SUNNY: A New Algorithm for Trust Inference in Social Networks Using Probabilistic Confidence Models

Ugur Kuter
Department of Computer Science and
Institute of Advanced Computer Studies,
University of Maryland, College Park,
College Park, MD 20742, USA
ukuter@cs.umd.edu

Jennifer Golbeck
College of Information Studies,
University of Maryland, College Park,
College Park, MD 20742, USA
jgolbeck@umd.edu

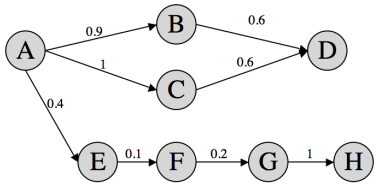

Figure 1: A sample social network with trust values (on a 0-1 scale) as edge labels.

Abstract

In many computing systems, information is produced and processed by many people. Knowing how much a user trusts a source can be very useful for aggregating, filtering, and ordering of information. Furthermore, if trust is used to support decision making, it is important to have an accurate estimate of trust when it is not directly available, as well as a measure of confidence in that estimate. This paper describes a new approach that gives an explicit probabilistic interpretation for confidence in social networks. We describe SUNNY, a new trust inference algorithm that uses a probabilistic sampling technique to estimate our confidence in the trust information from some designated sources. SUNNY computes an estimate of trust based on only those information sources with high confidence estimates. In our experiments, SUNNY produced more accurate trust estimates than the well known trust inference algorithm TIDALTRUST (Golbeck 2005), demonstrating its effectiveness.

Introduction

Trust is used in different ways in a variety of systems. *Social* trust is emerging as interesting and important, but it is as yet not well understood from a computational perspective. As with all social relationships, it is difficult to quantify trust and its properties are fuzzy. Still, developing methods for accurately estimating trust between people is important for the future of many systems; it holds promise for improving Web policy systems, as a component of provenance in scientific workflows and intelligence databases, and as a value for filtering, ordering, and aggregating data in many domains.

If trust is being used to support decisions, it is important to have an accurate estimate of trust when it is not directly available, as well as a measure of confidence in the computed trust value. Given a social network, existing inference algorithms either compute only trust, or conflate trust and confidence, yielding erroneous and ambiguous inferences. As an example, consider the network in Figure 1 which depicts a network with trust values on a 0 to 1 scale. When making a recommendation to A, one may be inclined to decrease the recommended trust value for H because it is derived from nodes who are not trusted much and the path from A to H is long. However, this is an incorrect approach since there is no way to tell if we are very certain that H should be trusted at that recommended level, or if we believe H is more trustworthy but simply do not have confidence in the sources of that information. The result of an algorithm that merges trust and confidence is not a trust value, but rather a new variable that is an amalgamation of the two concepts.

In this paper, we describe a new approach that gives an explicit probabilistic interpretation to confidence in social trust networks. Our contributions are as follows:

- A formal representation mapping that takes a trust network and produces a Bayesian Network suited for approximate probabilistic reasoning. This mapping also generates a probabilistic model for the result Bayesian Network, using similarity measures computed over the previous decisions of the entities in the input trust network.
- A new algorithm, called SUNNY, for trust inference based on probabilistic confidence models. To the best of our knowledge, SUNNY is the first trust inference algorithm that includes a confidence measure in its computation. SUNNY performs a probabilistic logic sampling procedure as in (Kuter *et al.* 2004) over the Bayesian Network generated by our representation mapping. In doing so, it computes estimates of the lower and upper bounds on the confidence values, which are then used as heuristics to generate the most accurate estimates of trust values of the nodes of the Bayesian Network.
- An experimental evaluation of SUNNY in comparison to the well-known existing work on TidalTrust (Golbeck 2005). Our experiments show that SUNNY significantly outperforms TIDALTRUST in the commonly-used FilmTrust social network.

Background

We use the usual definitions for social trust networks as in (Golbeck 2005). We define a *social domain* $\mathcal{S}$ as a tuple of the form (N, E, R, D). N is a finite set of *information-processing nodes* (or *IP nodes*, for short). For example, in social environments, an IP node corresponds to a person; on the Semantic Web, an IP node may also correspond to a Web service, a database, or any entity on the Web that takes information as input from other entities, processes that information, and relays it to other entities.

In $\mathcal{S}$, E is a finite set of *information elements* and R is an interval of the form $[0, r]$, for some predefined positive number r. A *decision* by an IP node n is an assignment of a number from R to an information element $e \in E$. Thus, D, the decision function of $\mathcal{S}$, is a function defined as

$$D : N \times E \rightarrow R.$$

If $D(n, e) = 0$ then n does not have a decision over e.

As an example, consider a social domain in which people make decisions over the ratings of a predefined set of movies. In this setting, N is the set of all people who participate in such decisions and E is the set of all of the possible movies that can be considered by the participants. If a person n makes a decision about a movie e, then $D(n, e) \neq 0$, and it is the rating that n has assigned to e.

A *trust network* is a directed graph $T = (\mathcal{S}, V, \alpha)$, where $\mathcal{S} = (N, E, R, D)$ is the social domain in which T is defined, V is the value function, and $\alpha > 0$ is the maximum possible absolute value of trust that appears in T. Intuitively, the value function V describes the *weights* of the trust relations between the IP nodes in the network. Formally, V is a function defined as

$$V : N \times N \rightarrow [0, \alpha].$$

If $V(n, n') = 0$ for two IP nodes $n, n' \in P$, then there is no direct trust relation (i.e., no edge) between n and n'; i.e., we have no knowledge about whether n trusts n' or not.

We say that an IP node n is an *T-ancestor* of n', if there exists a directed path in the trust network T that starts at n and ends in n'. Similarly, n' is a *T-descendant* of n in T.

A *trust inference problem* is a triple (T, n_0, n_∞), where $T = (\mathcal{S}, V, \alpha)$ is a trust network, and n_0 and n_∞ are the *source* and the *sink* nodes in T, respectively. A solution to a trust-inference problem is a trust value $0 < t \leq \alpha$ that describes the amount of trust that the source has for the sink in T. If there is no solution, then the amount of trust that the source has for the sink remains unknown.

We use the usual definitions for Bayesian Networks as in (Pearl 1988). A *Bayesian Network* is a directed acyclic graph $B = (X, A)$, where X is the set of *state variables* and A is the set of arcs between those nodes. Each variable in X describes a set of possible logical relations in the world. We assume in this paper that each state variable in X is Boolean, with the two possible truth values TRUE and FALSE. If there is an arc $x \rightarrow x'$ in A, such that $x, x' \in X$, then this means that the truth value of x' depends on the truth value of x.

The *parents* of a variable x in B is the set of variables, denoted as PARENTS(x), such that if there is an arc $x \rightarrow x'$ in A then x is in PARENTS(x'). x is a *terminal variable* in B, if PARENTS(x) is the empty set.

Each variable x in B is associated with a *conditional probability table (CPT)*, which describes the conditional probability of x being TRUE (or FALSE) given its parents. Note that if PARENTS$(x) = \{x_1, x_2, \ldots x_k\}$ then the conditional probability table associated with x has 2^k entries for each possible combination of the truth values for each of its parents. If x is a terminal variable in B, then the conditional probability table associated with x specifies the *a priori* probabilities of x being TRUE or FALSE.

We make the usual *conditional independence* assumption in Bayesian Networks: given PARENTS(x), x is conditionally independent from rest of the variables in B. That is, whether the computation of the probability that x will be TRUE or FALSE depends only on the truth values of x's parents and on no other state variable in B.

Modeling Confidence in Social Networks

We now describe a probabilistic interpretation of confidence in a given trust network T and a way to compute a probabilistic *confidence model* $\mathcal{C}$ for T. The next section describes a simple method that takes the trust network T and its confidence model $\mathcal{C}$, and generates a Bayesian Network B such that B structurally corresponds to T and is used for approximate probabilistic reasoning to assist trust computation.

Let $\mathcal{S} = (N, E, R, D)$ be a social domain, and n and n' be two IP nodes in N. We define the *confidence* that n has in n' as n's belief on the correctness of information provided by n'. We model the confidence of n for n' as the conditional probability $P(n|n')$ defined as follows: *given that n' conveys some information to n, the probability that n believes in the correctness of that information is $P(n|n')$.*

A *confidence model* $\mathcal{C}$ for the trust network $T = (\mathcal{S}, V, \alpha)$ is the set of conditional probabilities as above such that $\mathcal{C}$ specifies $P(n|n')$ for every pair of (n, n') in T such that $V(n, n') \neq 0$; i.e., if there is an arc from n to n' in T.

A confidence model for T can usually be generated in two ways. First is to take the confidence model as input from a "domain expert." This is the assumption made by most probabilistic reasoning and planning systems in the literature (see (Russell & Norvig 2003; Boutilier, Dean, & Hanks 1999) for two excellent surveys on these topics).

The second way to obtain a confidence model for T is to use approximation techniques such as statistical sampling and/or profile similarity measures. (Golbeck 2006) described how a combination of profile similarity measures reflect social opinions more accurately than overall similarity alone. Assume that all $D(n, e)$ are in the 0 to 1 range, where n is an IP node and e is an information element. The three profile features identified in (Golbeck 2006) were:

1. Overall Difference ($\Theta_{n,n'}$): Given the decisions that n and n' have made in common in the past (i.e., the set of values $D(n, e)$ and $D(n', e)$ for all information items e such that $D(n, e) \neq 0$ and $D(n', e) \neq 0$), $\Theta_{n,n'}$ is measured as the average absolute difference in their D values.
2. Difference on extremes ($\chi_{n,n'}$): A decision $D(n, e)$ is considered extreme if it is in the top 20% or the bottom

20% of the overall decisions made by n. $\chi_{n,n'}$ is computed as the average absolute difference on this set.

3. Maximum difference ($\nabla_{n,n'}$): The single largest difference on the same decision made by two IPs (i.e., $max\{(D(n,e) - D(n',e))\}, \forall e$ such that $D(n,e) \neq 0$ and $D(n',e) \neq 0$).

To indicate belief or disbelief, we define a coefficient $\sigma_{n,n'}$ in the range -1 to +1 such that if $\sigma(n,n') > 0$, then the conditional probability $P(n|n')$ denotes the amount of belief of n in the information provided by n' (i.e., the amount of the causal supportive influence that the information from n' has on n's decisions). Otherwise, $P(n|n')$ denotes the amount of disbelief of n in the information provided by n' (i.e., the amount of inhibitory influence that the information from n' has on n's decisions).

We compute $\sigma_{n,n'}$ as follows. If $\Theta_{n,n'}$ is more than one standard deviation above the mean Θ, then the IPs should be considered to generally disbelieve in one another, so $\sigma_{n,n'} = -1$. If $\Theta_{n,n'}$ is more than one standard deviation below the mean, then the IPs should be considered to generally believe in one another, so $\sigma_{n,n'} = 1$. When $\Theta_{n,n'}$ is within one standard deviation of the mean, it is not as clear whether there is belief or disbelief. In these cases, we used as a multiplier the Pearson correlation coefficient (ρ) over all common decisions made by n and n': $\sigma_{n,n'} = \rho$.

The result is Equation 1. Note that weights w_i may vary from network to network. The weight assignment $(w_1, w_2, w_3, w_4) = (0.7, 0.2, 0.1, 0.8)$ followed from our experiments and also provided the best results (Golbeck 2006).

$$P(n|n') = \begin{pmatrix} \sigma_{ij}|1 - 2(w_1\Theta_{n,n'} + w_2\nabla_{n,n'} + w_3\chi_{n,n'})|, \\ \qquad\qquad \text{if } \chi_{n,n'} \text{ exists} \\ \\ \sigma_{ij}|1 - 2(w_4\Theta_{n,n'} + (1-w_4)\nabla_{n,n'})|, \\ \qquad\qquad \text{otherwise} \end{pmatrix} \quad (1)$$

In the next section, we will use the above formula for computing confidence values (i.e., a conditional probabilities of belief and disbelief) between two information processors are already connected in the network.

A Bayesian Network Formulation of Trust Inference Problems

Given a trust inference problem (T, n_0, n_∞), we construct $B_T = (X, A)$, a Bayesian Network that is suited for approximate probabilistic confidence reasoning as follows.

Let L be the set of nodes in T such that for each node $n \in L$, $V(n, n_\infty) \neq 0$ and n is a T-descendant of n_0, where n_0 and n_∞ are the source and the sink nodes in T. Intuitively, the set L contains each n node in T that (1) has some direct trust information about the sink in T and (2) can relay that information to the source.

We compute the set N of nodes in B_T as follows (see the GENERATEBN pseudocode in Figure 2 for this procedure). The nodes in L described above correspond to the leaf variables in B_T; i.e., each n in L does not have any parents in B_T. Starting from the nodes in L, we perform

Procedure GENERATEBN(T, n_0, n_∞)
  $K \leftarrow$ the immediate neighbors of n_∞ in T
  $K' \leftarrow \emptyset$
  while $K \neq K'$ and $n_0 \notin K$ do
    $K' \leftarrow K$; $K'' \leftarrow \emptyset$
    while $K'' \neq K$ do
      $K'' \leftarrow K$
      $K \leftarrow K \cup$ WEAKPREIMG(K, T)
    $K \leftarrow$ PRUNE-STATES(K)
  if $n_0 \in K$ then return K
  return FAILURE

Figure 2: The algorithm for generating a Bayesian Network, given a trust network T and two nodes, n_0 and n_∞, that are the source and the sink in T, respectively.

a variant of backward breadth-first search over the nodes of T towards the source. The basis of this backward search is a *weak preimage* computation, which was originally developed and used in a number of automated AI planning algorithms (Cimatti *et al.* 2003). Formally, the weak preimage of a given set of nodes, say K, in a trust network T in the context of our representation mapping is defined as follows:

$$\text{WEAKPREIMG}(K, T) = \{n \mid n \text{ is a node in } T, n \notin K, \\ n' \in K \text{ and } V(n, n') \neq 0\}.$$

Intuitively, a WEAKPREIMG of a set K of nodes in a trust network contains each node n that is not in K and that has a direct trust relation with some node n' in K.

The backward breadth-first search does successive WEAKPREIMG computations starting from the leaf nodes in L. The search stops when there no new nodes left to be visited. This generates a subnetwork of the original trust network T in which every possible path is guaranteed to end in T's sink node. However, the generated subnetwork may contain redundant nodes that are not reachable from the source as well as cyclic paths. Since such redundant nodes and cyclic paths will not provide any new social information to the source node in determining its trust value of the sink, we perform a forward search in the result network in order to eliminate the redundancies and cycles. The PRUNE-STATES subroutine in Figure 2 implements this forward search.

After the redundancies are eliminated, there may be some nodes in the network that loose their connection to the sink node, and therefore, become redundant. To identify and eliminate those nodes, GENERATEBN successively performs the WEAKPREIMG and the PRUNE-STATES computations, until there are no redundant and cyclic nodes left in the network. At this point, if the source node is in set K of visited nodes, then GENERATEBN returns the set K. Otherwise, it returns FAILURE since there is no trust connection between the source and the sink in the trust network T.

The nodes in K returned by GENERATEBN and the edges between those nodes constitute the Bayesian Network B_T as follows. For each node n in K, we define a Boolean variable x that denotes the logical proposition that whether n believes in the sink node n_∞. In B_T, X is the set of such logical propositions. A is the set of all of the edges between

Procedure SUNNY(T, n_0, n_∞)
  $B_T \leftarrow$ GENERATEBN(T)
  for every leaf node n in B_T do
    $decision[n] \leftarrow$ UNKNOWN
    $\langle P_\perp(n_0), P_\top(n_0)\rangle \leftarrow$ SAMPLE-BOUNDS(B_T)
  for every leaf node n in B_T do
    set the lower and upper probability bounds such that
      $P_\perp(n) = P_\top(n) = 1.0$
    $\langle P'_\perp(n_0), P'_\top(n_0)\rangle \leftarrow$ SAMPLE-BOUNDS(B_T)
    if $|P'_\top(n_0) - P_\top(n_0)| < \epsilon$ and $|P'_\perp(n_0) - P_\perp(n_0)| < \epsilon$
      then $decision[n] \leftarrow$ TRUE
      else $decision[n] \leftarrow$ FALSE
  return (COMPUTE-TRUST$(B_T, decision)$)

Figure 3: SUNNY, the procedure for computing trust and confidence in a trust network.

the nodes in K, with the directions of those edges reversed.[1] Then, we use Equation 1 of the previous section in order to compute the conditional probabilities for every edge in B_T.

Note that GENERATEBN aggressively eliminates nodes and edges from the original network, which may result in the elimination of certain acyclic paths as well. This aggressive strategy guarantees the elimination of all possible cycles in trust networks with complicated structures, in order to create a Bayesian Network (i.e., a directed acyclic graph). However, as our experimental evaluation in the subsequent sections demonstrate, this does not have any significant effect on the accuracy of the trust computation: our algorithm was able to outperform the well-known TIDALTRUST algorithm, despite its aggressive edge elimination.

Note also that the Bayesian Network generated as above do not necessarily model a complete joint probability distribution, as one would expect from a standard Bayesian Network. This is due to two reasons: (1) the network generated above does not include all of the parents of a node since we are aggressively eliminating nodes during its construction, and (2) Equation 1 that we use for computing a confidence model does not compute a full CPT, but only the conditional probabilities between pairs of nodes.

SUNNY

Figure 3 shows the pseudocode of the SUNNY procedure for computing the confidence and trust values in a trust network. The input for the procedure is a trust-inference problem (T, n_0, n_∞). With this input, SUNNY first generates the Bayesian Network B_T that corresponds to T using the GENERATEBN procedure described above. SUNNY then generates estimates of the lower and upper bounds on the confidence values of each node in B_T in the sink node. SUNNY uses these estimates to make a decision on each leaf node in B_T whether to include that node in the final trust computation or not. During its computation, SUNNY considers three types of decisions for a leaf node: *include*, *exclude*, and *unknown*. Including a leaf node means that the trust information coming from that leaf node about the sink will be considered in the trust inference; otherwise, it will not be considered since SUNNY will have a low confidence in that trust information. An *unknown* decision on a leaf node means that SUNNY has not decided yet whether to include or to exclude that node in its operation.

SUNNY uses a probabilistic logic sampling technique in order to compute the confidence value of the source n_0 in the sink n_∞ in B_T. A *probabilistic logic sampling* algorithm runs successive simulations over a Bayesian Network and samples whether each variable is assigned to TRUE or FALSE (Henrion 1988). In SUNNY, we used a variant of probabilistic logic sampling described in (Kuter *et al.* 2004), with the following difference. In their sampling procedure, Kuter *et al.* use a complex probabilistic approximation rule for reasoning about the dependencies between the parents of a node that are not explicitly modeled in the network as well as for reasoning about unmodeled (i.e., hidden) nodes in the underlying Bayesian Network. In a social trust context, we usually do not have complex unseen implicit dependencies nor hidden nodes since the trust information flows through the existing links between the existing nodes in the network. Thus, our sampling algorithm does not include these aspects; instead as an approximation rule, it simply uses a standard Noisy-OR computation (Pearl 1988).

In Figure 3, the SAMPLE-BOUNDS subroutine performs our probabilistic sampling procedure, which provides a way to estimate the lower and upper bounds, $P_\perp(n)$ and $P_\top(n)$, on the confidence value of a node n. The lower and upper bounds $P_\perp(n)$ and $P_\top(n)$ for a leaf node n is simply 0.0 and 1.0, respectively, if SUNNY decides to mark the node n as *unknown*. For *include* and *exclude* decisions, SUNNY sets the bounds to both 1.0's and both 0.0's, respectively.

The bounds $P_\perp(n)$ and $P_\top(n)$ for an intermediate node n is computed based on the current estimates of the lower and upper bounds computed for the parents of n during the probabilistic sampling process. In each simulation, SAMPLE-BOUNDS samples all of the parents of n before it processes n. This way, the procedure always knows the truth values on the parents of a node n at the point it attempts to sample n, and therefore, for each parent n_i of n, the conditional probability $P(n|n_i)$ can easily be determined from the input confidence model.

For each intermediate node n, SAMPLE-BOUNDS probabilistically samples two cases: (1) the case where n is TRUE with minimum probability and (2) the case where n is TRUE with the maximum probability. The minimum and the maximum probabilities that n is TRUE are computed using the conditional probabilities between n and its parents, and the truth values of each of the parent's minimum and maximum case. Note that, this is a recursive definition: at a leaf node, the minimum and maximum cases are sampled by using the confidence bounds that depends on SAMPLE-BOUNDS's decision on that node, as described above.

SAMPLE-BOUNDS keeps a counter for both minimum and maximum case at each intermediate node, which are ini-

[1] Note that it is correct to have the reverse edges in B_T; when there is a trust relation in T between two nodes n and n' is represented as $n \rightarrow n'$, denoting "n knows n' with some amount of trust", then the information flows from n' to n, when we are computing trust values. Thus in B_T, the reverse edge $n \leftarrow n'$ captures the correct direction of this information flow.

tialized to 0 at the very beginning of the sampling process. If, in this simulation, the algorithm samples that n is TRUE in the minimum and/or in the maximum case, then the corresponding counters are incremented. At the end of the successive simulations, the counter values associated with each node are divided by the total number of simulations in order to compute the estimates for the lower and upper bounds $P_{\perp}(n)$ and $P_{\top}(n)$.

SUNNY uses the lower and upper bounds computed by the SAMPLE-BOUNDS subroutine as follows. First, SUNNY invokes SAMPLE-BOUNDS with no decisions made (i.e., every leaf node in B_T is marked UNKNOWN). This invocation produces a lower and an upper bound on the confidence of the source, which are essentially the minimum and the maximum possible confidence values, respectively. Then, SUNNY uses these bounds in a hill-climbing search in order to actually finalize a decision on whether to include or exclude a leaf node in the trust computation. More precisely, SUNNY invokes SAMPLE-BOUNDS for each leaf node n with the a priori probabilities on both of the bounds over n set to 1.0. If this invocation of SAMPLE-BOUNDS produces minimum and maximum possible confidence values within an error margin ϵ of the ones computed before, then n should be included in the final trust computation, since the source has these confidence bounds when it considers the information coming from n. Otherwise, n should be excluded from the final trust computation.

Once the decisions on all of the leaf nodes of B_T has been made, SUNNY computes the trust value for the source by performing a backward search from the leaf nodes of B_T towards the source. In Figure 3, COMPUTE-TRUST subroutine performs the backward search. At each iteration during this search, the trust value of a node is computed based on the trust values between that node and its immediate parents and the trust values of the parents in the sink node that are already computed in the search. Although there are several ways to combine these two pieces of trust information, SUNNY uses the same weighted-averaging formula used in the TIDALTRUST algorithm, as we used that algorithm in our experiments with SUNNY described in the next section.

Experimental Evaluation

Because SUNNY is the first trust inference algorithm that includes a confidence measure in its computation, there is no existing work against which the confidence component can be judged. However, for both values to be useful the computed trust value must be accurate. We computed the accuracy of the trust values and compared them against the performance of TIDALTRUST. TIDALTRUST is one of several well-known localized trust computation algorithms, and previous work has shown it to perform as well as or better than its peer algorithms.

We used the FilmTrust network as the data source for our experiments. FilmTrust[2] is a social networking website where users have assigned trust values to their relationships. Users also rate movies in the system, and we used these ratings to compute confidence values as described in the previous sections. Because it is publicly accessible on the Semantic Web, the FilmTrust network is a commonly used data set for this type of analysis.

[2]Detailed information on FilmTrust is available at http://trust.mindswap.org

Table 1: The comparisons between the accuracies of SUNNY and TIDALTRUST on the FilmTrust network.

Algorithm	Average Error	Std. Dev
TIDALTRUST	1.986	1.60
SUNNY	1.856	1.44

There are 575 people actively participating in the FilmTrust social network, with an average degree of 3.3. Each pair of users connected in the network shared an average of 7.6 movies in common. When pairs of users had no movies or only one movie in common, there was not sufficient information to compute confidence from similarity. In these cases, we scaled the trust value to a -1 to 1 scale and used the sign of that value as our belief/disbelief coefficient and the unsigned value as the confidence value.

To gauge the accuracy of the trust inference algorithms, we selected each pair of connected users, ignored the relationship between them, and computed the trust value. We then compared the computed value to the known trust value and measured error as the absolute difference. We performed this process for every connected pair of users in the network where there were other paths connecting them. If there is no path between users beside the direct connection, it is impossible to make a trust computation from the network, and these pairs are ignored. In total, 715 relationships were used in this evaluation.

As shown in Table 1, the average error for SUNNY is 6.5% lower than the average error for TIDALTRUST. A standard two-tailed t-test shows that SUNNY is significantly more accurate than TIDALTRUST for $p < 0.05$. While accuracy will vary from network to network due to variations in structure, users' ratings, and confidences, these results demonstrate that SUNNY is an effective algorithm for computing trust in social networks.

Related Work

(Pearl 1988) provides a comprehensive treatment of network models of probabilistic reasoning. A probabilistic network model defines a joint probability distribution over a given set of random variables, where these variables appear as the nodes of the network and the edges between these nodes denote the dependencies among them. One widely-known network model that we exploited in this work is *Bayesian Networks* (Pearl 1988). Bayesian networks are directed acyclic graphs in which the arcs denote the causal and conditional dependencies among the random variables of the probabilistic model. The inference mechanisms in Bayesian Networks are due to Bayesian Inference, which provides an incremental recursive procedure to update beliefs when new evidence about the input probabilistic model is acquired.

Probabilistic Logic Sampling (Henrion 1988) is a technique for performing stochastic simulations that can make

probabilistic inferences over a Bayesian Network with an arbitrary degree of precision controlled by the sample size. It runs successive simulations to estimate the probabilities of the nodes of a given Bayesian Network. As we run more and more simulations, the estimates of the probability of each variable get progressively more accurate, and it is shown in (Henrion 1988) that these estimates converge to the exact probability values in the limit. However, in practice, usually a termination criterion is used to determine the number of simulations (e.g., until the residual of every node becomes less than or equal to an epsilon value > 0 or until an upper bound k on the number of simulations is reached).

While there are no algorithms in the literature on social networks that compute both trust and confidence values (i.e., probabilistic belief models), there are several algorithms for computing trust. A thorough treatment can be found in (Golbeck 2005). The TIDALTRUST algorithm that we used in our experimental evaluation of SUNNY is one of several algorithms for computing trust in social networks and we chose it for our comparison because, like SUNNY, it outputs trust ratings in the same scale that users assign them, and the values are personalized based on the user's social context.

Trust has also been studied in the area of peer-to-peer systems. Among others, (Wang & Vasilleva 2003) describes a Bayesian Network based model of trust in peer-to-peer systems, where a naive Bayesian Network model is used to describe the trust relations between two agents (i.e., information processors in our context) and use a "small-world assumption" that enables an agent to keep its decisions based on the trust values for only its local neighborhood of other agents. In this paper, on the other hand, we focused on the propagation of confidence (i.e., belief) and trust values over the entire network of information processors and used confidence models to guide such a propagation to make it more efficient and accurate. In future, we are intending to investigate to generalize our trust computation techniques by incorporating Bayesian Network-based models for trust values.

Conclusions and Future Work

When using social trust in many applications, it is important to have an accurate estimate of trust when it is not directly available, as well as a measure of confidence in the computed value. In this work, we have introduced a probabilistic interpretation of confidence in trust network, a formal representation mapping from Trust Networks to Bayesian Networks, and a new algorithm, called SUNNY, for computing trust and confidence in social networks. In direct comparisons, SUNNY was shown to significantly outperform the well-known TIDALTRUST trust inference algorithm in terms of the accuracy of the computed trust values.

As future work we plan to integrate SUNNY into a working application to evaluate its effectiveness. We are involved in a project with the goal of developing a semantically rich web-based syndication framework. When matching queries over streaming news feeds, often times conflicting information will enter the knowledge base. If the source of each fact is known, trust can be used to prioritize information and select the most trusted of the inconsistent facts to include. Earlier work (Katz & Golbeck 2006) illustrated the use of trust as a prioritization tool for default logics. We plan to integrate our approach into this syndication system to study the use of the confidence values as well as the effectiveness of trust for this sort of prioritization.

Acknowledgments

This work, conducted at the Maryland Information and Network Dynamics Laboratory Semantic Web Agents Project, was funded by Fujitsu Laboratories of America – College Park, Lockheed Martin Advanced Technology Laboratory, NTT Corp., Kevric Corp., SAIC, the National Science Foundation, the National Geospatial-Intelligence Agency, DARPA, US Army Research Laboratory, NIST, and other DoD sources.

References

Boutilier, C.; Dean, T. L.; and Hanks, S. 1999. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research* 11:1–94.

Cimatti, A.; Pistore, M.; Roveri, M.; and Traverso, P. 2003. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence* 147(1-2):35–84.

Golbeck, J. 2005. *Computing and Applying Trust in Web-based Social Networks*. Ph.D. Dissertation, University of Maryland, College Park, MD, USA.

Golbeck, J. 2006. Trust and nuanced profile similarity in online social networks. In *MINDSWAP Technical Report TR-MS1284*.

Henrion, M. 1988. Propagating uncertainty in bayesian networks by probabilistic logic sampling. In *Uncertainty and Artificial Intelligence II*. Elsevier North Holland.

Katz, Y., and Golbeck, J. 2006. Social network-based trust in prioritized default logic. *Proceedings of the Twenty-First National Conference onArtificial Intelligence (AAAI-06)*.

Kuter, U.; Nau, D.; Gossink, D.; and Lemmer, J. F. 2004. Interactive Course-of-Action Planning using Causal Models. In *Third International Conference on Knowledge Systems for Coalition Operations (KSCO-2004)*.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Fransisco, CA: Morgan Kaufmann. Chapters 2-3.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence, A Modern Approach (Second Edition)*. Upper Saddle River, NJ: Prentice-Hall.

Wang, Y., and Vasilleva, J. 2003. Bayesian Network-Based Trust Model. In *IEEE/WIC International Conference on Web Intelligence*, 372. IEEE Computer Society.