Rigorous probabilistic trust-inference with applications to clustering

Thomas DuBois, Jennifer Golbeck, Aravind Srinivasan
University of Maryland
College Park, Maryland, U.S.A.
{*tdubois,golbeck,srin*}*@cs.umd.edu*

Abstract

The World Wide Web has transformed into an environment where users both produce and consume information. In order to judge the validity of information, it is important to know how trustworthy its creator is. Since no individual can have direct knowledge of more than a small fraction of information authors, methods for inferring trust are needed. We propose a new trust inference scheme based on the idea that a trust network can be viewed as a random graph, and a chain of trust as a path in that graph. In addition to having an intuitive interpretation, our algorithm has several advantages, noteworthy among which is the creation of an inferred trust-metric space *where the shorter the distance between two people, the higher their trust. Metric spaces have rigorous algorithms for clustering, visualization, and related problems, any of which is directly applicable to our results.*

1. Introduction

There are two important entities on the web: people and information. As the web has transformed into an interactive environment filled with billions of pages of user-generated content, trust becomes a critical issue. When interacting with one another, users face a wide range of risks. Similarly, information provided by users can be overwhelming because there is so much of it and because it is often contradictory. Fortunately, as trust has become a concern, potential solutions have also emerged.

Social networks on the web are a major phenomenon. This large corpus of publicly-accessible relationship information has the potential to transform the way intelligent systems on the web are built. The trust relationship is particularly powerful since it speaks directly to the "quality" of a person and what they produce online.

In a large network, a given user likely knows only a small fraction of the people with whom he or she will interact; thus, the user has no knowledge of how trustworthy most people are. To handle this, methods are needed for inferring trust between users who do not know one another directly. We present a novel way of interpreting trust networks that leads to an immediate method for taking local trust values and computing implicit trust between all pairs of nodes, including those who have no direct knowledge of each other's trustworthiness. Our approach also leads rigorously to a metric space among the users, with closer pairs corresponding to higher trust-values; this naturally leads to efficient algorithms for clustering the population.

There are a number of prior algorithms for inferring trust in social networks. To comply with space constraints we do not review these algorithms here, but a full treatment can be found in [1]. Our approach is quite different from the methods developed in the literature so far, for its probabilistic treatment of trust, its integrated notions of trust and confidence, and the algorithm itself. Also see Andersen et al. [2] for an axiomatic approach to a different type of trust-inference problem, where the initial trust votes are "-" or "+".

The idea that trust networks can be treated as random graphs drives our work. For every pair of nodes (u, v), we place an edge between them with some probability that depends on the direct trust value between them which we denote by $t_{u,v}$. We then infer trust between two people from the probability that they are connected in the resulting graphs. Formally we choose a mapping f from trust value to probabilities. We then construct a random graph G in which each edge (u, v) exists independently with probability $f(t_{u,v})$. We then use this graph to generate inferred trust values, $T_{u,v}$, such that $f(T_{u,v})$ equals the probability that there is a path from u to v in the random graph.

A very intuitive idea motivates this model. Consider the following scenario: Alice knows Bob and thinks he has an $f(t_{a,b})$ chance of being trustworthy. Bob knows Eve and thinks she has a $f(t_{b,e})$ chance of being trustworthy, and he tells this to Alice if he is trustworthy. If Bob is not trustworthy, he may lie about p_e and give any value to Alice. Alice reasons that Eve is trustworthy if Bob is trustworthy and gives her the correct value $f(t_{b,e})$ and Eve is trustworthy with respect to Bob. This combination happens with probability $f(t_{a,b})f(t_{b,e}) = f(T_{a,e})$ if Bob's trustworthiness and Eve's trustworthiness are independent. Thus we view a path through the network as a Bayesian chain. To illustrate this view, define X_B, X_E to be the random events that Bob and Eve are trustworthy from Alice's

perspective. This gives the formula:

$$Pr[X_E] = Pr[X_E|X_B] \cdot Pr[X_B] + Pr[X_E|\overline{X_B}] \cdot Pr[\overline{X_B}]$$
$$\geq Pr[X_E|X_B] \cdot Pr[X_B] = Pr[X_B \wedge X_E].$$

The same analysis can be used if trust is a proxy for similarity: Alice and Bob's mutual trust can be a measure of how similar they are. If trust is interpreted as a probability of being in the same category, then Alice's category is the same as Eve's if (but not necessarily only if) Alice and Bob share a category and Bob and Eve share a category.

In large, complex networks the Bayesian chain view still applies. If there exists a path from Alice to Eve in a random network constructed from trust values, then that path is a chain of people from Alice to Eve who each correctly trust their successor, and Alice can trust Eve. Therefore Alice trusts Eve with the probability that there is a path from Alice to Eve in the random graph. Since it is $\#P-$Complete [3] to compute connectivity probabilities exactly, we rely on random sampling. If the true connectivity probability between Alice and Eve is p and we sample the graph k times, then kp of them will contain an Alice to Eve path in expectation. We then apply Chernoff bounds which show that when k is reasonably large, our sampled value will be very close to the actual value kp. In fact, for any $\epsilon > 0$, if we take $k = \Theta(\frac{\log n}{\epsilon^2})$ samples, then for any pair u, v the probability that our estimate is off by more than ϵ is at most $e^{-\Theta(\epsilon^2 \log n/\epsilon^2)} = n^{-\Omega(1)}$. We then take a union bound over all pairs to bound the probability that any pair deviates by more than ϵ. If we take as few as $5 \log n/\epsilon^2$ samples this probability is at most n^{-3} for each pair. Then taking a union bound over all pairs shows that with probability at least $1 - \frac{1}{n}$, $T_{u,v}$ will be within ϵ of the true value for *every* pair u, v simultaneously.

In addition to having an intuitive motivation, our algorithm is also novel within the area of trust inference in the extent to which it allows us to make use of established algorithms in graph and clustering theory. Because of the graph-theoretic nature of the algorithm, we can make use of the probabilistic method as well as theory of random graphs pioneered by Erdös and Rényi [4] and heavily studied since then. Additionally, because our algorithm defines a metric space on the people in a trust network – as demonstrated in Section 3 – we obtain the flexibility and utility of a variety of metric-clustering algorithms that we can apply.

2. Illustrative Examples

In Figure 1 we introduce a few small example graphs to demonstrate some of the desirable qualities that our path probability formulation exhibits. In these examples trust is symmetric, however it could just as easily be asymmetric.

In our first example, the graph consists of two cliques connected by a single edge. Since any path from one clique to the other includes this edge, the trust between any two nodes in different cliques is bounded by this edge's trust.

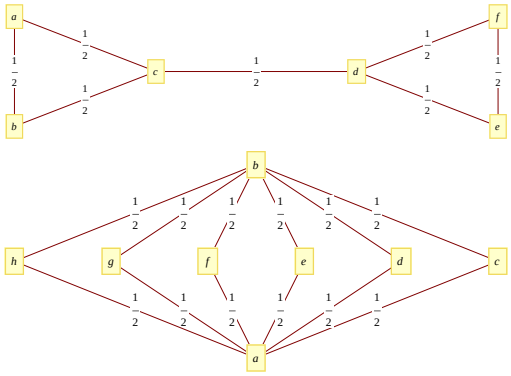

Figure 1. These figures show important trust properties.

In our second example, the nodes a and b have no direct trust, instead they are connected through a sequence of common neighbors. If the trust between neighbors is uniformly p, then each path connecting a and b occurs with probability p^2, and the cumulative path probability is $1 - (1 - p^2)^k \approx 1 - e^{-p^2 k}$. This can be close to 1 even when p is low. Intuitively this corresponds to Alice having many acquaintances who also know a little about Eve's trustworthiness. They all vouch for Eve a little bit and collectively provide a strong link from a to b.

3. Additional Benefits

The function $d(u, v) = \log \frac{1}{f(T_{u,v})}$ defines a metric space (or an asymmetric metric space) on the nodes. This holds primarily because $f(T(u, v)) \geq f(T(u, w)) \cdot f(T(w, v))$, and taking logs of one divided by these terms gives $d(u, v) \leq d(u, w) + d(w, v)$. A metric space on the nodes allows us to make use of existing metric clustering algorithms to partition the nodes into groups. A clustering algorithm takes a set of points in a metric space and groups them in a way that tries to optimize some criteria. Examples that have good approximation algorithms include, k-centers [5], [6], k-means [7], and correlation clustering [8], [9].

Another major analytical benefit of our algorithm involves the identification of key edges. A quick inspection of Figure 1's first graph shows that the edge (c, d) is in some sense critical in that removing it would drastically alter some of the distances in the graph. Our technique gives a simple way of quantifying the importance of such edges.

For each trust edge, we define its criticality $c_{u,v}$ as the difference between the inferred trust $T_{u,v}$, and what the inferred trust would be without the edge (u, v), which we denote by $T'_{u,v}$. Criticality measures how important a direct relationship is. A redundant edge's criticality is small, and can be weakened or removed without changing the graph

distances much. Conversely, if the criticality $c_{u,v}$ is large, most paths from u to v require the edge u, v.

We make use of the probabilistic details of our trust estimates to efficiently determine criticality for all edges. We only need to acquire one set of estimates on the $T_{u,v}$, and we can directly compute corresponding $T'_{u,v}$ values. The edge (u, v) is included in the random graph with probability $f(t_{u,v})$, we denote this event by $E_{u,v}$. $E_{u,v}$ is independent of the event that any other path from u to v exists, which we denote by $P_{u\to v}$. We can compute the criticality $c_{u,v}$ by:

$$\begin{aligned} f(T_{u,v}) &= Pr[E_{u,v} \vee P_{u\to v}] \\ &= Pr[E_{u,v}] + Pr[P_{u\to w} \wedge \overline{E_{u,v}}] \\ &= f(t_{u,v}) + f(T'_{u,v})(1 - f(t_{u,v})) \\ c_{u,v} = T_{u,v} = T'_{u,v} &= T_{u,v} - f^{-1}\left(\frac{f(T_{u,v}) - f(t_{u,v})}{1 - f(t_{u,v})}\right). \end{aligned}$$

4. Application to Existing Datasets

We used two social networks with trust values as test networks: the Trust Project network [10] and the FilmTrust social network [11]. In both networks, we selected the giant component and removed nodes with a degree of 1. This left 330 nodes with 1,059 edges in the FilmTrust network. In the Trust Project network, the same filter left 62 people and 177 edges.

Both of these networks contain directed edges with asymmetric trust values. However, either symmetric or asymmetric trust relationships can occur in real networks. For the purpose of our experiments, we worked with both the directed graph and with a version where we converted the networks to undirected graphs with symmetric trust values.

4.1. Symmetric Trust

Our datasets are inherently asymmetric, each trust value comes from one person rating the other. Since in both datasets trust is a type of similarity measure, we resolve differing trust values by taking their mean. We then address two issues with our choice of the function f. First, people appear to be biased towards giving high trust, treating low values as negative trust, so we bias our function to compensate. Second, someone could become one of the most trusted nodes in the graph by rating many others regardless of the ratings others give them. To compensate we truncate the amount of outgoing trust for any node at 5 times the maximum individual trust. The choice of 5 was fairly arbitrary, though the choice of a small constant is motivated by the work of Erdös and Rènyi which showed that a random graph with more than one expected edge per node is likely to have a giant component.

We show the largest component of our leftmost dataset in Figure 2. We tried many different mappings from trust to probabilities, and most yielded similar results. Looking

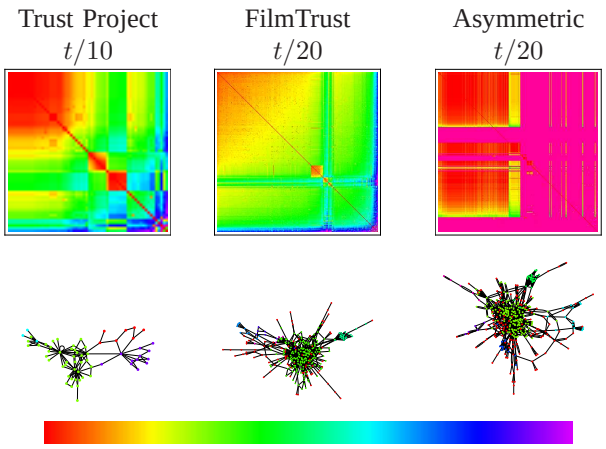

Figure 2. The top data row gives the function f. The top figures show distances between all pairs of nodes. The distance from u to v is given by the color from row u column v. The distance grids are sorted to highlight clusters. The middle figures show a clustering based on the trust metric. The bottom figure is the key for the grids, distances increase linearly from 0 at red to 10 or greater at violet.

at the graphs and the metric distance grids, you can pick out some of the natural groups. Specifically, there are three mostly red blocks (indicating high mutual trust) along the diagonal in the grid that correspond to the main clusters. Notice that while the distances change, the two main clusters are fairly robust to changes in the function f.

Next we examine the FilmTrust dataset. The middle column of Figure 2 breaks down our results similarly to the previous dataset. It is dominated by a single, highly connected cluster. Yet our algorithm is still able to identify a few isolated groups, as well as which nodes within the cluster are loosely connected enough to be separate from the core.

4.2. Asymmetric Trust

When trust is asymmetric, all of the same fundamentals apply. However there are a few noticeable differences. First, we need a much richer graph. In the symmetric case, a large connected component is enough to make the problem interesting. However with asymmetric trust, we can have a situation where there are no non-trivial paths. Second, in the symmetric case a person who rates everyone else can become the most trusted node, so we truncate total outgoing trust. This is unnecessary in the directed case. Because of these reasons, the smaller dataset is not particularly interesting, and we will not examine it in detail.

In the rightmost column of Figure 2 we display the asymmetric FilmTrust dataset. The distance grid shows one large

mutually trusting group, as well as several progressively smaller mutually trusting groups. The largest of the groups is trusted by a large portion of the network. It also shows a secondary group that is well trusted by this largest group.

5. Applications of Clustered Networks

A clustering of a network is a partition of the nodes into meaningful groups. A good clustering will identify groups of nodes where a node is more similar to the others in its cluster than to those in other clusters. There are many ways to find a "good" clustering. We generally use a correlation clustering algorithm which minimizes the sum of distances within groups and maximizes the sum of distances between groups. This seems well suited to the trust domain, but, based on the particular needs of an application, any clustering algorithm over a metric space can be applied.

Once a network has been clustered, there are a number of interesting applications. First, visualizing large networks is difficult, as is identification of important groups within them. A clustering algorithm that groups similar, trusted individuals together can be used to display the network and support visual analysis. In addition, some applications use trust as a background for other operations. For example, trust-based recommender systems (e.g. [12]) could use clusters to limit the search space and optimize the items shown to users.

6. Conclusions

Trust is an important issue in the type of large social networks available on the World Wide Web. It helps us estimate the quality of people and the information they produce, which in turn helps us to filter or validate that information. Since these networks are often so large that no one knows more than a small fraction of the other people, direct trust has limited usefulness. To overcome this, trust inference algorithms have been proposed.

We present a novel trust inference algorithm based on the intuitive idea that a trust network can correspond to a random graph where an edge from a to b occurs with a probability that is a function of a's direct trust in b. If we interpret the graph so that an edge in the graph from a to b meaning that a was correct to trust b, then we infer that c can trust d if there is any path in the graph from c to d.

We show that this trust inference scheme leads to good results for inferred trust, and because of its basis in probability theory, it offers additional benefits as well. Perhaps the most important of these is the creation of a trust metric space on the people in the network where the closer together two people are, the greater the inferred trust between them. There are many, well studied, applications of metric spaces, including clustering and visualization. Any one of which can be used on the metric we produce, and we demonstrate the effectiveness of applying clustering to several real datasets.

7. Acknowledgment

Thomas DuBois and Aravind Srinivasan's contributions are supported in part by NSF ITR Award CNS-0426683 and NSF Award CNS-0626636. Part of Aravind's work was done while attending the DIMACS Conference on Probabilistic Combinatorics & Algorithms, DIMACS Center, Piscataway, NJ, USA, April 2006. Jennifer Golbeck's work was supported, in part, by a DARPA Seedling grant.

References

[1] J. Golbeck, "The science of trust on the web," *Foundations and Trends in Web Science*, 2008.

[2] R. Andersen, C. Borgs, J. T. Chayes, U. Feige, A. D. Flaxman, A. Kalai, V. S. Mirrokni, and M. Tennenholtz, "Trust-based recommendation systems: an axiomatic approach," in *WWW*, 2008, pp. 199–208.

[3] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal on Computing*, vol. 8, no. 3, pp. 410–421, 1979.

[4] P. Erdös and A. Rényi, "On the evolution of random graphs," in *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 1960, pp. 17–61.

[5] D. S. Hochbaum and D. B. Shmoys, "Best possible heuristic for the k-center problem," *Mathematics of Operations Research*, no. 2, pp. 180–184, May 1985.

[6] A. Archer, "Two O(log*k)-approximation algorithms for the asymmetric k-center problem," in *Proceedings of the 8th Conference on Integer Programming and Combinatorial Optimization*. Springer-Verlag, 2001, pp. 1–14.

[7] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "A local search approximation algorithm for k-means clustering," in *SCG '02: Proceedings of the eighteenth annual symposium on Computational geometry*. New York, NY, USA: ACM, 2002, pp. 10–18.

[8] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," in *Machine Learning*, 2002, pp. 238–247.

[9] N. Ailon, M. Charikar, and A. Newman, "Aggregating inconsistent information: Ranking and clustering," *Journal of the ACM*, vol. 55, no. 5, pp. 1–27, 2008.

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

[11] ——, "Generating predictive movie recommendations from trust in social networks," in *Proceedings of the Fourth International Conference on Trust Management*, 2006. [Online]. Available: http://trust.mindswap.org/papers/iTrust06.pdf

[12] J. O'Donovan and B. Smyth, "Trust in recommender systems," in *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*. New York, NY, USA: ACM, 2005, pp. 167–174.