Leveraging Community Detection for Accurate Trust Prediction

Ghazaleh Beigi
Department of Computer Engineering
Sharif University of Technology
ghbeigi@ce.sharif.edu

Mahdi Jalili
Department of Computer Engineering
Sharif University of Technology
mjalili@sharif.edu

Hamidreza Alvari
Department of EECS
University of Central Florida
halvari@eecs.ucf.edu

Gita Sukthankar
Department of EECS
University of Central Florida
gitars@eecs.ucf.edu

ABSTRACT

The aim of trust prediction is to infer trust values for pairs of users when the relationship between them is unknown. The unprecedented growth in the amount of online interactions on e-commerce websites has made the problem of predicting user trust relationships critically important, yet sparsity in the amount of known (labeled) relationships poses a significant challenge to the usage of machine learning techniques. This paper presents a community detection approach which leverages the network of available trust relations and rating similarities to compensate for the lack of labels. The key insight behind our framework is that trust values from the central community members can be used as a predictor for relationships between other community members. Here we evaluate the usage of two community detection algorithms, one of which works merely on the trust network while the other one uses both. Our algorithm outperforms other existing trust prediction methods on datasets from the well-known product review websites *Epinions* and *Ciao*.

I INTRODUCTION

Trust prediction, the ability to identify how much to trust to allocate an unknown user, is an important prerequisite toward the development of scalable online e-commerce communities. We are more likely to purchase an item from a seller on an e-commerce websites such as eBay or Amazon, if our trusted acquaintances have reported positive experiences with that seller in the past. Reviews from trusted users will carry more weight towards the purchasing decision than reviews from anonymous or unknown customers. Trust can be gained or lost through direct personal interactions, but this is impractical for popular e-commerce systems which boast millions of users. Thus, these platforms must support computational mechanisms for propagating trust between users. This problem is complicated by the fact that most customers only have interactions with a small set of other users and products, resulting in a sparse dataset of known trust relationships.

In this paper, we propose a novel community-based mechanism for propagating trust between users, even when they are not closely connected by existing links. The underlying assumption is that customer trust values are likely to be strongly correlated with other customers within the same community. Using community detection, users are grouped into non-exclusive communities (i.e., each user can be a member of several communities), which are represented by a prototypical highly-connected community member. Our model uses the community membership vector to infer trust values between two users by examining the similarities between the users and representative community members.

This paper introduces a two-phase approach to predict the trust values between each pair of users. In the first phrase, we cluster users into communities. This paper evaluates the usage of two different community detection algorithms: a game-theoretic approach (originally introduced in [1]) that operates under the assumption that users join communities to maximize their utility, which is calculated from a combination of rating similarity and the network neighborhood of known trust relations. For the second algorithm we use smart local moving (SLM) community detection [2] which detects communities by maximizing a modularity function. SLM is only designed to work on a single network, so we we run it on trust network only.

In the second phase, we predict the trust between each pair of users by comparing the similarities between their respective community membership vectors. To calculate the similarity between two communities, our community-based algorithm compares the central, or prototypical, community members. This paper evaluates the relative merits of different cen-

©ASE 2012

trality measures (*Betweenness*, *Eigenvector*, *MaxDegree*, *MaxTrustor* and *MaxTrustee*) in selecting community centers. These central members are then used to determine the similarity between the communities; communities with similar central users are assumed to be similar to one another. Our aim is to find the pair of communities from the users' community membership vectors that are both 1) similar to each other and 2) are a good match for the users (i.e., the users are themselves similar to the central member of the community). Intuitively, cases where both users belong to the same community will often have the highest match score, since two identical community centers will have the highest possible similarity score.

The paper concludes with a comparison between our community-based trust prediction method, a set of commonly used trust prediction heuristics, and hTrust (a low rank matrix factorization approach) [3].

II RELATED WORK

Bootstrapping trust between users is a general problem in many e-commerce platforms; it is useful to have a method to infer the trust value between two users before collecting a substantial amount of interaction data. Skopic et al. described two general approaches for initializing trust values between users, mirroring and teleportation [4].

Trust prediction can be framed as a supervised [5, 6] or an unsupervised [7, 8] learning problem. Unlike many other classification problems, it is easy to obtain labels for trust prediction since any known link serves as a positive training instance for binary classification; however, these approaches need to compensate for the extremely imbalanced datasets. Unsupervised methods are capable of inferring trust values even for indirectly connected users, but can also suffer from the sparsity of known trust relations. Ma et al. extracted features from writer-reviewer interactions and employed them in cluster-based classification methods [9]. Their method clusters users which are then used to train a personalized trust classifier for each user. Sherchan et al. proposed a five-state temporal Hidden Markov Model for predicting reputation where each state was represented by four hidden factors [10].

One of the earliest works on trust prediction was done by Golbeck [11] who defined properties of trust such as transitivity, composability and asymmetry while also introducing a number of algorithms for inferring binary and weighted trust values based on a specific propagation model. In [12], Kuter and Golbeck proposed a sampling method to estimate confidence values in the trust information.

An efficient trust propagation algorithm was introduced in [13]. The algorithm computes a weighted average and assigns it to a certain sink by removing untrustworthy members whose trust ratings fall below a threshold. Guha et al. introduced four atomic trust and distrust propagation primitives based on matrix operations; their trust inference algorithm was able to deal with the large numbers of iterations required to propagate trust through a large graph [8].

One area of particular research interest is trust prediction for consumer data (e.g., [14] and [15]). Noor and Sheng compute trustworthiness as a sum of feedbacks weighted by their trust credibilities, which in turn are calculated based on feedback density and majority consensus [14].

In this paper, we compare our work to Tang et al. who formulated trust prediction as an optimization problem. [3]. The authors first demonstrated the existence of homophily in trust relations and then used homophily regularization to exploit the effect. Their method, hTrust, uses low-rank matrix factorization and homophily regularization for unsupervised trust prediction.

III TRUST PREDICTION MODEL

To perform trust prediction, our algorithm first extracts and then compares users' community membership profiles. We compare the performance of two community detection approaches for generating the membership vector: game-theoretic [1] and smart local moving (SLM) [2].

1 COMMUNITY DETECTION

1.1 GAME-THEORETIC

Suppose that we have a graph $G = (V, E)$, with $n = |V|$ vertices and $m = |E|$ edges representing the sparse trust relationship network data T. Further, suppose that there exists rating relationship network data R consisting of users and items and the ratings that users have given to the items. Following the work described in [1, 16], we consider the process of community detection as an iterative game performed in a multi-agent environment in which each node of the underlying graph is a selfish agent who decides to maximize its total utility u_i. This process can be simulated using an agent-based model that seeks to detect communities by optimizing each user's utility through a stochastic search process. For calculating the utility function, we examine the contribution of two factors, trust similarity (T_{ij}) and rating similarity (R_{ij}), toward community detection.

During the game, each agent can periodically take an action (*join*, *switch*, *leave* and *no operation*) to modify or retain the labels of communities that it belongs to, based on its current utility. The set of all such communities is denoted by $[k] = 1, 2, \ldots, n$. We define a strategy profile $S = (s_1,\ s_2,\ ...,\ s_n)$ which represents the set of all strategies of all agents, where $s_i \subseteq [k]$ denotes the strategy of agent i, i.e. the set of its labels.

In our framework, the best response strategy of an agent i with respect to strategies S_{-i} of other agents is calculated as: $\arg\max_{s_i \subseteq [k]} u_i(S_{-i}, s_i)$. We consider a linear function of T_{ij} and R_{ij} as the gain function of each agent, where $\alpha \in [0, 1]$:

$$g_i(S_{-i}, s_i) = \frac{1}{m} \sum_{l \in s_i} \sum_{j \in l} (\alpha T_{ij} + (1 - \alpha) R_{ij}). \qquad (1)$$

As in real life, joining communities always has expenses (e.g. fees), so here we also consider loss function l_i for each agent, which is linear in the number of labels each agent has:

$$l_i(S_{-i}, s_i) = \frac{1}{m}(|s_i| - 1). \qquad (2)$$

Therefore the utility function for each agent is calculated by:

$$u_i(S_{-i}, s_i) = g_i(S_{-i}, s_i) - l_i(S_{-i}, s_i). \qquad (3)$$

The strategy profile S forms a pure Nash equilibrium of the community formation game if all agents play their best strategies.

For calculating the similarities between each pair of vertices in G, we can use local or global properties, regardless of whether or not the nodes are directly connected. In this work we use separate similarity measures for the two halves of the utility function. For the first half, we use neighborhood similarity [1] with different normalization factors to quantify trust similarity between users:

$$T_{ij} = \begin{cases} w_{ij}(1 - d_i d_j / 2m) & A_{ij} = 1, w_{ij} >= 1 \\ w_{ij}/n & A_{ij} = 0, w_{ij} >= 1 \\ d_i d_j / 2m & A_{ij} = 1, w_{ij} = 0 \\ -d_i d_j / 2m & A_{ij} = 0, w_{ij} = 0 \end{cases} \qquad (4)$$

where w_{ij} is the number of common neighbors node i and j have and d_i is the degree of node i. T_{ij} assumes its highest value when two nodes have at least one common neighbor and are also directly connected, i.e. $A_{ij} = 1$.

To evaluate the value of rating similarity between users, we calculate the *cosine similarity* over the ratings using:

$$R_{ij} = \frac{\sum_k r_{ik} r_{jk}}{\sqrt{\sum_k r_{ik}^2} \sqrt{\sum_k r_{jk}^2}} \qquad (5)$$

where vectors r_i and r_j are rating vectors for user i and user j, respectively.
Algorithm 1 shows our proposed framework. After calculating trust similarities between each pair of agents (Equation 4) and rating similarity (Equation 5), the multi-agent game commences. The community structure of the network emerges after agents reach the local equilibrium.

1.2 SMART LOCAL MOVING (SLM)

The smart local moving algorithm (SLM) detects communities in networks by maximizing a modularity function; nodes are repeatedly transferred between communities in such a way that each movement causes an increase in modularity [2]. In more detail, the local moving heuristic iterates over the nodes in random order and checks whether the modularity increases by moving that node from its current community to another one. This process continues until no more movement is possible (Algorithm 2).

2 TRUST PREDICTION

Once we have extracted the communities, we select a representative (central) user from each community. In this paper, we evaluate the usage of different measures for selecting this representative user:

1. *Betweenness*
2. *Eigenvector*

©ASE 2012

Algorithm 1 Game-theoretic based trust predictor

```
Input: *trust* and *rating* networks
Output: Predicted trust values
Calculate trust similarities $T_{ij}$ between pairs of users with trust relations
Calculate rating similarities $R_{ij}$ between pairs of users' rating vectors
while $NOT$ convergence in the agents' utilities do
    Iterate over agents
    Iterate over actions (join, switch, leave and no action)
    Calculate the change in agent utility resulting from the action
    if change exceeds a threshold then
        Execute action
        Update communities
    end if
end while
Detect centers of communities
Iterate over all possible pairs of users $(i, j)$ without trust relations
Find the labels of communities which agent $i$ and agent $j$ belong to
Calculate predicted trust values based on equation 6
```

Algorithm 2 SLM based trust predictor

```
Input: *trust* and *rating* networks
Output: Predicted trust values
Calculate rating similarities $R_{ij}$ between pairs of users' rating vectors
SLM(*trust*)
Detect centers of communities
Iterate over all possible pairs of users $(i, j)$ without trust relations
Find the labels of communities which user $i$ and user $j$ belong to
Calculate predicted trust values based on equation 6
```

3. *MaxDegree*
4. *MaxTrustor*
5. *MaxTrustee*
6. *Random*.

These centrality measures are calculated using functions from the JUNG package[1] on the community subgraphs. A high betweenness scores indicates that a node lies on a large number of geodesics within the subgraph. Eigenvector centrality for each node is defined as the proportion of time that a random walker will visit that node over the time horizon. Max degree selects the node with the highest overall degree, and max trustor/trustee treat the in degree and out degree separately. We compare these centrality methods against a baseline in which the central community node is randomly selected.

This prototypical user is then treated as being the center of the community for the purposes of measuring similarities between users. After detecting centers for all communities, we calculate the rating similarity $R_{ic_l^i}$ (Equation 5) between rating vectors of user i and each of the centers c_l^i of all labels that it belongs to, where $l \in s_i$. We repeat this process for user j and their corresponding centers. We also maintain a list of rating similarities between all the community centers. The final trust value between users is the maximum over the possible average values of these numbers:

$$P_{ij} = \max_{c_i \in c_{s_i}, c_j \in c_{s_j}} Avg\{R_{ic_l^i}, R_{jc_l^j}, R_{c_l^i c_l^j}\} \tag{6}$$

The aim of this process is to find the pair of centers that are both 1) similar to each other and 2) similar to the users themselves.

IV EXPERIMENTS

We use the Epinions and Ciao datasets[2] to evaluate our method. First, the datasets are preprocessed by eliminating users with less than two trustors and items with less than two available ratings. Table 1 gives the statistics of the datasets after filtering. Also, trustor and trustee distributions for both datasets are shown in Figure 1 and Figure 2 respectively.

Following the evaluation in [3], we choose $(100 - x)\%$ of the pairs of users with known existing trust relations as the trust relations N to predict and remove

[1] http://jung.sourceforge.net/

[2] http://www.public.asu.edu/$\sim$jtang20/datasetcode/truststudy.htm

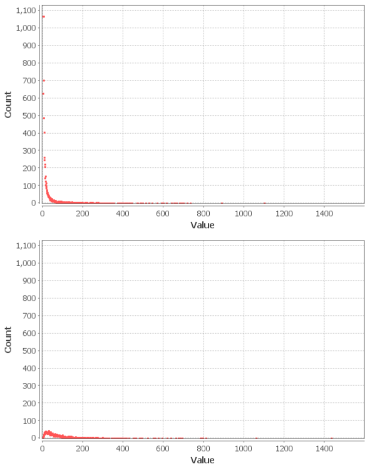


Figure 1: Trustor (top) and trustee (bottom) distributions for the Epinions dataset

	Epinions	Ciao
# Users	9,497	5,329
# Items	114,983	56,134
# Ratings	367,741	198,230
# Trust Relations	321,305	106,388
Max # of Trustors	1,047	98
Max # of Trustees	1,432	780
Avg. Degree	21.667	19.964
Trust Network Density	0.004	0.004
Avg. Clustering Coefficient	0.154	0.153

Table 1: Statistics of datasets after filtering

their trust values by setting $G(i, j) = 0$. The new representation of G is fed to each predictor. x is varied as $\{50, 60, 70, 80, 90\}$. We then use prediction accuracy (PA) [17] to report the performance of the predictors. More specifically, each predictor ranks the pairs of $B \cup N$ in decreasing order, where B is the randomly chosen subset of pairs of users with unknown trust relation with size equal to $4 * |N|$. The final set of predicted trust relations, T, is the first N pairs in the sorted list. Finally we compare T with set N to see how many pairs are predicted correctly. Hence we have the following equation:

$$PA = \frac{|N \cap T|}{|N|} \tag{7}$$

Since we select the users in B randomly, we report the final results by taking the average of 10 runs for each method.

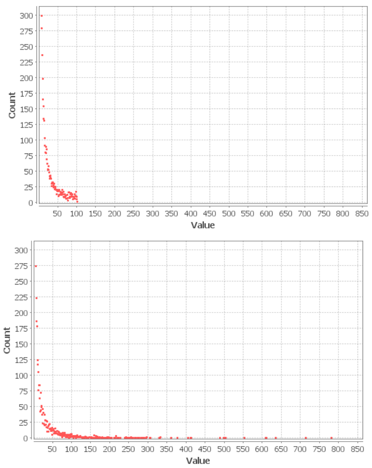


Figure 2: Trustor (top) and trustee (bottom) distributions for the Ciao dataset

1 RESULTS

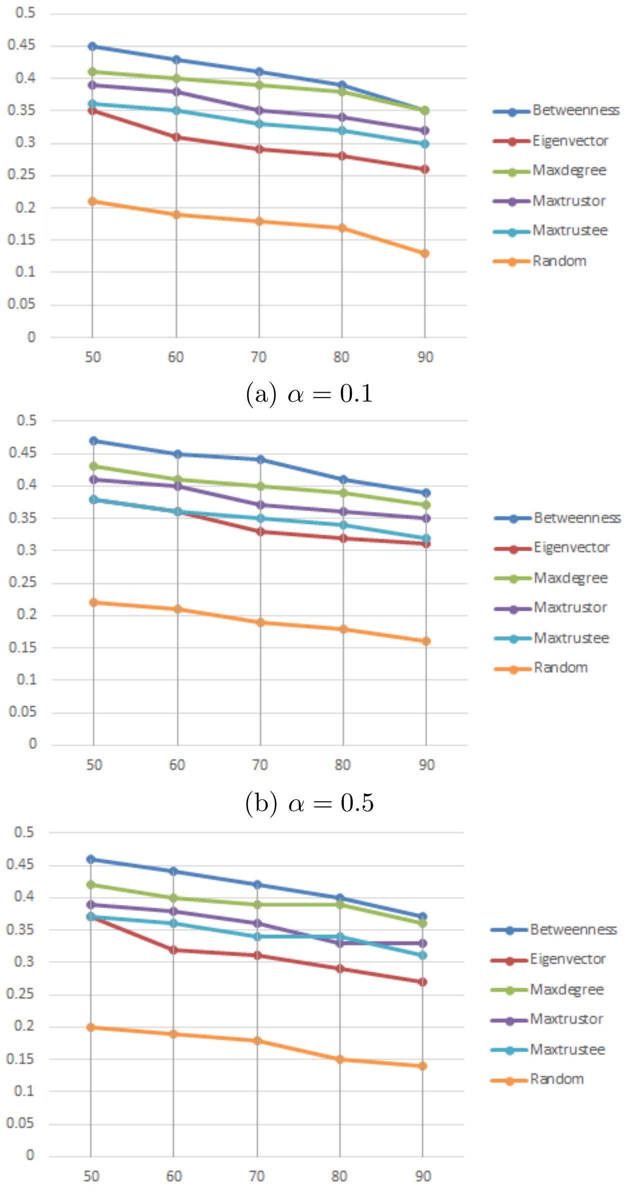
This section presents results on the performance of different variants of our proposed trust prediction framework: 1) the usage of game-theoretic vs. SLM community detection methods and 2) different centrality measures for identifying community centers. The results of the game-theoretic trust predictor with $\alpha = \{0.1, 0.5, 0.9\}$ are shown in Figures 3 and 4; those of the SLM based trust predictor are shown in Figure 5. Then, we compare our framework against a set of baselines:

- hTrust: Infers trust values using low-rank matrix factorization and homopily regularization [3].
- RS: Ranks the pairs of users based on Cosine similarity (Equation 5).
- JC: Ranks the pairs of users based on Jaccard similarity:

$$R_{ij} = \frac{|I(i) \cap I(j)|}{|I(i) \cup I(j)|} \tag{8}$$

  where $I(i)$ refers to the set of items user i has rated. Jaccard similarity counts the total number of unique items that user i and user j have rated.
- Random: Ranks the pairs of users after assigning random values to each of them.

Based on these experiments, we make the following observations. The game-theoretic version of our community based trust prediction outperforms the use of SLM for community detection. In both the game-theoretic and SLM community detection approaches (all conditions), betweenness is the best method for

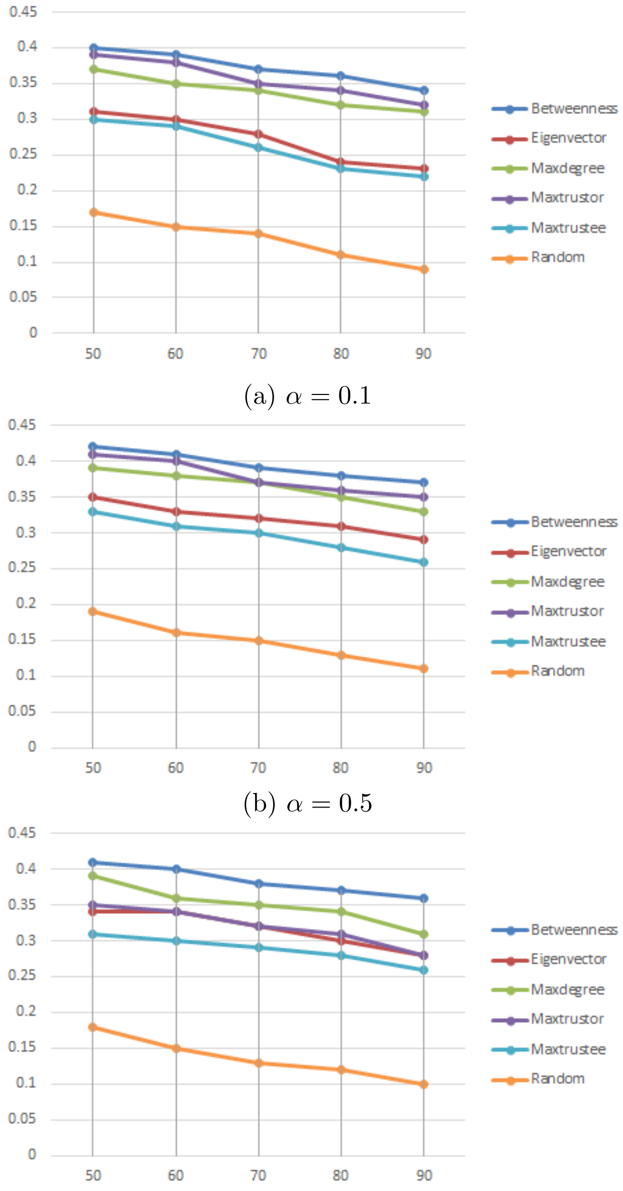


(a) $\alpha = 0.1$

(b) $\alpha = 0.5$

(c) $\alpha = 0.9$

Figure 3: Prediction accuracy of game-theoretic community detection variant vs. training dataset size (x) on the Epinions dataset

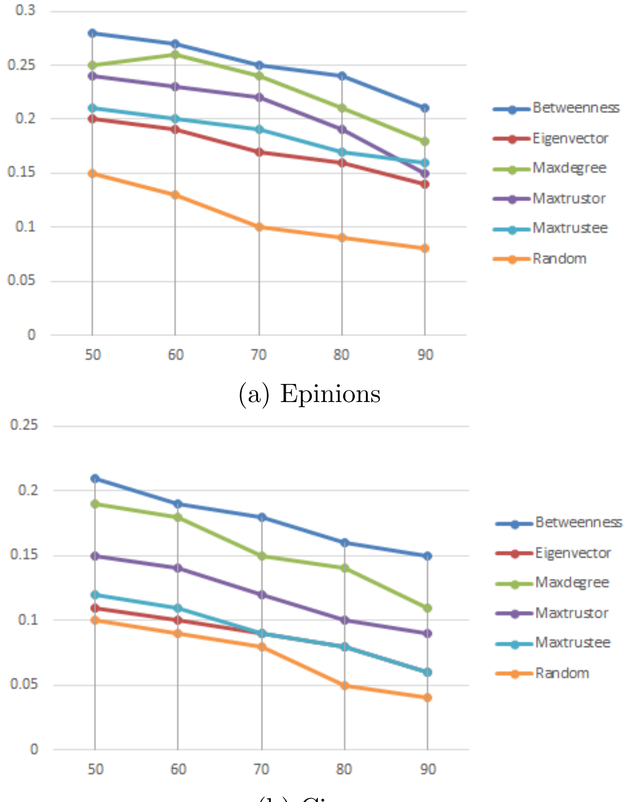


(a) $\alpha = 0.1$

(b) $\alpha = 0.5$

(c) $\alpha = 0.9$

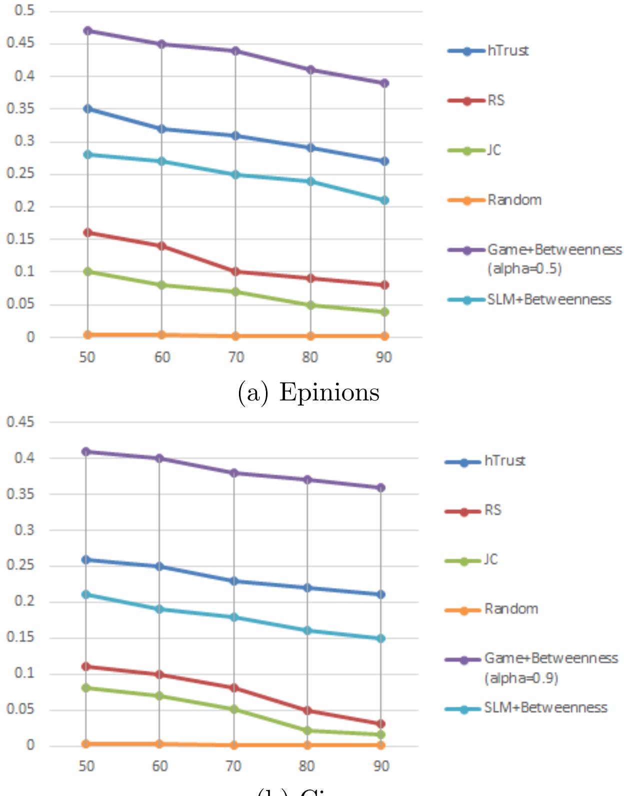
Figure 4: Prediction accuracy of game-theoretic community detection variant vs. training dataset size (x) on the Ciao dataset



(a) Epinions

(b) Ciao

Figure 5: Prediction accuracy of SLM community detection variant vs. training dataset size (x)



(a) Epinions

(b) Ciao

Figure 6: Prediction accuracy of the baseline methods vs. training dataset size. The baseline methods are compared with our proposed community-based trust prediction methods (both game-theoretic and SLM community detection with community centers selected by betweeness). The game-theoretic version of our method is the top performing approach on both datasets.

identifying community centers, followed by MaxDegree and MaxTrustor. Different values of α seem to have minimal impact on the performance of our utility function in game-theoretic community detection. In the comparison against other baselines, the game-theoretic version outperforms hTrust, the strongest baseline method, and the SLM version outperforms the other heuristics (but not hTrust). Increasing the training data set size paradoxically leads to small decreases in prediction accuracy; this phenomenon is described in greater detail in [3].

V CONCLUSION

This paper presents a community detection based approach for bootstrapping trust prediction on product review websites. The intuition behind our method is that comparing rating similarities between communities is more robust than comparing ratings between individuals. First, communities are detected using both the trust and rating networks. Second, community centers are identified using centrality measures to find representative users. Finally, trust prediction is performed by selecting corresponding communities from the users' membership vectors that 1) are similar to each other and 2) match the users well, as measured by similarity between the users and the community centers. Here we demonstrate that the game-theoretic version of our proposed method outperforms a set of baseline trust prediction methods. For the next part of our research agenda, we plan to explore alternate distance metrics for measuring distances between users.

VI ACKNOWLEDGMENTS

This research was supported in part by NSF IIS-08451. The authors acknowledge the University of Central Florida Stokes Advanced Research Computing Center for providing computational resources and support that have contributed to results reported herein. URL: `http://webstokes.ist.ucf.edu`.

References

[1] H. Alvari, S. Hashemi, and A. Hamzeh, "Detecting overlapping communities in social networks by game theory and structural equivalence concept," *Artificial Intelligence and Computational Intelligence*, pp. 620–630, 2011.

[2] L. Waltman and N. J. van Eck, "A smart local moving algorithm for large-scale modularity-based community detection." *CoRR*, vol. abs/1308.6604, 2013.

[3] J. Tang, H. Gao, X. Hu, and H. Liu, "Exploiting homophily effect for trust prediction." in *WSDM*. ACM, 2013, pp. 53–62.

[4] F. Skopik, D. Schall, and S. Dustdar, "Start trusting strangers? Bootstrapping and prediction of trust," in *WISE*, 2009, pp. 275–289.

[5] H. Liu, E.-P. Lim, H. W. Lauw, M.-T. Le, A. Sun, J. Srivastava, and Y. A. Kim, "Predicting trusts among users of online communities: an Epinions case study," in *ACM Conference on Electronic Commerce*, 2008, pp. 310–319.

[6] V.-A. Nguyen, E.-P. Lim, J. Jiang, and A. Sun, "To trust or not to trust? Predicting online trusts using trust antecedent framework." in *ICDM*. IEEE Computer Society, 2009, pp. 896–901.

[7] P. Borzymek, M. Sydow, and A. Wierzbicki, "Enriching trust prediction model in social network with user rating similarity," in *CASoN*, 2009, pp. 40–47.

[8] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of Trust and Distrust," in *WWW*, 2004, pp. 403–411.

[9] N. Ma, E.-P. Lim, V.-A. Nguyen, A. Sun, and H. Liu, "Trust relationship prediction using online product review data," in *CIKM-CNIKM*, 2009, pp. 47–54.

[10] W. Sherchan, S. Nepal, and A. Bouguettaya, "A trust prediction model for service web," in *IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2011, pp. 258–265.

[11] J. Golbeck, "Computing and applying trust in web-based social networks," Ph.D. dissertation, University of Maryland, College Park, 2005.

[12] U. Kuter and J. Golbeck, "SUNNY: A new algorithm for trust inference in social networks using probabilistic confidence models." in *AAAI*, 2007, pp. 1377–1382.

[13] P. Massa and P. Avesani, "Controversial users demand local trust metrics: An experimental study on epinions.com community." in *AAAI*, 2005, pp. 121–126.

[14] T. H. Noor and Q. Z. Sheng, "Credibility-based trust management for services in cloud environments," in *ICSOC*, 2011, pp. 328–343.

[15] J. K. Sinclaire, R. B. Wilkes, and J. C. Simon, "A prediction model for initial trust formation in b2c ecommerce." in *AMCIS*, 2009, p. 507.

[16] H. Alvari, K. Lakkaraju, G. Sukthankar, and J. Whetzel, "Predicting guild membership in massively multiplayer online games," in *SBP*, Washington, D.C., April 2014.

[17] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

©ASE 2012