the following process for each node. First, a node was chosen as the source. For each neighbor of the source, n_i, a list of common neighbors of the source and n_i was compiled. For each of those common neighbors, the difference between the source's rating and n_i's rating was recorded as a measure of accuracy. A smaller difference means a higher accuracy. This difference was recorded along with the source's rating of n_i. Figure 2 illustrates one step in the process.

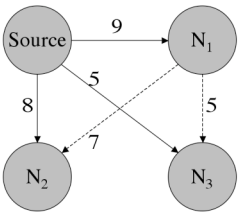

Fig. 2. Finding points of comparison in the network. In these experiments, this network would produce two data points: the difference between the source and N_1's ratings of N_2 (in this case, 1) and the difference between the source and N_1's ratings of N_3 (in this case, 0)

These experiments produced a pair of numbers for each data point: the trust value from the source to its neighbor (n_1), and the difference between the ratings of a common neighbor (Δ). The number of data points for each trust value indicates how frequently common neighbors are shared between pairs of nodes at each trust level.

The frequency of common neighbors among pairs of nodes with high trust levels is much higher than the frequency of those ratings in the original network. These indicate that people with stronger trust connections share more common social connections. In fact, over 40% of the common neighbors were found between nodes that shared a high trust rating. If the experimental results show that the results are more *accurate* when there is more trust, this distribution means that a the increased accuracy will be reinforced by the increased frequency of common neighbors among pairs with high trust.

If individuals with higher trust ratings agree with the source more, we would expect average difference (Δ) would decrease as trust ratings increase. In the datasets used, there were very few comparisons available for trust ratings 1-5. Because the number of comparisons for the lower trust ratings is so small, and the margin of error so large, these data points were not included in the analysis here. Instead, we focused on the comparisons made for trust values 6-10.

As shown in Figure 3, there appears to be a strong negative linear relationship between trust value and Δ. This is confirmed by the statistics; the Pearson's correlation is –0.991, indicating that there is an almost perfect negative linear relationship between the variables. These results are statistically significant for $p<.001$.

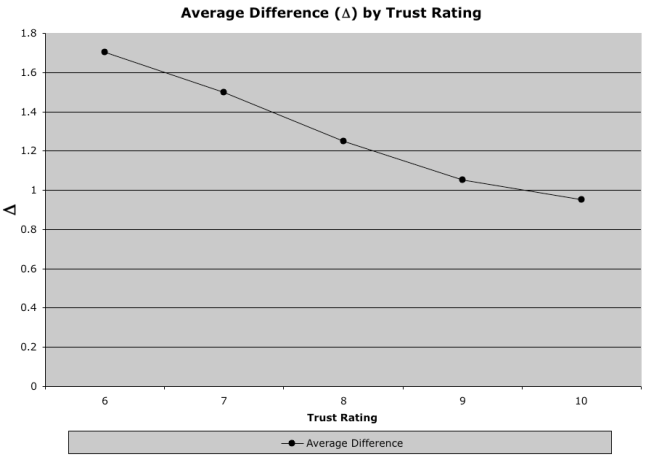

Fig 3. The relationship between Δ and Trust Rating.

These analyses show that in this trust network, there is more agreement between nodes connected by high trust ratings than nodes connected by lower trust ratings. Furthermore, common neighbors are found more frequently among pairs of nodes with higher trust ratings. Thus, the increased accuracy among highly trusted neighbors is amplified by the increased frequency of receiving data from those highly trusted neighbors. This leads to results that are more accurate overall. These elements will become a critical in the development of the trust inference algorithm.

2.2 Path Length and Accuracy

The length of a path is determined by the number of edges the source must traverse before reaching the sink. For example, source→n_1→sink has length two. Does the length of a path affect the agreement between individuals? Specifically, should the source expect that neighbors who are connected more closely will give more accurate information than people who are further away in the network? To study this relationship between path length and accuracy, we follow a similar approach used in section 2.1. The source is selected and for each source's neighbor n_i, we search for paths of length l to n_i. We compare the source's rating of n_i to the rating given to n_i along the path of length l.

Table I. Minimum average Δ for paths of various lengths containing the specified trust rating.

Trust Rating \ *Path Length*	2	3	4	5
10	0.953	1.52	1.92	2.44
9	1.054	1.588	1.969	2.51
8	1.251	1.698	2.048	2.52
7	1.5	1.958	2.287	2.79
6	1.702	2.076	2.369	2.92

Figure 4 illustrates the relationships from Table I.

For each path length, Table I shows the minimum average Δ. These are grouped according to the minimum trust value along that path.

In Figure 4, the effect of path length can be compared to the effects of trust ratings. For example, consider the average Δ for trust values of 7 on paths of length 2. This is approximately the same as the average Δ for trust values of 10 on paths of length 3 (both are close to 1.5). The average Δ for trust values of 7 on paths of length 3 is about the same as the average Δ for trust values of 9 on paths of length 4. A precise rule cannot be derived from these values because there is not a perfect linear relationship, and also because the points in Figure 4 are only the minimum average Δ among paths with the given trust rating.

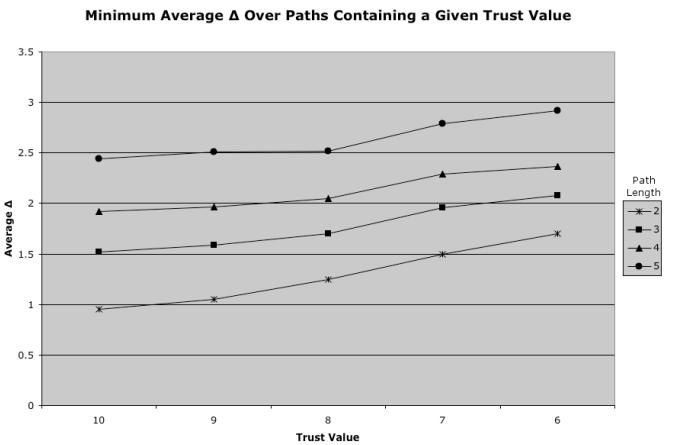

Fig 4. Minimum average Δ from all paths of a fixed length containing a given trust value.

This relationship will be integrated into the algorithms for inferring trust presented in the next section.

2.3 TidalTrust: An Algorithm for Inferring Trust

The in-depth look at the effects of trust ratings and path length in the previous section guided the development of TidalTrust, an algorithm for inferring trust in networks with continuous rating systems. The following guidelines can be extracted from the analysis of the previous sections:

1. For a fixed trust rating, shorter paths have a lower average Δ.
2. For a fixed path length, higher trust ratings have a lower average Δ.

This section describes how these features are used in the TidalTrust algorithm.

2.3.1 Incorporating Path Length The analysis in section 2.2 indicates that a limit on the depth of the search should lead to more accurate results, since the average Δ increases as depth increases. If accuracy decreases as path length increases, as the earlier analysis suggests, then shorter paths are more desirable. However, the tradeoff is that fewer nodes will be reachable if a limit is imposed on the path depth. To

balance these factors, the path length can vary from one computation to another. Instead of a fixed depth, the shortest path length required to connect the source to the sink becomes the depth. This preserves the benefits of a shorter path length without limiting the number of inferences that can be made.

2.3.2 Incorporating Continuous Trust The previous results also indicate that the most accurate information will come from the highest trusted neighbors. As such, we may want the algorithm to limit the information it receives so that it comes from only the most trusted neighbors, essentially giving no weight to the information from neighbors with low trust. If the algorithm were to take information only from neighbors with the highest trusted neighbor, each node would look at its neighbors, select those with the highest trust rating, and average their results. However, since different nodes will have different maximum values, some may restrict themselves to returning information only from neighbors rated 10, while others may have a maximum assigned value of 6 and be returning information from neighbors with that lower rating. Since this mixes in various levels of trust, it is not an ideal approach. On the other end of possibilities, the source may find the maximum value it has assigned, and limit every node to returning information only from nodes with that rating or higher. However, if the source has assigned a high maximum rating, it is often the case that there is no path with that high rating to the sink. The inferences that are made may be quite accurate, but the number of cases where no inference is made will increase. To address this problem, we define a variable *max* that represents the largest trust value that can be used as a minimum threshold such that a path can be found from source to sink.

$$t_{is} = \frac{\sum_{j \in adj(i) \ni t_{ij} >= \max} t_{ij} t_{js}}{\sum_{j \in adj(i) \ni t_{ij} >= \max} t_{ij}} \qquad (1)$$

2.3.3 Full Algorithm for Inferring Trust Incorporating the elements presented in the previous sections, the final TidalTrust algorithm can be assembled. The name was chosen because calculations sweep forward from source to sink in the network, and then pull back from the sink to return the final value to the source.

The source node begins a search for the sink. It will poll each of its neighbors to obtain their rating of the sink. Each neighbor repeats this process, keeping track of the current depth from the source. Each node will also keep track of the strength of the path to it. Nodes adjacent to the source will record the source's rating assigned to them. Each of those nodes will poll their neighbors. The strength of the path to each neighbor is the minimum of the source's rating of the node and the node's rating of its neighbor. The neighbor records the maximum strength path leading to it. Once a path is found from the source to the sink, the depth is set at the maximum depth allowable. Since the search is proceeding in a Breadth First Search fashion, the first path found will be at the minimum depth. The search will continue to find any other paths at the minimum depth. Once this search is complete, the trust threshold (the variable *max* in formula 1) is established by taking the maximum of the trust paths leading to the sink. This is illustrated in Figure 5 .

With the *max* value established, each node can complete the calculations of a weighted average by taking information from nodes that they have rated at or above the *max* threshold.

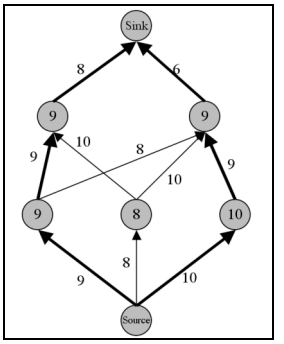

Fig 5. The process of determining the trust threshold. The label on each edge represents the trust rating between nodes. The label on each node indicates the maximum trust strength on the path leading to that node. The two nodes adjacent to the sink have values of 9, so 9 is the *max* value. The bold edges indicate which paths will ultimately be used in the calculation because they are at or above the *max* threshold.

2.4 Accuracy of TidalTrust

As presented above, TidalTrust strictly adheres to the observed characteristics of trust: shorter paths and higher trust values lead to better accuracy. However, there are some things that should be kept in mind. The most important is that networks are different. Depending on the subject (or lack thereof) about which trust is being expressed, the user community, and the design of the network, the effect of these properties of trust can vary. While we should still expect the general principles to be the same – shorter paths will be better than longer ones, and higher trusted people will agree with us more than less trusted people – the proportions of those relationships may differ from what was observed in the sample networks used in this research.

Table II. Average Δ for TidalTrust and Simple Average recommendations in both the Trust Project and FilmTrust networks. Numbers are absolute error on a 1-10 scale.

	Algorithm	
Network	**TidalTrust**	**Simple Average**
Trust Project	1.09	1.43
FilmTrust	1.35	1.93

There are several algorithms that output trust inferences, but none of them produce values within the same scale that users assign ratings. Some trust algorithms form the Public Key Infrastructure (PKI) are more appropriate for comparison. A comparison of this algorithm to PKI can be found in [2], but due to space limitations that comparison is not included here. One direct comparison to make is to compare the average Δ from TidalTrust to the average Δ from taking the simple average of all

ratings assigned to the sink as the recommendation. As shown in Table II, the TidalTrust recommendations outperform the simple average in both networks, and these results are statistically significant with $p<0.01$. Even with these preliminary promising results, TidalTrust is not designed to be the optimal trust inference algorithm for every network in the state it is presented here. Rather, the algorithm presented here adheres to the observed rules of trust. When implementing this algorithm on a network, modifications *should be made* to the conditions of the algorithm that adjust the maximum depth of the search, or the trust threshold at which nodes are no longer considered. How and when to make those adjustments will depend on the specific features of a given network. These tweaks will not affect the complexity of implementation.

3 Applying the Analysis: FilmTrust and TrustMail

In this section, we look at using trust values as recommender systems. Similar techniques for integrating trust and recommendations have appeared in recent work, including that by Massa et al [11], and Zeigler [5].

3.1 FilmTrust: Social Networks and Movie Ratings

FilmTrust is a website that utilizes trust ratings in a social network to make personalized predictive recommendations about movies to the user. Using the trust inferences from TidalTrust in an algorithm for generating ratings, we are able to show that in certain cases, the predictive ratings are far more accurate than more traditional methods.

The social networking component of the website requires users to provide a trust rating for each person they add as a friend. When creating a trust rating, users are advised to rate how much they trust their friend about movies. The other features of the website are movie ratings and reviews. Users can choose any film and rate it on a scale of a half star to four stars. They can also write free-text reviews about movies.

The social network is integrated with the movie ratings with the "Recommended Rating" feature. This is personalized using the trust values for the people who have rated the film (the *raters*). The process for calculating this rating is done with a weighted average, similar to the process for calculating trust ratings. Using the Tidal Trust algorithm, set of highly trusted nodes who have rated the given film are chosen. For the set of selected nodes S, the recommended rating r from node s to movie m is computed as the average of the movie ratings from nodes in S weighted by the trust value t from s to each node:

$$r_{sm} = \frac{\sum_{i \in S} t_{si} r_{im}}{\sum_{i \in S} t_{si}}$$

This average is rounded to the nearest half-star, and that value becomes the "Recommended Rating" that is personalized for each user.

As a simple example, consider the following:

- Alice trusts Bob 9
- Alice trusts Chuck 3
- Bob rates the movie "Jaws" with 4 stars

- Chuck rates the movie "Jaws" with 2 stars

Then Alice's recommended rating for "Jaws" is calculated as follows:

$$\frac{t_{Alice->Bob} * r_{Bob->Jaws} + t_{Alice->Chuck} r_{Chuck-Jaws}}{t_{Alice->Bob} + t_{Alice->Chuck}} = \frac{9*4+3*2}{9+3} = \frac{42}{12} = 3.5$$

Judging the accuracy of these ratings can also be done in a way similar to the analysis of the accuracy of the trust calculations. For each movie the user has rated, the recommended rating can be compared to the actual rating that the user assigned. For further comparison, we also compared the user's rating to the simple average rating of a movie (commonly shown on other movie websites), and to a recommended rating generated by a Pearson correlation-based automated collaborative filtering (ACF) algorithm [12].

On first analysis, it did not appear that that the personalized ratings offered any benefit over the average. The difference between the actual rating and the trust-based *recommended* rating (call this ∂r) was not statistically different than the difference between the actual rating and the *average* rating (call this ∂a). A close look at the data suggested why. Most of the time, the majority of users actual ratings are close to the average. Of course, it should be expected that there is a relatively normal distribution of ratings around the mean, and that a large percentage of ratings will fall close to that mean. A random sampling of movies showed that about 50% of all ratings were within +/- a half star of the mean. For these users, a personalized rating could not offer much benefit over the average. However, the point of the recommended rating is to perform well when the average does not, i.e. when the user's opinion is different from the average opinion or ∂a is high. In those cases, the personalized rating should give the user a better recommendation, because we expect the people they trust will have tastes similar to their own [5].

To test if this benefit is real, and experiment was conducted by computing the ∂ values with a minimum threshold on ∂a. The first set of comparisons was taken with no threshold, where the difference between ∂a and ∂r was not significant. As the minimum ∂a value was raised, a smaller group of user-film pairs were selected where the users made ratings that differed increasingly with the average. We incremented the minimum threshold of ∂a by 0.5, and then computed the new ∂ values The results are shown in Figure 6.

Notice that the ∂a value increases about as expected. The ∂r, however, is clearly increasing at a slower rate than ∂a. At each step, as the threshold for ∂a is increased by 0.5, ∂r increases by an average of less than 0.1. A two-tailed t-test shows that at each step where the minimum ∂a threshold is greater than or equal to 0.5, the recommended rating is significantly closer to the actual rating than the average rating is, with $p<0.01$. When compared to the ACF algorithm, there were similar results. For $\partial a<1$, there was no significant difference between the accuracy of the ACF ratings and the trust-based recommended rating. However, when the gap between the actual rating and the average increases, for $\partial a>=1$, the trust-based recommendation outperforms the ACF as well as the average, with $p<0.01$.

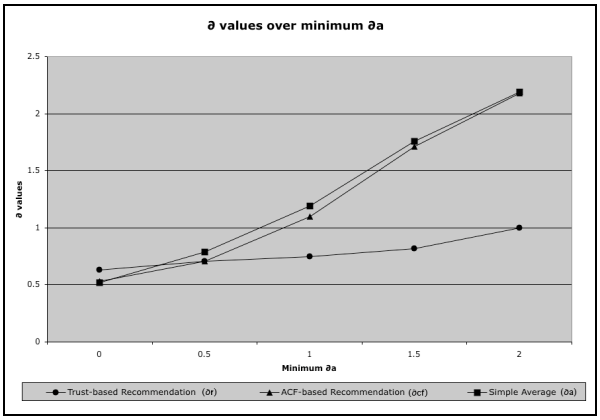

Fig 6. The increase in ∂ as the minimum ∂a is increased. Notice that the ACF-based recommendation (∂cf) follows the average (∂a). The more accurate Trust-based recommendation (∂r) significantly outperforms both other methods.

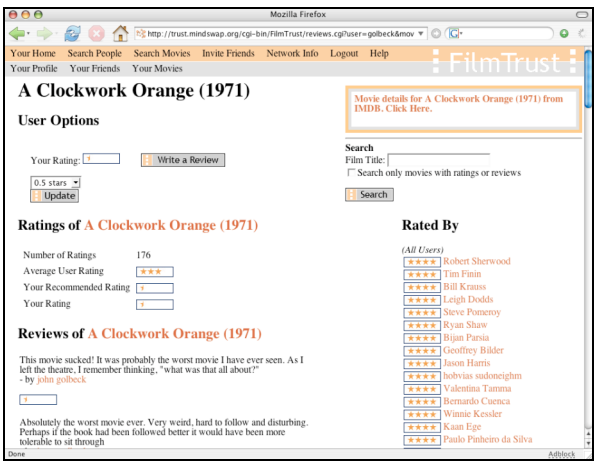

Fig 7. A user's view of the page for "A Clockwork Orange," where the recommended rating matches the user's rating, even though ∂a is very high (∂a = 2.5).

Figure 7 shows a clear example of the personalized rating at work. "A Clockwork Orange" is one of the films in the database that has a strong collective of users who hated the movie, even though the average rating was 3 stars and many users gave it a full 4-star rating. For the user shown, the average rating of 3 stars is 2.5 stars above the user's actual rating while the recommended rating exactly matches the user's low rating of 0.5 stars. These are precisely the type of cases that the recommended rating is designed to address.

The purpose of this work is not necessarily to replace more traditional methods of collaborative filtering. It is very possible that a combined approach of trust with correlation weighting or another form of collaborative filtering may offer equal or better accuracy, and it will certainly allow for higher coverage. However,

these results clearly show that, in the FilmTrust network, basing recommendations on the expressed trust for other people in the network offers significant benefits for accuracy.

In addition to presenting personalized ratings, the experience of reading reviews is also personalized. The reviews are presented in order of the trust value of the author, with the reviews from the most trustworthy people appearing at the top, and those from the least trustworthy at the bottom. The expectation is that the most relevant reviews will come from more trusted users, and thus they will be shown first. A preliminary user study suggests that this ordering is beneficial to users, but further work is necessary to refine and confirm these results.

3.2 TrustMail: Trust Networks for Email Filtering

TrustMail is a prototype email client that adds trust ratings to the folder views of a message. This allows a user to see their trust rating for each individual, and sort messages accordingly. This is, essentially, a message scoring system. The benefit to users is that relevant and potentially important messages can be highlighted, even if the user does not know the sender. The determination of whether or not a message is significant is made using the user's own perspective on the trust network, and thus scores will be personalized to and under the control of each user.

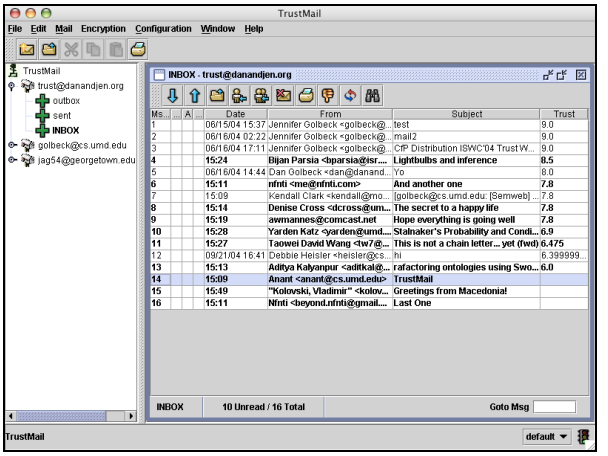

Fig 8.The TrustMail Interface. In this window, messages are sorted according to the trust rating of the sender, with the most trusted appearing highest in the list.

The values shown next to each message are trust ratings calculated with the TidalTrust algorithm were recipient as the source, and the sender is the sink. Techniques that build social networks from messages that the user has sent or received can identify whether or not a message has come from someone in the network. However, because they are built only from the user's local mail folders, no information is available about people that the user has not previously seen. If the user's personal network is connected in to a larger social network with information from many other users, much more data is available. Previously unseen senders can be identified as part of the network.

Furthermore, since trust values are available in the system, the methods for inferring trust can be applied to present more information to the user about the sender of a message. In the FilmTrust system, preliminary studies suggest that users benefited from having movie reviews sorted by the trustworthiness of the author. These results also suggest a benefit from sorting messages by the trustworthiness of the sender in TrustMail. However, unlike the FilmTrust where every review was authored by someone in the social network, people will undoubtedly receive many email messages from people who are not in their social network. To understand what benefit TrustMail might offer to users, it is important to understand what percentage of messages we can expect to have ratings for in TrustMail. The next section uses a real email corpus to gain some insight into this question.

To gain some insight into how TrustMail may impact a user's mailbox, a large network with many users is required. The Enron email dataset is a collection of the mail folders of 150 Enron employees, and it contains over 1.5 million messages, both sent and received. There are over 6,000 unique senders in the corpus, and over 28,000 unique recipients. These numbers are much greater than the number of users whose mailboxes have been collected because they represent everyone who has sent a message to the users, everyone who has been cc-ed on a message to the users, and everyone the users have emailed. The collection was made available by the Federal Energy Regulatory Commission in late 2003 in response to the legal investigation of the company. Because the messages represent a single community, they are ideal for analyzing the potential of TrustMail. Each message in the corpus was read, and an edge was added from the sender to each of the recipients. This produced an initial social network, although the connections are weak. To be more sure that the links between people represented a relationship, connections were removed for any interactions that occurred only once; edges were only added from source to sink when the source had emailed the sink at least twice.

An analysis of the Enron network showed the following statistics:

- 37% of recipients had direct connections to people who sent them email in the social network; in other words, 37% of the time the recipient had emailed the sender of a received message.
- 55% of senders who were not directly connected to the recipient could be reached through paths in the social network.
- Thus, a total of 92% of all senders can be rated if trust values were present in the social network.

These numbers indicate that for users in a community like Enron, an application like TrustMail can provide information about a majority of the incoming messages. While the Enron corpus is a close community of users, it is reasonable to expect that, if users are properly supported in making trust ratings as part of their email client, a similarly high percentage of senders and messages would receive ratings in other contexts.

4 Conclusions

In this paper, we have used an analysis of the properties of trust networks to develop the TidalTrust algorithm for inferring trust relationships between people with no direct connections. We integrated this algorithm into two systems: FilmTrust, where it was used to generate predictive ratings of movies, and TrustMail where the results are

used as a score for email messages. The data that allows these analyses to be performed and applications to be created all comes from the large repository of social network data on the Semantic Web.

While trust was the relationship feature analyzed here, the general technique of network analysis for developing algorithms is one we believe holds much promise. For example, a simplified version of the TrustMail application that utilizes basic social connectivity instead of trust ratings may be quite effective for filtering and scoring messages. Millions of people's social networks are represented in FOAF on the Semantic Web, and taking advantage of this large network as a source of application enhancing information holds promise for improving the usability and utility of many applications.

References

1. FOAF Vocabulary: http://xmlns.com/foaf/0.1/
2. Golbeck, Jennifer (2005) "Computing and Applying Trust in Web-Based Social Networks," Ph.D. Dissertation, University of Maryland, College Park.
3. Sinha, R., and Swearingen, K. (2001) "Comparing recommendations made by online systems and friends." *In Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries* Dublin, Ireland.
4. Swearingen, K. and R. Sinha. (2001) "Beyond algorithms: An HCI perspective on recommender systems," *Proceedings of the ACM SIGIR 2001 Workshop on Recommender Systems*, New Orleans, Louisiana.
5. Ziegler, Cai-Nicolas, Georg Lausen (2004) Analyzing Correlation Between Trust and User Similarity in Online Communities" *Proceedings of Second International Conference on Trust Management*, 2004.
6. Gil, Yolanda and Varun Ratnakar. (2002) "Trusting Information Sources One Citizen at a Time," *Proceedings of the First International Semantic Web Conference (ISWC)*, Sardinia, Italy.
7. Kamvar, Sepandar D. Mario T. Schlosser, Hector Garcia-Molina (2003) “The EigenTrust Algorithm for Reputation Management in P2P Networks”, *Proceedings of the 12th International World Wide Web Conference*, May 20-24, 2003, Budapest, Hungary.
8. Page, L., Brin, S., Motwani, R., & Winograd, T. (1998) “The PageRank citation ranking: Bringing order to the web.” *Technical Report 1998*, Stanford University, Stanford, CA.
9. Levin, Raph and Alexander Aiken. (1998)"Attack resistant trust metrics for public key certification." *7th USENIX Security Symposium*, San Antonio, Texas.
10. Richardson, Matthew, Rakesh Agrawal, Pedro Domingos. (2003) “Trust Management for the Semantic Web,” *Proceedings of the Second International Semantic Web Conference*. Sanibel Island, Florida.
11. Massa, P., P. Avesani. 2004. Trust-aware Collaborative Filtering for Recommender Systems. In *Proceedings of the International Conference on Cooperative Information Systems (CoopIS) 2004*.
12. Herlocker , Jonathan L., Joseph A. Konstan , Loren G. Terveen , John T. Riedl, (2004) Evaluating collaborative filtering recommender systems, *ACM Transactions on Information Systems (TOIS)*, v.22 n.1, p.5-53, January 2004.