Incorporating Evidence into Trust Propagation Models Using Markov Random Fields

Hasari Tosun
Department of Computer Science
Montana State University
EPS 357, PO Box 173880
Bozeman, MT 59717-3880
tosun@cs.montana.edu

John W. Sheppard
Department of Computer Science
Montana State University
EPS 357, PO Box 173880
Bozeman, MT 59717-3880
john.sheppard@cs.montana.edu

***Abstract*—Current trust models for social networks commonly rely on explicit voting mechanisms where individuals vote for each other as a form of trust statement. However, there is a wealth of information about individuals beyond trust voting in emerging web based social networks. Incorporating sources of evidence into trust models for social networks has not been studied to date. We explore a trust model for social networks based on *Markov Random Fields*, which we call MRFTrust, that allows us to incorporate sources of evidence. To allow comparative evaluation, a state-of-the-art local trust algorithm—MoleTrust—is also investigated. Experimental results of the algorithms reveal that our trust algorithm that incorporates evidence performs better in terms of coverage. It is competitive with the MoleTrust algorithm in prediction accuracy and superior when focusing on controversial users.**

***Keywords*-Trust Metrics, Reputation System, Social Network, Markov Random Fields**

I. Introduction

Recently, online Web services such as MySpace, FaceBook, Friendster, LiveJournal, Blogger, LinkedIn, Twitter, and Orkut have emerged as popular social networks. This new generation of social networks is enormous, rich in information, and extremely dynamic. Moreover, in today's Web, a vast amount of content is created by users. This content can range from factual information to opinions about a person, a product, or a company. People constantly interact with other people about whom they have no immediate information. As a result, users of these services are constantly faced with questions of how much they should trust the content created or opinion provided by another person and how much they should trust the unknown person with whom he or she is about to interact.

With this uncertainty in the mind, many e-commerce companies such as eBay and Amazon enable users to rate other users or their reviews by providing a trust vote. Most online forums have some mechanism for users to rate others' opinions or responses. In some cases, the voting is implicit. For example, reading an article can be considered an implicit positive vote. Utilizing this vast amount of trust data or aggregating trust scores for users have become a real challenge for those companies. Trust and reputation is also very relevant to Peer-to-Peer (P2P) networks such as file-sharing networks. P2P networks are mainly used for sharing and distributing information. Thus, they are vulnerable to the spread of unauthentic files [1], [2], [3], [4]. An alternative utilization of the trust concept is used by the Google search engine; a link from one web site to another is an expression of trust [5].

As the Semantic Web gains acceptance, understanding the credibility of metadata about authors is becoming important [6]. While designing recommender systems, one researcher found that there is a strong correlation between trust and user similarity [7]. Thus, trust became the essential variable in computing user similarity [8]. Finally, trust concept is extensively applied to social networks. There is a wealth of information on trust and reputation scoring in social networks [9], [10], [8], [6], [11]. For example, Mui documented the theories and approaches about trust scores and reputation systems using Bayesian networks for inference on social networks [12].

There is no universal definition of trust and reputation. Barbalet characterized trust and its consequences in detail. He postulates that it is insufficient to define trust in terms of "confident expectation regarding another's behavior" as many researchers defined [13]. Instead, the author characterizes trust in terms of acceptance of dependency (the trust giver grants control or power to trustee; thus, the trust giver accepts dependence on trustee) in the absense of information about the other's reliability in order to create an outcome otherwise unavailable. Golbeck and Hendler adopted a narrower definition of trust for social networks—"trust in a person is a commitment to an action based on belief that the future actions of that person will lead to a good outcome" [11].

Although, researchers generally don't aggree on the definition of trust, two properties of trust are used for aggregation: *transitivity* and *asymmetry*. *Transitivity* means if A trusts B who trusts C, then, A trusts C. *Asymmetry* means if A trusts B, it doesn't mean that B will also

978-1-61284-937-9/11/$26.00 ©2011 IEEE

trust A. The majority of trust propagation algorithms utilize the transitivity property [10], [1], [14], [6], [2]. It should be noted this property may not always work with distrust [15]. Moreover, [16] and [17] defined two types of trusts: *referreral trust* and *direct functional trust*. If A trusts B who trusts C, then, the trust between A-B and B-C is *direct functional trust*. However, if B recomends C to A as trustworthy, it is *referral trust*.

Modeling trust networks and propagating trust is a challenging task: 1) trust networks are huge and sparse, and 2) it is often difficult to model human belief and trust. Thus, researchers have often proposed simplistic approaches for trust propagation. Ziegler and Lausen categorized trust metrics on three dimensions [14]: a network perspective, computation locus, and link evaluation. For the network perspective, they categorized trust metrics as global and local. Global trust metrics consider all links and nodes in the networks, where local trust metrics take into account only a partial network. Computational locus refers to the place where trust relationships are computed. In determining distributed or local trust metrics, the computation load is distributed to every node in the network, whereas in computationally centralized systems, all metric computations are performed on a single machine. The tradeoff between the two approaches involves memory, performance, and security. Finally, link evaluation determines whether the trust metrics themselves are scalar or group trust metrics. In group trust metrics, trust scores are computed for a set of individuals whereas for scalar metrics only trust scores between two individuals are computed.

An increasing number of articles have been published on modeling trust networks and evaluating trust metrics [4], [15], [14], [11], [1], [18], [16], [13], [3] using different computational methods. For example, Wang and Vassileva designed a Bayesian Network-based trust model for P2P [3]. The model represents different features of trust as leaf nodes of Naive Bayes networks. On the other hand, [18] developed a model based on Fuzzy logic. Another popular trust model is the *Appleseed* Trust metric based on a *Spreading Activation Model* [14], [6]. Two different models based on eigenvalue propagation were designed by [1] and [15].

Massa and Avesani studied challenges of computing trust metrics in a social network where data are sparse [19]. In such networks, neither a global reputation nor a simple trust score is a viable option since a large percentage of the participants are considered to be controversial; they are distrusted by some and trusted by others. Thus, the authors proposed a state-of-the-art framework and algorithm called *MoleTrust* that uses local trust metrics. However, this approach does not incorporate other sources of evidence. For example, the *epinion.com* dataset contains articles written by users [20] where these articles are also rated by other users. Determining if such information is useful for a trust algorithm is the challenge. We hypothesize that including more evidence into a trust model will improve the prediction power of the model and its coverage.

The rest of the paper is organized as follows. Our trust prediction algorithm based on Markov Random Fields is described in Section II. Then in Section III, we describe the dataset and present our results. The last section gives the conclusions and future directions for our research.

II. Methods and Procedures

In this paper, we describe our approach to developing and using a trust network model based on Markov Random Fields (MRFs). A detailed introduction to MRFs is given in [21]. An MRF is a stochastic process that exhibits the Markov property in terms of the interaction of neighboring nodes in the network. MRF models have a wide range of application domains. The nodes in the MRF graph represent random variables, and the edges represent the dependencies between variables. In our approach, we use the same type of model for propagating the trust scores in social networks.

The joint probability distribution over *X* and *Y* can be represented by an MRF in the following way:

$$\mathbf{P}(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{i,j} \psi(x_i, x_j) \prod_{i} \phi(x_i, y_i)$$

where *Z* is a normalization factor (also called the partition function), $\psi(x_i, x_j)$ represents pairwise influence between node x_i and x_j in the network (often referred to as the pairwise compatibility matrix), and $\phi(x_i, y_i)$ is a local evidence function that forms a distribution over possible states, x_i, given only its observations y_i. When considering the application of MRFs to social network trust prediction, we note that the social network results in a trust network whenever users rate each other. Based on this observation, we developed a local algorithm for learning trust metrics by augmenting an MRF representation of social networks with additional sources of evidence. Our framework allows us to evaluate an active user's trust for an unknown person in the network.

There are two common methods for inference with MRF models [22]: 1) Markov Chain Monte Carlo (MCMC) sampling, such as Gibbs sampling, and 2) Belief Propagation. The approach we used is based on belief propagation, so we begin by describing the main steps in the Belief Propagation algorithm. Essentially, belief propagation proceeds as follows:

1) Select random neighboring nodes x_k, x_j
2) Send message M_j^k from x_k to x_j
3) Update the belief about the marginal distribution at node x_j
4) Go to step one until convergence.

Message passing in step 2 is carried out as

$$M_j^k = \sum_{x_k} \psi_{kj}(x_k, x_j)\, b(x_k) \tag{1}$$

where $b(x_k)$ is the current belief value associated with node x_k. Belief updating in step 3 is then computed as

$$b(x_j) = \kappa\, \phi(x_j, y_j) \prod_{k \in Neighbor(j)} M_j^k \qquad (2)$$

where κ is a normalization factor, and $Neighbor(j)$ is the set of nodes adjacent to node x_j.

To infer the trust score of users unknown to the current user in that network, a local network is generated from the global social network. The local trust network has a limited horizon. In other words, instead of propagating trust statements using the global network, we create a local network based on a specific user's neighborhood. For example, for a given user *A*, we generate a local network that contains all neighboring users that are only a finite distance (in terms of the number of links crossed) away from *A*. Thus, the trust score of a person in the local network can be evaluated with respect to the active user *A*.

We compared the results of our approach to modeling trust propagation to the *MoleTrust* algorithm, presented in [19]. The process of generating the local network is similar to *MoleTrust* in that it is based on the intuition that the average trust path length between two individuals is small [14]. Moreover, due to computational complexity and the objective that any trust prediction system operate online, the local network needs to be small.

For purposes of our experiments, we re-implemented *MoleTrust* to ensure a fair and carefully-controlled comparison. To predict how much a user *A* trusts a user *B*, denoted $T(A, B)$, *MoleTrust* generates a local directed graph from a given global social network whose root is *A*. For each graph depth, it adds links that represent trust statements between users. To avoid cycles, it does not add nodes if they are already in the local network. The depth or distance of the graph is determined by a parameter called the *horizon*. If the target user is in the local graph, a trust prediction is made. Otherwise, the trust prediction is not made. As we will see, this restriction has a direct impact on the coverage of the *MoleTrust* algorithm. Trust propagates from the root node to the leaf nodes with equation 3, where $b(x_j)$ is the trust value or belief predicted at node x_j:

$$b(x_j) = \frac{\sum_{k \in Predecessor(j)} b(x_k)\, T(x_k, x_j)}{\sum_{k \in Predecessor(j)} b(x_k)} \qquad (3)$$

Here $T(x_k, x_j)$ is the trust value on the edge between node x_k and x_j, and $Predecessor(j)$ is the set of nodes with edges terminating at x_j. The trust values for the nodes are calculated from this equation, whereas the trust values on the edges are specified explicitly in the network. Edge trust values represent explicit trust voting of one user about the other. To start the belief propagation process, the belief of the root node is initialized to 1.0.

MoleTrust was designed to address the issue of predicting trust in the presences of controversial users. The most controversial users have approximately equal numbers of distrust and trust statements. The controversiality level of a user is given in equation 4 [19]:

$$c(x_j) = \frac{|\mathbf{Trust}(x_j)| - |\mathbf{Distrust}(x_j)|}{|\mathbf{Trust}(x_j)| + |\mathbf{Distrust}(x_j)|} \qquad (4)$$

where $\mathbf{Trust}(x_j)$ is the set of trust statements for user/node x_j, and $\mathbf{Distrust}(x_j)$ is the set of distrust statements for x_j. This controversiality level has the range of $-1.0 \ldots 1.0$. A user with controversiality level of -1.0 is distrusted by all his or her judgers, where a user with controversiality level of 1.0 is trusted by all users who voted. On the other hand, a user with controversiality 0.0 has an equal number of trust and distrust votes. Therefore, a user with 0.0 controversiality is the most controversial user. We discretized the controversiality levels of the users into buckets of width 0.1. Finally, we define the *coverage* of a prediction algorithm to be the percentage of statements that are predictable by that algorithm.

The *MoleTrust* algorithm accepts an incoming link to node x_j if the predicted trust value of x_j's corresponding parent node is above a threshold. Otherwise, the link between them is blocked from propagating the trust scores. We claim this approach biases the performance of the *MoleTrust* algorithm by limiting its predictions to those that are "easy." Although this approach may result in accurate trust predictions, it results in low coverage. For a given graph depth or horizon, if the user is not in the local network, *MoleTrust* cannot make a prediction. Moreover, if trust propagation does not reach the target user due to the link blocking explained above, the prediction cannot be made. For example, if we were to predict *Alice*'s trust in *Mark* based on the network in Figure 1 without a direct link between them, the trust will propagate indirectly through nodes *Bob* and *Dave*. However, if neither *Bob* nor *Dave* have direct links to *Mark*, a prediction cannot be made by *MoleTrust*. The other situation arises when the all parent nodes of the target node *Mark*, namely *Bob* and *Dave*, have the calculated trust score below the given threshold. Since they will be blocked from propagating their trust scores, once again *MoleTrust* cannot make a prediction. As a result the coverage of *MoleTrust* is bounded by the local network structure and the threshold parameter.

In contrast to the approach above, we designed an algorithm based on the Belief Propagation in MRFs that overcomes the coverage limitations of *MoleTrust*. In our approach, we construct a local network similar to *MoleTrust* but with undirected links. In addition, each node is also attached to evidence nodes. An evidence node represents a respective user's observation about the target node. The evidence node, thus, creates an indirect link between two nodes. For example, in Figure 2, if both *Bob* and *Dave* rated one or more articles written by *Mark*, they have created

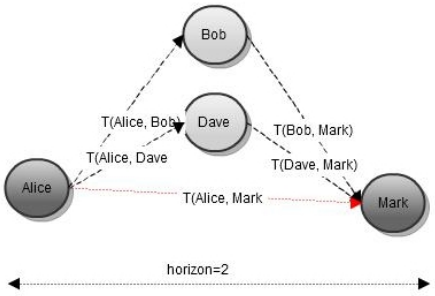

Figure 1. Determining Coverage in *MoleTrust*

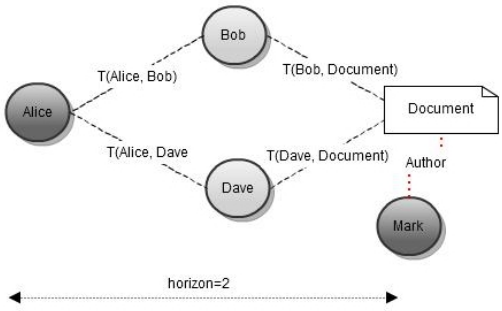

Figure 2. Determining Coverage in *MRFTrust* with Evidence

indirect links based on this evidence about *Mark*. When neither *Bob* nor *Dave* have direct links to *Mark*, a prediction still can be made through these indirect links.

As described in section III below, the *epinions.com* dataset contains article ratings that can be used as evidence. Our algorithm exploits this evidence to increase the prediction coverage. To the best of our knowledge, no other trust propagation algorithm has been developed that incorporates other sources of evidence into their trust calculations. Our MRF based algorithm, which we call *MRFTrust*, uses the Belief Propagation algorithm defined above; however, unlike *MoleTrust*, messages are propagated in our model from nodes that are also connected to the target node. We proceed from the assumption that the neighbors of the target node have a more reliable estimate of trust. In other words, a user may be globally more controversial even though he or she is less controversial in a local graph. Thus, propagating from neighbors to the source node is in essence propagating this more reliable assessment to the source node, resulting in a better prediction. When constructing our graph, we ensure that the resulting graph is a tree structure to control the computational complexity of the belief propagation algorithm. For example, Figure 3 shows the result of generating the tree from the graph constructed in Figure 2. The marginal at the source node represents the probability that the person associated with the source node

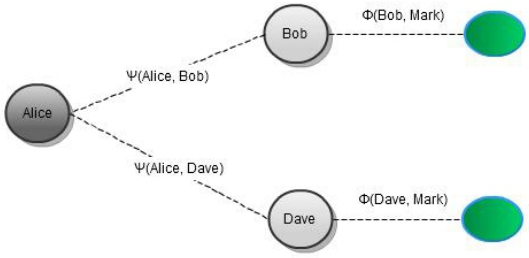

Figure 3. MRFTrust Network with Evidence Nodes

will trust the person associated with the target node. The process by which the subgraph/tree is constructed results in that marginal being conditioned on the evidence about the target node.

In our algorithm, the initial belief for the neighbor nodes of the target node in equation 1 is calculated as

$$b(x_k) = \begin{cases} T(x_k, Target) & \text{Link Exists} \\ \phi(x_k, y_k) & \text{Otherwise} \end{cases}$$

where $\phi(x_k, y_k)$ is node k's observation about the target node, calculated as

$$\phi(x_k, y_k) = \begin{bmatrix} \theta \\ 1-\theta \end{bmatrix}$$

and θ is the average rating issued by x_k on articles or documents written by the target user, normalized by the maximum rating for the articles. For example, if *Bob* rated three articles written by *Mark* with an average score of 4.0 out of maximum score of 5.0, the local evidence is $\begin{bmatrix} 0.8 & 0.2 \end{bmatrix}^T$. Thus, with probability 0.8, *Bob* trusts *Mark*. Equivalently, with probability 0.2, *Bob* distrusts *Mark*. Since the algorithm needs to be run for each pair of users and the inference has to be done online, the time complexity of the inference should be kept linear. As such, we limited the propagation of belief to only one direction. Considering bidirectional propagation will be considered as future work. Thus, marginal probability distributions are only approximate. The compatibility matrix $\psi(x_k, x_j)$ is then constructed as

$$\psi(x_k, x_j) = \begin{bmatrix} T(x_k, x_j) & 1 - T(x_k, x_j) \\ 1 - T(x_k, x_j) & T(x_k, x_j) \end{bmatrix}$$

$T(x_k, x_j)$ represents the average value of trust between nodes *k* and *j*. For example, if node *k* trust node *j*, the value is 1.0. A value of 1.0 effectively means that users are compatible. On the other hand, if node *k* trusts *j* and *j* does not trust *k*, its value is 0.5. Finally, in our algorithm the prediction is made if at least 10 users have evidence on the target user or the target user is in the local network. Otherwise, no prediction is made.

III. Experiments and Analysis

In this study, the *epinions.com* dataset for collaborative filtering was used [20]. *Epinions.com* is a website where users can write reviews about products and rate other reviews. It also allows users to express their *Web of Trust* (i.e., reviewers they trust) and their *Block List* (i.e., a list of authors they distrust). Thus, the dataset contains user ratings of articles created by other users as well as user ratings of each other in the form of *Web of Trust* and the *Block List*. The value of 1 is used for trust statements (*Web of Trust*), and -1 is used for distrust statements (*Block List*). We rescaled the trust scores so that the distrust statements would have a value of 0. This ensured the trust values were in the range $0.0 \ldots 1.0$. The article ratings represent how likely a user rates a certain textual article written by another user. The rating value has a range $1 \ldots 5$ where 1 is "not helpful" and 5 is "very helpful."

The *epinions.com* data set contains about 132,000 users, who issued 841,372 statements of trust or distrust. We removed 573 trust statements corresponding to statements where the recipients of the statements were also the senders of the same statements. In testing our approach, we used a *leave-one-out* experimental design to evaluate the performance of the model. Specifically, for every existing relationship between users *A* and *B* in the data set, we removed a true trust statement from user *A* to user *B* and then constructed the local network between them. Then, we predicted *A*'s trust statement for *B* based on that network. We repeated this procedure to evaluate *MoleTrust* as well. We also implemented two naive algorithms—*Gullible* and *Skeptic*. *Gullible*, outputs "trust" if the target is in the local network; otherwise, it it doesn't do any prediction. *Skeptic*, also outputs "trust" if the target is in the local network; however, unlike *Gullible*, it outputs "distrust" if the target is not in the local network. Thus, *Skeptic*'s coverage is 100%. All tests were performed using $horizon = 2$ since the reported experiments on *MoleTrust* used this horizon. Because of this, we denote all implemented algorithms with a "2" after the name (e.g., *MRFTrust2*). We re-implemented *MoleTrust2* as described in [19] with the same parameters they report using. Specifically, the threshold value of blocking a trust links was set to 0.6.

MRFTrust2 did not perform as accurately as *MoleTrust2* for distrusted users. However, *MRFTrust2* performed comparably with high controversiality users and virtually identically for trusted users. Figure 4 shows the coverage plotted against the controversiality level for each algorithm. Although *MoleTrust2* has a lower error rate (especially in areas of distrust), its coverage is also very low; as described in section II, its coverage is bounded by the structure of the local network and the threshold parameter. As shown in the Figure 4, its coverage is bounded by *Gullible2*'s coverage. Moreover, *MoleTrust2*'s coverage is around 30% on the most controversial bucket of users. On the other hand,

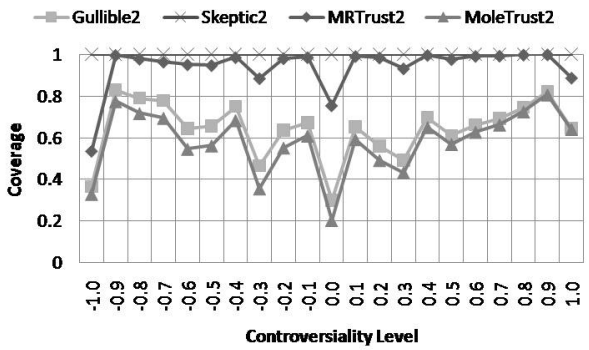

Figure 4. Coverage Percentage by Controversiality Level

MRFTrust2's coverage is significantly higher. As shown in the figure, its coverage is close to 75% on the most controversial users and approaches 100% on most of the remaining levels. The highest coverage *MoleTrust2* ever achieves is around 80%.

The fact that *MRFTrust* and *MoleTrust* have radically different coverage makes direct comparison between them difficult. In an attempt to make the comparison between these two algorithms fair, we designed two experiments to compare *MRFTrust2* to *MoleTrust2* in ways that ensure the coverage is the same for both algorithms. In the top part of Figure 5, *MRFTrust2+* is the *MRFTrust2* algorithm that runs only on cases where *MoleTrust2* can make a prediction, whereas *MRFTrust2–* is the *MRFTrust2* algorithm that runs on cases where *MoleTrust2* cannot make a prediction. It is not surprising that *MRFTrust2–* has higher error rates. However, in most cases, the error rate is below 50%. Furthermore, *MRFTrust2+* and *MoleTrust2* have comparable results. *MoleTrust2* continues to be significantly better than *MRFTrust2+* in the controversiality range $-1.0 \ldots -0.4$ (while being much closer in performance as compared to the original experiments), but the results are statistically comparable in the range $-0.3 \ldots -0.1$. Then from $0.0 \ldots 1.0$, MRFTRUST2+ is significantly better. Confidence intervals for *MRFTrust2+* and *MoleTrust2* are given in the second and third columns of Table III.

Similar to the previous experiment, we designed another experiment to compare *MoleTrust2* to *MRFTrust2* where we increased the coverage of *MoleTrust2* to be equivalent to *MRFTrust2*. In the bottom part of Figure 5, *MoleTrustRandom2+* is a version of *MoleTrust2* where it makes a random prediction if the target is not in the local network. Moreover, it only runs on cases where *MRFTrust2* can make a prediction. On the other hand, *MoleTrustRandom2–* is similar to *MoleTrustRandom2+* except it only runs on cases where *MRFTrust2* cannot make a prediction. The results show that *MRFTrust2* performs equivalent to or better than *MoleTrustRandom2+* if the two algorithms have the

Cntr	*Mole2*	*MRF2+*	*MRF2*	*MoleRnd2+*
-1.0	0.0014 ± 0.0006	0.0667 ± 0.0040	0.2543 ± 0.0050	0.1941 ± 0.0042
-0.9	0.0270 ± 0.0037	0.0929 ± 0.0068	0.1787 ± 0.0072	0.1306 ± 0.0058
-0.8	0.0569 ± 0.0060	0.1430 ± 0.0099	0.2263 ± 0.0094	0.1733 ± 0.0075
-0.7	0.0714 ± 0.0071	0.1301 ± 0.0106	0.2180 ± 0.0102	0.1898 ± 0.0084
-0.6	0.1052 ± 0.0129	0.1523 ± 0.0165	0.2902 ± 0.0143	0.2717 ± 0.0121
-0.5	0.1231 ± 0.0114	0.1594 ± 0.0145	0.2838 ± 0.0126	0.2779 ± 0.0106
-0.4	0.1723 ± 0.0139	0.2321 ± 0.0184	0.3056 ± 0.0158	0.2652 ± 0.0124
-0.3	0.1696 ± 0.0170	0.1520 ± 0.0177	0.3326 ± 0.0132	0.3674 ± 0.0112
-0.2	0.1855 ± 0.0163	0.1821 ± 0.0188	0.3053 ± 0.0155	0.3230 ± 0.0129
-0.1	0.2151 ± 0.0136	0.2274 ± 0.0178	0.2958 ± 0.0143	0.3251 ± 0.0113
0.0	0.2904 ± 0.0189	0.2134 ± 0.0191	0.3322 ± 0.0101	0.4422 ± 0.0082
0.1	0.2409 ± 0.0109	0.2285 ± 0.0144	0.3138 ± 0.0116	0.3522 ± 0.0089
0.2	0.2479 ± 0.0126	0.1842 ± 0.0142	0.2976 ± 0.0110	0.3738 ± 0.0089
0.3	0.2466 ± 0.0100	0.1731 ± 0.0113	0.2866 ± 0.0085	0.3860 ± 0.0069
0.4	0.2588 ± 0.0077	0.1825 ± 0.0097	0.2513 ± 0.0084	0.3413 ± 0.0065
0.5	0.2422 ± 0.0066	0.1458 ± 0.0075	0.2456 ± 0.0066	0.3506 ± 0.0053
0.6	0.2258 ± 0.0046	0.1353 ± 0.0054	0.2211 ± 0.0049	0.3257 ± 0.0040
0.7	0.1692 ± 0.0035	0.0943 ± 0.0038	0.1724 ± 0.0038	0.2783 ± 0.0034
0.8	0.1183 ± 0.0018	0.0603 ± 0.0018	0.1312 ± 0.0021	0.2216 ± 0.0020
0.9	0.0555 ± 0.0008	0.0278 ± 0.0007	0.0792 ± 0.0010	0.1414 ± 0.0011
1	0.0079 ± 0.0003	0.0041 ± 0.0003	0.0886 ± 0.0010	0.1446 ± 0.0011

Table I
MEAN ABSOLUTE ERROR 95% CONFIDENCE INTERVALS FOR *MoleTrust2 (Mole2)* VS *MRFTrust*2+ (MRF2+) AND *MrfTrust2 (MRF2)* VS. *MoleTrustRandom2+* (*MoleRnd2*)

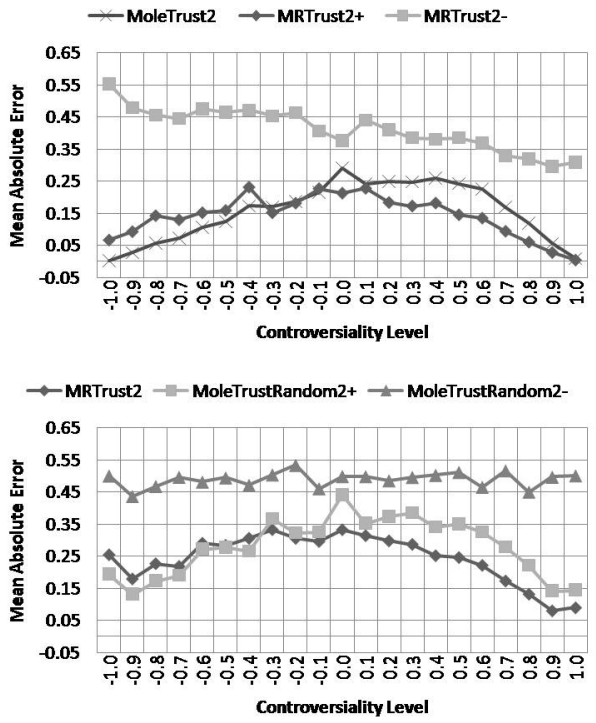

Figure 5. *MRFTrust*2 Compared to *MoleTrust*2 Given Equivalent Coverage

same coverage. Confidence intervals for *MRFTrust2* and *MoleTrustRandom2+* are given in columns four and five of Table III.

IV. CONCLUSION

We have proposed a new algorithm for incorporating evidence into trust prediction within the context of social networks. Our algorithm utilizes a Markov Random Field formulation to represent trust/distrust relationships between users and incorporates evidence nodes capturing user ratings of articles written by other users in the network. We discussed several experiments, comparing our *MRFTrust* algorithm to the state-of-the-art *MoleTrust* using the *epinions.com* data set. Incorporating evidence into the model enabled us to utilize implicit trust relationships to predict trust and thereby increase the prediction coverage.

One of the key complexities in performing our experiments arose based on the substantial differences in coverage exhibited by the two algorithms. To address this, we included two experiments where we attempted to equalize coverage. In one case, we limited the instances covered by *MRFTrust2* to be the same as those covered by *MoleTrust2*. In the second case, we extended *MoleTrust2* to make random predictions on those examples it did not cover but that were covered by *MRFTrust2*. Of particular interest was that, under both of these conditions, the performance between the *MRFTrust* and *MoleTrust* algorithms either became comparable, or the *MRFTrust* algorithms beat the *MoleTrust* algorithms. This is particularly interesting in the light of the fact the base algorithms showed a much stronger advantage to the *MoleTrust* algorithm (ignoring the issue of coverage). At this point, it is difficult to explain the difference in performance; however, we plan to focus on such an analysis by considering the particular characteristics of the data and the role of evidence in the data.

As future work, we plan to explore alternative and additional sources of evidence to be included into the MRF model. A candidate for another source of evidence could be based on how source and target users rate common articles. We speculate that if they both rate the same article with the same score, the likelihood of them trusting each other would be higher than if they rated the same article with different scores. Thus by incorporating evidence nodes that reflect "correlation" in article scoring, we may find even higher accuracy in predicting trust. In addition, we plan to evaluate the effect on computational burden and prediction accuracy when increasing the horizon in the local graphs. Finally, we began to explore different ways of combining the results of *MRFTrust* and *MoleTrust* to yield an ensemble-based model. Initial results indicated an over-emphasis on the *MRFTrust* portion that failed to incorporate the high-accuracy parts of *MoleTrust* adequately. Alternative approaches will be explored to compensate for this problem.

REFERENCES

[1] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *WWW '03: Proceedings of the 12th international conference on World Wide Web*. New York, NY, USA: ACM, 2003, pp. 640–651.

[2] Y.-F. Wang, Y. Hori, and K. Sakurai, "Characterizing economic and social properties of trust and reputation systems in p2p environment," *Journal of Computer Science*

and Technology, vol. 23, pp. 129–140, 2008. [Online]. Available: http://dx.doi.org/10.1007/s11390-008-9118-y

[3] Y. Wang and J. Vassileva, "Bayesian network-based trust model," in *WI '03: Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*. Washington, DC, USA: IEEE Computer Society, 2003, p. 372.

[4] L. Ding, P. Kolari, S. Ganjugunte, T. Finin, and A. Joshi, "Modeling and evaluating trust network inference," in *Proceedings of The Workshop on Deception, Fraud and Trust in Agent Societies at The Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2004*, 2004, pp. 21–32.

[5] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford, Tech. Rep., 1998.

[6] C.-N. Ziegler and G. Lausen, "Propagation models for trust and distrust in social networks," *Information Systems Frontiers*, vol. 7, pp. 337–358, 2005. [Online]. Available: http://dx.doi.org/10.1007/s10796-005-4807-3

[7] C. Ziegler and G. Lausen, "Analyzing correlation between trust and user similarity in online communities," in *Trust Management*, ser. Lecture Notes in Computer Science, C. Jensen, S. Poslad, and T. Dimitrakos, Eds. Springer Berlin / Heidelberg, 2004, vol. 2995, pp. 251–265. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24747-0_19

[8] F. Walter, S. Battiston, and F. Schweitzer, "A model of a trust-based recommendation system on a social network," *Autonomous Agents and Multi-Agent Systems*, vol. 16, pp. 57–74, 2008. [Online]. Available: http://dx.doi.org/10.1007/s10458-007-9021-x

[9] J. Golbeck, "Computing and applying trust in web-based social networks. ph.d. thesis." University of Maryland, Tech. Rep., 2005.

[10] P. Massa and P. Avesani, "Controversial users demand local trust metrics: an experimental study on epinions.com community," in *Proceedings of the 25th American Association for Artificial Intelligence Conference*, 2005.

[11] J. Golbeck and J. Hendler, "Inferring binary trust relationships in web-based social networks," *ACM Trans. Internet Technol.*, vol. 6, no. 4, pp. 497–529, 2006.

[12] L. Mui, "Computational models of trust and reputation:agents, evolutionary games, and social networks. ph.d. thesis," Massachusetts Institute of Technology, Tech. Rep., 1995.

[13] J. Barbalet, "A characterization of trust, and its consequences," *Theory and Society*, vol. 38, pp. 367–382, 2009. [Online]. Available: http://dx.doi.org/10.1007/s11186-009-9087-3

[14] C.-N. Ziegler and G. Lausen, "Spreading activation models for trust propagation," in *IEEE International Conference on e-Technology,e-Commerce, and e-Service*, 2004, pp. 83–97.

[15] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *Proceedings of the 13th international conference on World Wide Web*, ser. WWW '04. New York, NY, USA: ACM, 2004, pp. 403–412. [Online]. Available: http://doi.acm.org/10.1145/988672.988727

[16] A. Josang, R. Hayward, and S. Pope, "Trust network analysis with subjective logic," in *ACSC '06: Proceedings of the 29th Australasian Computer Science Conference*. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2006, pp. 85–94.

[17] K. Thirunarayan, D. Althuru, C. Henson, and A. Sheth, "A local qualitative approach to referral and functional trust," in *Proceedings of the The 4th Indian International Conference on Artificial Intelligence (IICAI-09)*, 2009.

[18] E. Chang, E. Damiani, and T. Dillon, "Fuzzy approaches to trust management," in *Computational Intelligence, Theory and Applications*, B. Reusch, Ed. Springer Berlin Heidelberg, 2006, pp. 425–436. [Online]. Available: http://dx.doi.org/10.1007/3-540-34783-6_43

[19] P. Massa and P. Avesani, "Trust metrics on controversial users: Balancing between tyranny of the majority and echo chambers," in *International Journal on Semantic Web and Information System 3, no. 1*, 2007.

[20] trustlet.org, "Extended epinions dataset. www.trustlet.org," 2009.

[21] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*. American Mathematical Society, 1980.

[22] D. Koller and N. Friedman, *Probabilistic Graphical Models; Principles and Techniques*. The MIT Press, 2009.