Trust Management for the Semantic Web

Matthew Richardson[1†], Rakesh Agrawal[2], Pedro Domingos[1]

[1] University of Washington, Box 352350, Seattle, WA 98195-2350
[2] IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120-6099
{mattr, pedrod}@cs.washington.edu, ragrawal@acm.org

Abstract. Though research on the Semantic Web has progressed at a steady pace, its promise has yet to be realized. One major difficulty is that, by its very nature, the Semantic Web is a large, uncensored system to which anyone may contribute. This raises the question of how much credence to give each source. We cannot expect each user to know the trustworthiness of each source, nor would we want to assign top-down or global credibility values due to the subjective nature of trust. We tackle this problem by employing a web of trust, in which each user maintains trusts in a small number of other users. We then compose these trusts into trust values for all other users. The result of our computation is not an agglomerate "trustworthiness" of each user. Instead, each user receives a personalized set of trusts, which may vary widely from person to person. We define properties for combination functions which merge such trusts, and define a class of functions for which merging may be done locally while maintaining these properties. We give examples of specific functions and apply them to data from Epinions and our BibServ bibliography server. Experiments confirm that the methods are robust to noise, and do not put unreasonable expectations on users. We hope that these methods will help move the Semantic Web closer to fulfilling its promise.

1. Introduction

Since the articulation of the Semantic Web vision [9], it has become the focus of research on building the next web. The philosophy behind the Semantic Web is the same as that behind the World-Wide Web – anyone can be an information producer or consume anyone else's information. Thus far, most Semantic Web research (e.g., [6][27]) has focused on defining standards for communicating facts, rules, ontologies, etc. XML, RDF, RDF-schema, OWL and others form a necessary basis for the construction of the Semantic Web. However, even after these standards are in wide use, we still need to address the major issue of how to decide how trustworthy each information source is. One solution would be to require all information on the Semantic Web to be consistent and of high quality. But due to its sheer magnitude and diversity of sources, this will be nearly impossible. Much as in the development of the WWW, in which there was no attempt made to centrally control the quality of information, we believe that it is infeasible to do so on the Semantic Web.

Instead, we should develop methods that work under the assumption that the information will be of widely varying quality. On the WWW, researchers have found that one way to handle this is to make use of "statements" of quality implicit in the link structure between pages [23][26]. This collaborative, distributed approach is far more cost-effective than a centralized approach. We propose that a similar technique will

---

[†] Researched while at IBM Almaden Research Center

work on the Semantic Web, by having each user explicitly specify a (possibly small) set of users she trusts. The resulting web of trust may be used recursively to compute a user's trust in any other user (or, more precisely, in any other user in the same connected component of the trust graph). Note that, unlike PageRank, the result of our computation is not an agglomerate "trustworthiness" of each user. Instead, each receives her own personalized set of trusts, which may be vastly different from person to person. In this paper, we propose and examine some methods for such a computation.

In Section 2 we formulate a model that explicitly has the dual notions of *trust* and *belief*. Then, in Sections 3, 4, and 5, we define the meaning of belief combination under two different interpretations, and show an equivalence between the two. We also show a correspondence between combining beliefs and trusts that allows the use of whichever is more computationally efficient for the given system. We then give experimental results that show that our methods work across a wide variation of user quality and noise. We conclude with a discussion of related and future work.

2. Model

We assume content on the Semantic Web is (explicitly or implicitly) in the form of logical assertions. If all these assertions are consistent and believed with certainty, a logical calculus can be used to combine them. If not, a probabilistic calculus may be used (e.g., knowledge-based model construction [25]). However, our focus here is not on deriving beliefs for new statements given an initial set of statements. Rather, we propose a solution to the problem of establishing the degree of belief in a statement that is explicitly asserted by one or more sources on the Semantic Web. These beliefs can then be used by an appropriate calculus to compute beliefs in derived statements. Our basic model is that a user's belief in a statement should be a function of her trust in the sources providing it. Given each source's belief in the statement and the user's trust in each source, the user's belief in the statement can be computed in many different ways, corresponding to different models of how people form their beliefs. The framework presented in this paper supports a wide variety of combination functions, such as linear pool [17][18], noisy OR [28], and logistic regression [4]. We view the coefficients in these functions (one per source) as measuring the user's trust in each source,[1] and answer the question: how can a user decide how much to trust a source she does not know directly? Our answer is based on recursively propagating trust: if A has trust u in B and B has trust v in C, then A should have some trust t in C that is a function of u and v. We place restrictions on allowable methods for combining trusts that enable the efficient and local computation of derived trusts. Similar restrictions on belief combination allow it to also be done using only local information.[2]

Consider a system of N users who, as a whole, have made M statements. Since we consider statements independently, we introduce the system as if there is only one.

[1] Trust is, of course, a complex and multidimensional phenomenon, but we make a start in this paper by embodying it in a single numeric coefficient per user-source pair.

[2] While this may not guarantee the probabilistic soundness of the resulting beliefs, we believe it is necessary for scalability on the size of the Web, and our experiments indicate it still produces useful results. Scalable probabilistic approximations are a direction for future research.

Beliefs. Any user may assert her *personal belief* in the statement, which is taken from [0,1]. A high value means that the statement is accurate, credible, and/or relevant. Let b_i represent user i's personal belief in the statement. If user i has not provided one, we set b_i to 0. We refer to the collection of personal beliefs in the statement as the column vector **b** (see Section 8 for a discussion on more complex beliefs and trusts).

Trusts. User i may specify a *personal trust*, t_{ij}, for any user j. Trust is also a value taken from [0,1], where a high value means that the user is credible, trustworthy, and/or shares similar interests. If unspecified, we set t_{ij} to be 0. Note that t_{ij} need not equal t_{ji}. The collection of personal trusts can be represented as a N×N matrix **T**. We write $\mathbf{t}_i$ to represent the row vector of user i's personal trusts in other users.

Merging. The web of trust provides a structure on which we may compute, for any user, their belief in the statement. We will refer these as *merged beliefs* ($\boldsymbol{\mathscr{b}}$), to distinguish them from the user-specified *personal beliefs* (**b**). The trust between any two users is given by the *merged trusts matrix* ($\boldsymbol{\mathscr{T}}$), as opposed to the user-specified *personal trusts matrix* (**T**).

3. Path Algebra Interpretation

In order to compute merged beliefs efficiently, we first make the simplifying assumption that a merged belief depends only on the paths of trust between the user and any other user with a personal belief in the statement. In Section 4 we consider an alternative probabilistic interpretation. For the moment, we consider only acyclic graphs (we generalize later to cyclic graphs).

Borrowing from generalized transitive closure literature [3], we define merged beliefs under the path algebra interpretation with the following conceptual computation:

1. Enumerate all (possibly exponential number of) paths between the user and every user with a personal belief in the statement.
2. Calculate the belief associated with each path by applying a *concatenation function* to the trusts along the path and also the personal belief held by the final node.
3. Combine those beliefs with an *aggregation function*.

(See Figure 1). Some possible concatenation functions are multiplication and minimum value. Some possible aggregations functions are addition and maximum value. Various combinations lead to plausible belief-merging calculations such as measuring the most-reliable path or the maximum flow between the user and the statement.

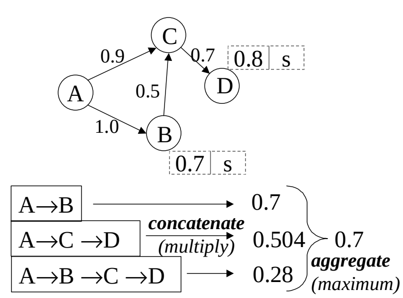

Figure 1: Path Algebra belief merging on an example web of trust.

Let ○ and ◊ represent the concatenation and aggregation functions respectively. For example, $t_{ik} \circ t_{kj}$ is the amount that user i trusts user j *via k*, and the amount that i trusts j via any single other node is $\lozenge(\forall k\colon t_{ik} \circ t_{kj})$. If $\lozenge$ is addition and $\circ$ is multiplication, then $\lozenge(\forall k\colon t_{ik} \circ t_{kj}) \equiv \sum_k t_{ik} t_{kj}$. We define the matrix operation $\mathbf{C}=\mathbf{A}\bullet\mathbf{B}$ such that $\mathbf{C}_{\mathrm{ij}} = \lozenge(\forall k\colon A_{ik} \circ B_{kj})$. Note that for the previous example, $\mathbf{A}\bullet\mathbf{B}$ is simply matrix multiplication.

3.1 Local Belief Merging

The global meaning of beliefs given above assumes a user has full knowledge of the network including the personal trusts between all users, which is practically unreasonable. Can we instead merge beliefs locally while keeping the same global interpretation? Following [3], let *well-formed decomposable path problems* be defined as those for which $\lozenge$ is commutative and associative, and $\circ$ is associative and distributes over $\lozenge$ (The above examples for $\lozenge$ and $\circ$ all result in well-formed path problems). These may be computed using *generalized transitive closure* algorithms, which use only local information. One such algorithm is as follows:

1. $\boldsymbol{b}^{(0)} = \mathbf{b}$
2. $\boldsymbol{b}^{(n)} = \mathbf{T}\bullet\boldsymbol{b}^{(n\text{-}1)}$, or alternatively, $b_i^{(n)}=\lozenge(\forall k\colon t_{ik} \circ b_k^{(n\text{-}1)})$
3. Repeat step 2 until $\boldsymbol{b}^{(n)} = \boldsymbol{b}^{(n\text{-}1)}$

(where $\boldsymbol{b}^{(i)}$ represents the value of $\boldsymbol{b}$ in iteration i. Recall $\boldsymbol{b}$ are the merged beliefs)

Notice that in step 2, the user needs only the merged beliefs of her immediate neighbors, which allows her to merge beliefs *locally* while keeping the same *global* interpretation. We will use the term *belief combination function* to refer to the above algorithm and some selection of $\circ$ and $\lozenge$.

3.2 Strong and Weak Invariance

Refer to Figure 2 (*Case I*). Suppose a node is removed from the web of trust, and the edges to it are redirected to its trusted nodes (combining the trusts). If the merged beliefs of the remaining users remain unchanged, we say the belief combination function has *weak global invariance*. The path interpretation has this important property.

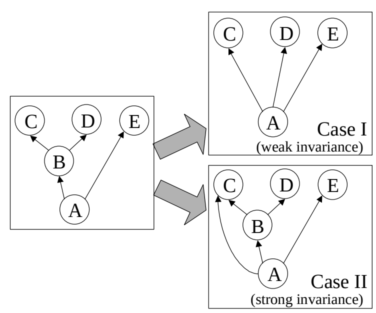

Figure 2: Strong and weak invariance.

We can imagine another property that may be desirable. Again refer to Figure 2 (*Case II*). If we add an arc of trust directly from A to C, and the trust between A and C is unchanged, we say that the belief combination function has *strong global invariance.* Any belief combination function with weak invariance for which the aggregation function is also *idempotent* (meaning, $\lozenge(x, x) = \lozenge(x)$), will have strong invariance. This follows from the fact that the aggregation function is associative. Interestingly, whether or not the aggregation function must be idempotent is the primary difference between Agrawal's well-formed decomposable path problems [3] and Carre's path algebra [11] (also related is the definition of a closed semiring in [5]). One example of a belief combination function with strong global invariance is the one defined with $\lozenge$ as maximum and $\circ$ as multiplication.

3.3 Merging Trusts

The majority of the belief merging calculation involves the concatenation of chains of trust. Beliefs only enter the computation at the endpoint of each path. Instead of merging beliefs, can we merge trusts and then reuse these to calculate merged beliefs?

We define the interpretation of globally merged trusts in the same way as was done for beliefs: the trust between user *i* and user *j* is an aggregation function applied to the concatenation of trust along every path between them. It falls directly from path algebra that, if ◊ is commutative and associative, and ○ is associative and distributes over ◊, then we can combine trusts locally while still maintaining global meaning:

$$\mathcal{T}^{(0)} = \mathbf{T}, \qquad \mathcal{T}^{(n)} = \mathbf{T}\bullet\mathcal{T}^{(n-1)}, \qquad \text{Repeat until } \mathcal{T}^{(n)} = \mathcal{T}^{(n-1)}$$

($\mathcal{T}^{(i)}$ is the value of $\mathcal{T}$ in iteration *i*. Recall $\mathcal{T}$ is the matrix of merged trusts). To perform the computation, a user needs only to know her neighbors' merged trusts. This leads us to the following theorem, which states that, for a wide class of functions, merging trusts accomplishes the same as merging beliefs (the proof is in the Appendix)

Theorem 1: If ◊ is commutative and associative, and ○ is associative and distributes over ◊, and $\mathbf{T}$, $\mathcal{T}$, $\mathbf{b}$, and $\mathcal{b}$ are as above, then $\mathbf{T}\bullet\mathcal{b} = \mathcal{T}\bullet\mathbf{b}$.

3.4 Cycles

Thus far, we have assumed the graph is acyclic. However, it is improbable that a web of trust will be acyclic. Indeed, the Epinions web of trust (see Section 6.1) is highly connected and cyclic. Borrowing terminology from path algebra, we define a combination function as *cycle-indifferent* if it is not affected by introducing a cycle in the path between two users. With cycle indifference, the aggregation over infinite paths will converge, since only the (finite number of) paths without cycles affect its calculation.

Proposition 1: All of the results and theorems introduced thus far are applicable to cyclic graphs if ◊ and ○ define a cycle-indifferent path problem.

On cyclic graphs, a combination function that is not cycle-indifferent has the questionable property that a user may be able to affect others' trusts in him by modifying her own personal trusts. However, requiring a cycle-indifferent combination function may be overly restrictive. In Section 4 we explore an alternative interpretation that allows the use of combination functions that are not cycle-indifferent.

3.5 Selection of Belief Combination Function

The selection of belief combination function may depend on the application domain, desired belief and cycle semantics, and the expected typical social behavior in that domain. The ideal combination function may be user-dependent. For the remainder of the paper, we will always use multiplication for concatenation, though in the future we would like to explore other functions (such as the minimum value). The following is a brief summary of three different aggregation functions we have considered.

Maximum Value. Using maximum to combine beliefs is consistent with fuzzy logic, in which it has been shown to be the most reasonable function for performing a generalized *or* operation over [0,1] valued beliefs [8]. Maximum also has the advantages that it is cycle-indifferent, strongly consistent, and naturally handles missing values (by letting them be 0). With maximum, the user will believe anything believed by at least one of the users she trusts – a reasonable, if not overly optimistic, behavior.

Minimum Value. Minimum is not cycle-indifferent. In fuzzy logic, minimum value is used to perform the *and* operation. With minimum, the user will only believe a statement if it is believed by all of the users she trusts.

Average. Average does not satisfy the requirements for a well-formed path algebra outlined above (average is not associative). However, average can still be computed by using two aggregation functions: sum and count (count simply returns the number of paths by summing 1's). By passing along these two values, each node can locally compute averages. Average is not cycle-indifferent.

3.6 Computation

Since cycle-indifferent, weakly consistent combination functions are well-formed path problems, $\mathcal{B}$ and $\mathcal{T}$ may be computed using standard transitive closure algorithms. The simplest of these is the semi-naïve algorithm [7], which runs in $O(N^4)$ time, and essentially prescribes repeated application of the belief update equation. If running as a peer-to-peer system, the semi-naïve algorithm may be easily parallelized, requiring $O(N^3)$ computations per node [2]. Another algorithm is the Warshall algorithm [33], which computes the transitive closure in $O(N^3)$. Some work on parallel versions of the Warshall algorithm has been done in [2]. There has also been much research on optimizing transitive closure algorithms, such as for when the graph does not fit into memory [3]. In practice most users will specify only a few of the users as neighbors, and the number of iterations required to fully propagate information is much less than N, making the computation quite efficient. Theorem 1 allows us to choose whether we wish to merge trusts or merge beliefs. The most efficient method depends on, among other things, whether the system is implemented as a peer-to-peer network or as a server, the number of neighbors for a given user, the number of users, the number of statements in the system, and the number of queries made by each user.

4. Probabilistic Interpretation

In this formulation, we consider a probabilistic interpretation of global belief combination. The treatment is motivated by random walks on a Markov chain, which have been found to be of practical use in discovering high-quality web pages [26]. In what follows, we assume the set of personal trusts for a given user has been normalized.

Imagine a random knowledge-surfer hopping from user to user in search of beliefs. At each step, the surfer probabilistically selects a neighbor to jump to according to the current user's distribution of trusts. Then, with probability equal to the current user's belief, it says "yes, I believe in the statement". Otherwise, it says "no". Further, when choosing which user to jump to, the random surfer will, with probability $\lambda_i \in [0,1]$, ignore the trusts and instead jump directly back to the original user, *i*. We define a combination method to have a *global probabilistic interpretation* if it satisfies the following:

1) $\mathcal{T}_{ij}$ is the probability that, at any given step, user *i*'s random surfer is at user *j*.
2) $\mathcal{b}_i$ is the probability that, at any given step, user *i*'s random surfer says "yes".

The convergence properties of such random walks are well studied; $\boldsymbol{\mathcal{b}}$ and $\boldsymbol{\mathcal{T}}$ will converge as long as the network is irreducible and aperiodic [24]. λ_i can be viewed as a *self-trust*, and specifies the weight a user gives to her own beliefs and trusts. The behavior of the random knowledge-surfer is very similar to that of the intelligent surfer presented in [32], which is a generalization of PageRank that allows non-uniform transitions between web pages. What personalizes the calculation to user *i* is the random restart, which "grounds" the surfer to *i*'s trusts. The resulting trusts may be drastically different than using PageRank, since the number of neighbors will typically be small.

4.1 Computation

User *i*'s trust in user *j* is the probability that her random surfer is on a user *k*, times the probability that the surfer would transition to user *j*, summed over all *k*. Taking λ_i into account as well, we have $\mathcal{T}_{ij} = \lambda_i \delta(i-j) + (1-\lambda_i)\sum_k \mathcal{T}_{ik} t_{kj}$,
where $\delta(0)=1$ and $\delta(x\neq 0)=0$ and each row of **t** is normalized. In matrix form:

$$\mathcal{T}_i = \lambda_i \mathbf{I}_i + (1-\lambda_i)\mathcal{T}_i \mathbf{T} , \tag{1}$$

where $\mathbf{I}_i$ is the i^{th} row of the identify matrix. In order to satisfy the global probabilistic interpretation, $\boldsymbol{\mathcal{b}}_i$ must be the probability that user *i*'s random surfer says "yes". This is the probability that it is on a given user times that user's belief in the statement:

$$\mathcal{b}_i = \sum_k \mathcal{T}_{ik} b_k \ , \quad \text{or,} \quad \mathcal{b}_i = \mathcal{T}_i \mathbf{b} \tag{2}$$

4.2 Local Belief and Trust Merging

As in section 3.1, we wish to perform this computation using only local information. We show that this is possible in the special case where $\lambda_i=\lambda$ is constant.
Unrolling Equation 1:

$$\mathcal{T} = \lambda\left[\sum_{m=0}^{\infty}(1-\lambda)^m \mathbf{T}^m\right]. \tag{3}$$

Note that $\mathbf{T}^0=\mathbf{I}$. Substituting into Equation 2,

$$\mathcal{b} = \lambda\left[\sum_{m=0}^{\infty}(1-\lambda)^m \mathbf{T}^m\right]\mathbf{b} , \tag{4}$$

which is satisfied by the recursive definition:

$$\mathcal{b} = \lambda\mathbf{b} + (1-\lambda)\mathbf{T}\mathcal{b} \tag{5}$$

Thus we find that in order to compute her merged belief, each user needs only to know her personal belief, and the merged beliefs of her neighbors. Besides having intuitive appeal, it has a probabilistic interpretation as well: user *i* selects a neighbor probabilistically according to her distribution of trust, $\mathbf{T}_i$, and then, with probability (1-λ), accepts that neighbor's (merged) belief, and with probability λ accepts her own belief. Further, Equation 3 is also equivalent to the following, which says that a user may compute her merged trusts knowing only the merged trusts of her neighbors:

$$\mathcal{T} = \lambda \mathbf{I} + (1-\lambda)\mathbf{T}\mathcal{T} \qquad (6)$$

The probabilistic interpretation for belief combination is essentially taking the weighted average of the neighbors' beliefs. We will thus refer to this belief combination as *weighted average* for the remainder of the paper. Note that for weighted average to make sense, if the user has not specified a belief we need to impute the value. Techniques such as those used in collaborative filtering [30] and Bayesian networks [13] for dealing with missing values may be applicable. If only relative rankings of beliefs are necessary, then it may be sufficient to use 0 for all unspecified beliefs.

5. Similarity of Probabilistic and Path Interpretations

There are clearly many similarities between the probabilistic and path interpretations. In both, beliefs may be merged by querying neighbors for their beliefs, multiplying (or concatenating) those by the trust in each neighbor, and adding (or aggregating) them together. Both interpretations also allow the computation of merged beliefs by first merging trusts. If we let the aggregation function be addition, and the concatenation function be multiplication, then the only difference between the two interpretations is due to the factor, λ. If λ=0, then Equation 5 for computing $\boldsymbol{\mathcal{b}}$ is functionally the same as the algorithm for computing $\boldsymbol{\mathcal{b}}$ in the path algebra interpretation. However, consider this: If λ is 0 then Equation 1 for computing $\mathcal{T}_i$ simply finds the primary eigenvector of the matrix $\mathbf{T}$. Since there is only one primary eigenvector, this means that $\mathcal{T}_i$ would be the same for all users (assuming the graph is aperiodic and irreducible). How do we reconcile this with the path algebra interpretation, in which we expect different trust vectors per user? The answer is that the corresponding path algebra combination function is not cycle indifferent, and as a result the user's personal beliefs will get "washed out" by the infinite aggregation of other users' beliefs. Hence, as in the probabilistic interpretation, all users would end up with the same merged beliefs.

Both methods share similar tradeoffs with regards to architectural design. They may easily be employed in either a peer-to-peer or client-server architecture. We expect the system to be robust because a malicious user will be trusted less over time. Further, since the default trust in a user is 0, it is not useful for a user to create multiple pseudonyms, and users are motivated to maintain quality of information.

The web of trust calculation is not susceptible to "link-spamming," a phenomenon in PageRank whereby a person may increase others' trust in him by generating hundreds of virtual personas which all trust him. In PageRank, the uniform random jump of the surfer means that each virtual persona is bestowed some small amount of PageRank, which they 'give' to the spammer, thus increasing her rank. With a web of trust, this technique gains nothing unless the user is able to convince others to trust her virtual personas, which we expect will only occur if the personas actually provide useful information.

6. Experiments

In this section, we measure some properties of belief combination using the methods from this paper. We present two sets of experiments. The first uses a real web of trust, obtained from Epinions (www.epinions.com), but uses synthetic values for personal beliefs and trusts. We wanted to see how *maximum* (path interpretation) compared with *weighted average* (probabilistic interpretation) for belief combination. We also wanted to see what quality of user population is necessary for the system to work well, and what happens if there are mixes of both low and high quality users. Finally, these methods would have little practical use if we required that users be perfect at estimating trusts of their neighbors, so we examine the effect that varying the quality of trust estimation has on the overall accuracy of the system. For the second experiment, we implemented a real-world application, now available over the web (BibServ, www.bibserv.org). BibServ provides us with both anecdotal and experimental results.

6.1 Experiments with the Epinions Web of Trust

For these experiments, we used the web of trust obtained from Epinions, a user-oriented product review website. In order to maintain quality, Epinions encourages users to specify which other users they trust, and uses the resulting web of trust to order the product reviews seen by each person[3]. In order to perform experiments, we needed to augment the web of trust with statements and real-valued trusts.

We expected the information on the Semantic Web to be of varying quality, so we assigned to each user *i* a *quality* $\gamma_i \in [0,1]$. A user's quality determined the probability that a statement by the user was true. Unless otherwise specified, the quality of a user was chosen from a Gaussian distribution with $\mu = 0.5$ and $\sigma = 0.25$. These parameters are varied in the experiments below.

The Epinions web of trust is Boolean, but our methods require real-valued trusts. We expected that over time, the higher a user's quality, the more they were likely to be trusted. So, for any pair of users *i* and *j* where *i* trusts *j* in Epinions:

$$t_{ij} = \text{uniformly chosen from } [\max(\gamma_j - \delta_{ij}, 0), \min(\gamma_j + \delta_{ij}, 1)] \qquad (7)$$

where γ_i is the quality of user *i* and δ_{ij} is a noise parameter that determines how accurate users were at estimating the quality of the user they were trusting. We supposed that a user with low quality was bad at estimating trust, so for these experiments we let $\delta_{ij}=(1-\gamma_i)$.

We generated a random world that consisted of 5000 true or false "facts" (half of the facts were false). Users' statements asserted the truth or falsity of each fact (there were thus 10,000 possible statements, 5000 of which were correct). A user's personal belief (b_i) in any statement she asserted was 1.0.

The number of statements made by a user was equal to the number of Epinions re-

[3] The trust relationships can be obtained by crawling the site, as described in [31]. Though the full graph contains 75,000 users, we restricted our experiments to the first 5000 users (by crawl-order), which formed a network of 180,000 edges.

Table 1: Average precision and recall for various belief combination functions, and their standard deviations.

Comb. Function	Precision	Recall
Maximum	0.87 ± 0.13	0.98 ± 0.13
Weighted Average	0.69 ± 0.06	0.98 ± 0.15
Local	0.57 ± 0.13	0.44 ± 0.32
Random	0.51 ± 0.05	0.99 ± 0.11

views that user wrote. The few users with highest connectivity tended to have written the most reviews, while the majority of users wrote few (or none).

For each fact, each user computed her belief that the fact was true and her belief that the fact was false. For each user i, Let S_i be the set of statements for which $\mathcal{b}_i > \tau$. If a user had non-zero belief that a fact was true and a non-zero belief that a fact was false, we used the one with highest belief. Let G_i be the set of correct statements "reachable" by user i (A statement is reachable if there is a path in the web of trust from user i to at least one user who has made the statement). Then $S_i \cap G_i$ is the set of statements that user i correctly believed were true, so $precision_i = | S_i \cap G_i | / | S_i |$ and $recall_i = | S_i \cap G_i | / | G_i |$. Precision and recall could be traded off by varying the belief threshold, τ. We present precision and recall results averaged over all users, and at the highest recall by using τ=0.

Comparing Combining Functions. In Table 1, we give results for a variety of belief combination functions. The combination functions *maximum* and *weighted average* are the same as introduced earlier (unless otherwise specified, λ is 0.5 for weighted average). With *random*, $\mathcal{T}_{ij}$ was chosen uniformly from [0,1]. Since the average quality is 0.5, half of the facts in the system are true, so *random* led to a precision of (roughly) 0.5. *Local* means that a user incorporated only the personal beliefs of her immediate neighbors, and resulted in a precision of 0.57. *Weighted average* and *maximum* significantly outperformed the baseline functions, and *maximum* outperformed *weighted average*. We found that (data not presented) the precision differed only slightly between users with high quality and users with low quality. We believe this is because a low quality user would still have good combined beliefs if all of her neighbors had good combined beliefs.

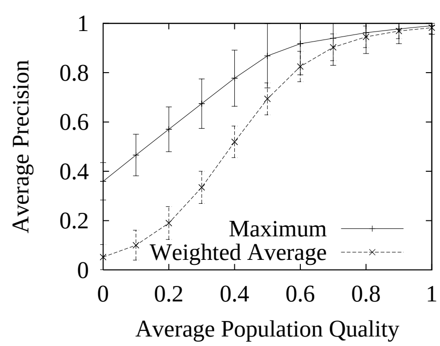

Figure 3: Average precision (±σ) for *maximum* and *weighted average*.

Varying the Population Quality. It is important to understand how the average precision is affected by the quality of the users. We explored this by varying μ, the average population quality (see Figure 3). Overall, m*aximum* significantly outperformed *weighted average* (p<0.01), with the greatest difference at low quality.

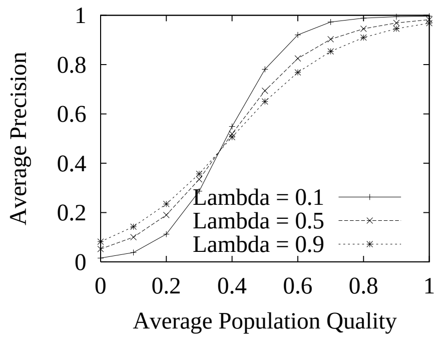

Figure 4: Effect of λ on the precision when combining with *weighted average*.

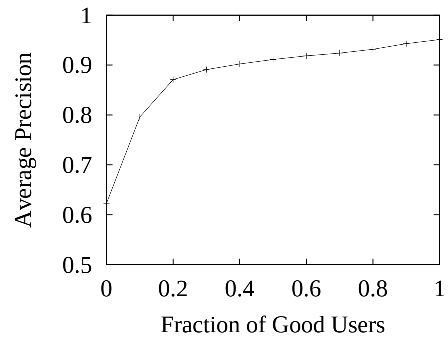

Figure 5: Precision for various fractions of good people in the network, using *maximum* belief combination.

We also explored the effect of varying λ for *weighted average*. In Figure 4, we see that λ had only a small effect on the results. We found that the better the population, the lower λ should be, which makes sense because in this case, the user should put high trust in the population. Because *maximum* seemed to consistently outperform *weighted average*, and has the additional advantage of being cycle-indifferent and producing absolute beliefs, we restricted the remaining experiments to it.

Good and Bad Users. To measure the robustness of the network to bad (or simply clueless) users, we selected user qualities from two Gaussian distributions, with means of 0.25 (*bad*) and 0.75 (*good*) (both had the same standard deviation as earlier, 0.25). We varied the fraction of users drawn from each distribution.

We found the network to be surprisingly robust to bad users (see Figure 5). The average precision was very high (80-90%) when only 10-20% of the users were good. Consider also the network for which the fraction of good people is 0.5. This network has the same average population quality as the network used for Table 1, except in this case the population is drawn from a bimodal distribution of users instead of a unimodal distribution. The result is a higher precision, which shows that it is better to have a few good users than many mediocre ones.

Varying Trust Estimation Accuracy. We also investigated how accurate the trusts must be in order to maintain good quality beliefs. We let the trust noise parameter be the same for all users ($\delta_{ij}=\delta$) and varied δ (see Equation 7). Note that when δ=0, t_{ij} was γ_j, and when δ=1, t_{ij} was chosen uniformly from [0,1]. Figure 6 shows the average precision for various values of δ. Even with a noise level of 0.3, acceptable precision (>80%) was maintained.

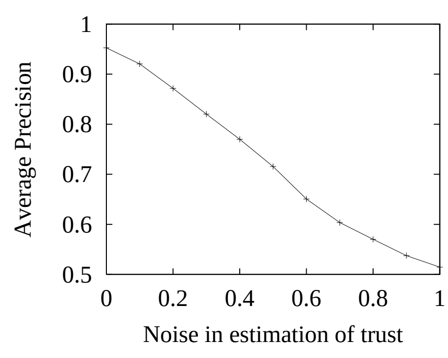

Figure 6: Effect of varying the quality of trust estimation.

The results show that the network is robust to noise and low quality users. Also, *maximum* outperformed *weighted average* in these experiments.

6.2 Experiments with the BibServ Bibliography Server

We have implemented our belief and trust combination methods in our BibServ system, which is publicly accessible at www.bibserv.org. BibServ is a bibliography service that allows users to search for bibliography entries for inclusion in technical publications. Users may upload and maintain their bibliographies, create new entries, use entries created by others, and rate and edit any entry.

Why Bibliographies? We felt that bibliographies have many characteristics that make them a good starting point for our research into the Semantic Web. The bibliography domain is simple, yet gives rise to all of the issues of information quality, relevance, inconsistency, and redundancy that we desire to research. The BibServ beta site currently has 70 users, drawn mainly from the UW computer science department and IBM Almaden, and over half a million entries, of which 18000 entered by the users.

Implementation. BibServ is implemented as a centralized server, so we chose to store the merged trusts $\mathcal{T}$ and compute the merged beliefs as needed. This requires O(NM) space. Since there are many more bibliography entries than users, this is much less than the $O(M^2)$ space that would be required if we instead stored the merged beliefs.

By our definition, a user's merged belief in a bibliography entry represents the quality and relevance of that entry to them. Hence, search results are ordered by belief.[4] The computation of merged trusts and beliefs is implemented in SQL and, in the case of beliefs, is incorporated directly into the search query itself. The overhead of computing beliefs is typically less than 10% of the time required to perform the query itself. Experiments were performed using weighted average (λ=0.5) as well as maximum as belief combination functions.

Belief as Quality and Relevance. The relation of belief combination to BibServ is as follows. When performing a search on BibServ, a user presumably is looking for a good bibliographic entry (e.g. has all of the important fields filled in correctly) that is related to her own field of study. Our concept of "belief" corresponds to this – a good and relevant entry should have high belief. We treat each entry as a statement. Users may set their beliefs explicitly, and we implicitly assume a belief of 1.0 for any entry in their personal bibliography (unless otherwise explicitly rated). This forms the vector **b** for each entry. BibServ users are also presented with a list of other users whom they may rate. A high rating is intended to mean they expect the user to provide entries which are high quality and relevant. This forms the trust matrix **T**.

Experimental Results. We asked BibServ users to think of a specific paper they were interested in, and use BibServ to search for it using keywords. We returned the search

[4] Incorporating traditional measures of query relevance (for instance, TFIDF) may lead to a better ordering of entries. One probabilistic-based technique for this is that of query-dependent PageRank [32].

results in random order, and asked the user to rate each result for quality (0-5) and relevance (either "yes, this is the paper I was looking for" or "no, this is not"). We required the user to make the search general enough to return at least 5 entries, and to rate them all. We used two metrics to evaluate the results. The first is whether there was a correlation between beliefs and either the rated quality or relevance of the entries. In many cases, such as ordering search results, we only care whether the best k results may be determined by belief. We thus calculated the ratio of the average rating of the top k results (ordered by belief) vs. the average rating of all results. Unfortunately, we could do this experiment with only a small number of users. The data set consists of 405 ratings of quality and relevance on 26 searches by 13 users. The average user involved in the study specified 9 trusted users. Because the results are based on a small quantity of data, they should at best be considered preliminary.

The highest correlation was obtained with weighted average, which produced beliefs that had a correlation of 0.29 with the quality ratings (λ=0.03). The other correlations were 0.10 (weighted average vs. relevance), 0.16 (maximum vs. quality), and -0.01 (maximum vs. relevance). These results are not as positive as we had hoped for. Many factors can contribute to a low correlation, such as having little variance in the actual quality and relevance of the entries. Currently, almost all of the entries in BibServ are related to computer science, and all of the users are computer scientists, so the web of trust gives little predictive power for relevance. We expect that as BibServ accumulates users and publications on more varying topics, the correlation results will improve.

The average ratio of the top k results to the rating of all results (across different searches) for relevance ranged from 1.2 to 1.6 for a variety of k (1-5) and for either belief combination function. The average ratio ranged from 0.96 to 1.05 for quality. The ratio rapidly tended toward 1.0 as k increased, indicating that, while belief was a good indicator for relevance, the data contained a lot of noise (making it possible only to identify the very best few entries, not order them all). This is consistent with the low relevance correlation found above.

The most interesting result of these experiments was with regard to λ. We found that the best results when measuring beliefs vs. quality ratings were when λ was very small, though still non-zero. On the other hand, the best results for relevance were when λ was very large, though not equal to one. This indicated that 1) Most users shared a similar metric for evaluating the quality of a bibliography entry, and 2) Users had a widely varying metric for evaluating an entry's relevance. The best λ was not 0 or 1, indicating that both information from others and personalized beliefs were useful.

7. Related Work

The idea of a *web of trust* is not new. As mentioned, it is used by Epinions for ordering product reviews. Cryptography also makes use of a web of trust to verify identity [10]. In Abdul-Rahman's system, John's trust in Jane, and John's trust in Jane's ability to determine who is trustworthy, are separate, though discrete and only qualitatively valued [1]. Such a separation would be interesting to consider in our framework as well.

The analog of belief combination for the WWW is estimating the quality and relevance of web pages. Information retrieval methods based solely on the content of the page (such as TFIDF [20]) are useful, but are outperformed by methods that also involve the connectivity between pages [12][23][26].

Gil and Ratnaker [19] present an algorithm that involves a more complex, though qualitative, form of trust based on user annotations of information sources, which are then combined. One shortcoming of such an approach is that it derives values of "trustworthiness" that are not personalized for the individual using them, requiring all users – regardless of personal values – to agree on the credibility of sources. Secondly, by averaging the statements of many users, the approach is open to a malicious attacker who may submit many high (or low) ratings for a source in order to hide its true credibility. By employing a web of trust, our approach surmounts both of these difficulties (assuming users reduce their trust in a user that provides poor information).

Kamvar et. al's EigenTrust algorithm [21], which computes global trusts as a function of local trust values in a peer-to-peer network, is very similar to our probabilistic interpretation of trusts presented in section 4. One key difference is that we allow trusts and beliefs to vary; they are personalized for each user based on her personal trusts. In contrast, EigenTrust computes a global trust value (similar to PageRank) and emphasizes security against malicious peers who aim to disturb this calculation.

Pennock et. al. looked at how web-based artificial markets may combine the beliefs of their users [29]. Social network algorithms have been applied to webs of trust in order to identify users with high network influence [16][31]. Applying the same methods to the Semantic Web's web of trust may prove fruitful in identifying useful contributors, highly respected entities, etc. Also in a similar vein is the ReferralWeb project, which mines multiple sources to discover networks of trust among users [22]. Also interesting is collaborative filtering [30], in which a user's belief is computed from the beliefs of users she is similar to. This can be seen as forming the web of trust implicitly, based solely on similarity of interests.

8. Future Work

In this work, we assumed that statements are independent. We would like to investigate how dependencies between statements may be handled. For example, if we consider a taxonomy to be a set of class-subclass relationships, and consider each relationship to be an independent statement, then merging such taxonomy beliefs is not likely to lead to a useful taxonomy. We would like to be able to merge structural elements like taxonomies; [14] and [15] may provide useful insights into possible solutions.

The path algebra and probabilistic interpretations were shown to be nearly identical, and the probabilistic interpretation is a generalization of PageRank. Considering PageRank works so well on web pages, it would be interesting to apply the ideas developed here back to the WWW for the purposes of ranking pages. For instance, might we find it useful to replace the sum with a maximum in PageRank? In general, we would like to consider networks in which not all users employ the same belief combination function, perhaps by modifying the global interpretation in order to relax the requirements put on the concatenation and aggregation functions.

There are many tradeoffs between computation, communication, and storage requirements for the different architectures (peer to peer, central server, hierarchical, etc.), algorithms (semi-naïve, Warshall, etc.), and strategies (merge beliefs on demand, store all beliefs, etc.). We would like to formalize these tradeoffs for better understanding of the efficiency of the various architectures.

We considered only single valued beliefs and trusts. In general, a belief could actually be multi-valued, representing a magnitude in multiple dimensions, such as 'truth', and 'importance', and 'novelty'. We would also like to consider multi-valued trusts, such as those used by Gil and Ratnakar [19], which may represent similar dimensions as beliefs (but applied to users). It may be possible to combine beliefs and trusts into one concept, 'opinion', which may be similarly applied to both statements and users. Similarly, we would also like to allow users to specify *topic-specific* trusts. With topic-specific trusts, the normalized sum combination function would probably be similar to query-dependent PageRank [32].

9. Conclusions

If it is to succeed, the Semantic Web must address the issues of information quality, relevance, inconsistency and redundancy. This is done on today's Web using algorithms like PageRank, which take advantage of the link structure of the Web. We propose to generalize this to the Semantic Web by having each user specify which others she trusts, and leveraging this "web of trust" to estimate a user's belief in statements supplied by any other user. This paper formalizes some of the requirements for such a calculus, and describes a number of possible models for carrying it out. The potential of the approach, and the tradeoffs involved, are illustrated by experiments using data from the Epinions knowledge-sharing site, and from the BibServ site we have set up for collecting and serving bibliographic references.

10. Acknowledgements

An off-hand discussion with Jim Hendler at the Semantic Web Workshop at WWW 2002 provided the initial impetus for this work. We also thank Ramanathan Guha for discussions on the web of trust and James Lin for his help with BibServ's site design. This research was partially supported by an IBM Ph.D. Fellowship to the first author, and by ONR grant N00014-02-1-0408.

References

[1] Abdul-Rahman, A., & Hailes, S. (1997). A distributed trust model. *Proceedings of New Security Paradigms Workshop* (pp. 48-60).

[2] Agrawal, R., & Jagadish, H. V. (1988). Multiprocessor transitive closure algorithms. *Proceedings of the International Symposium on Databases in Parallel and Distributed Systems* (pp. 56-66). Austin, TX.

[3] Agrawal, R., Dar, S., & Jagadish, H. V. (1990). Direct transitive closure algorithms: Design and performance evaluation. *ACM Transactions on Database Systems, 15*, 427-458.

[4] Agresti, A. (1990). *Categorical data analysis*. New York, NY: Wiley.

[5] Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1974). *The design and analysis of computer algorithms*. Reading, MA: Addison-Wesley.

[6] Ankolekar, A., Burstein, M. H., Hobbs, J. R., Lassila, O., Martin, D., McDermott, . V., McIlraith, S. A., Narayanan, S., Paolucci, M., Payne, T. R., & Sycara, K. P. (2002). Daml-s: Web service description for the Semantic Web. *International Semantic Web Conference* (pp. 348-363).

[7] Bancilhon, F. (1985). Naive evaluation of recursively defined relations. *On Knowledge Base Management Systems (Islamorada)* (pp. 165-178).

[8] Bellman, R., & Giertz, M. (1973). On the analytic formalism of the theory of fuzzy sets. *Information Sciences*, *5*, 149-156.

[9] Berners-Lee, T., Hendler, J., & Lassila, O. (May 2001). The Semantic Web. *Scientific American*.

[10] Blaze, M., Feigenbaum, J., & Lacy, J. (1996). Decentralized trust management. *Proceedings of the 1996 IEEE Symposium on Security and Privacy* (pp. 164-173). Oakland, CA.

[11] Carre, B. (1978). *Graphs and networks*. Oxford: Claredon Press.

[12] Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Raghavan, P., & Rajagopalan, S. (1998). Automatic resource compilation by analyzing hyperlink structure and associated text. *Proceedings of the Seventh International World Wide Web Conference* (pp. 65-74). Brisbane, Australia: Elsevier.

[13] Chickering, D. M., & Heckerman, D. (1997). Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*, *29*, 181-212.

[14] Doan, A., Madhavan, J., Domingos, P., & Halevy, A. Y. (2002). Learning to Map between Ontologies on the Semantic Web. *Proceedings of the Eleventh International World Wide Web Conference* (pp. 662-673).

[15] Doan, A., Domingos, P., & Halevy, A. (2001). Reconciling schemas of disparate data sources: A machine-learning approach. *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data* (pp. 509-520). Santa Barbara, CA: ACM Press.

[16] Domingos, P., & Richardson, M. (2001). Mining the network value of customers. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 57-66). San Francisco, CA: ACM Press.

[17] French, S. (1985). Group consensus probability distributions: A critical survey. In J. M. Bernardo, M. H. DeGroot, D. V. Lindley and A. F. M. Smith (Eds.), *Bayesian statistics 2*, 183-202. Amsterdam, Netherlands: Elsevier.

[18] Genest, C., & Zidek, J. V. (1986). Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, *1*, 114-148.

[19] Gil, Y., & Ratnakar, V. (2002). Trusting information sources one citizen at a time. *International Semantic Web Conference* (pp. 162-176). Sardinia,Italy.

[20] Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. *Proceedings of the Fourteenth International Conference on Machine Learning (ICML-97)* (pp. 143-151). San Francisco, CA: Morgan Kaufmann.

[21] Kamvar, S., Schlosser, M., & Garcia-Molina, H. (2003). The EigenTrust algorithm for reputation management in P2P networks. *Proceedings of the Twelfth International World Wide Web Conference*.

[22] Kautz, H., Selman, B., & Shah, M. (1997). ReferralWeb: Combining social networks and collaborative filtering. *Communications of the ACM*, *40*, 63-66.

[23] Kleinberg, J. M. (1998). Authoritative sources in a hyperlinked environment. *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 668-677). Baltimore, MD: ACM Press.

[24] Motwani95 Motwani, R., & Raghavan, P. (1995). *Randomized algorithms.* Cambridge University Press.

[25] Ngo, L., & Haddawy, P. (1997). Answering queries from context-sensitive probabilistic knowledge bases. *Theoretical Computer Science*, *171*, 147-177.
[26] Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). *The PageRank citation ranking: Bringing order to the web* (Technical Report). Stanford University, Stanford, CA.
[27] Patel-Schneider, P., & Simeon, J. (2002). Building the Semantic Web on XML. *International Semantic Web Conference* (pp. 147-161).
[28] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco, CA: Morgan Kaufmann.
[29] Pennock, D. M., Nielsen, F. A., & Giles, C. L. (2001). Extracting collective probabilistic forecasts from Web games. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 174-183). San Francisco, CA: ACM Press.
[30] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. *Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work* (pp. 175-186). New York, NY: ACM Press.
[31] Richardson, M., & Domingos, P. (2002). Mining knowledge-sharing sites for viral marketing. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 61-70). Edmonton, Canada: ACM Press.
[32] Richardson, M., & Domingos, P. (2002). The intelligent surfer: Probabilistic combination of link and content information in PageRank. In T. G. Dietterich, S. Becker and Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems 14*, 1441-1448. Cambridge, MA: MIT Press.
[33] Warshall, S. (1962). A theorem on boolean matrices. *Journal of the ACM*, *9*, 11-12.

Appendix

Here we give a proof of Theorem 1. We are assuming $\Diamond$ is commutative and associative, $\circ$ is associative and distributes over $\Diamond$, and $\mathbf{T}$, $\mathcal{T}$, $\mathbf{b}$, and $\mathscr{b}$ are defined as in Section 3. Also from Section 3, $(\mathbf{A}\bullet\mathbf{B})_{ij} = \Diamond(\forall k\colon \mathbf{A}_{ik}\circ\mathbf{B}_{kj})$.

We first prove that $\bullet$ is associative. Let $\mathbf{X}=(\mathbf{A}\bullet\mathbf{B})\bullet\mathbf{C}$. Then:

$\mathbf{X}_{ij} = \Diamond(\forall k\colon \Diamond(\forall l\colon \mathbf{A}_{il}\circ\mathbf{B}_{lk}) \circ \mathbf{C}_{kj})$ from the definition of $\bullet$

$= \Diamond(\forall k\colon \Diamond(\forall l\colon \mathbf{A}_{il}\circ\mathbf{B}_{lk}\circ\mathbf{C}_{kj}))$ since $\circ$ distributes over $\Diamond$ and $\circ$ is associative

$= \Diamond(\forall l\colon \Diamond(\forall k\colon \mathbf{A}_{il}\circ\mathbf{B}_{lk}\circ\mathbf{C}_{kj}))$ since $\Diamond$ is associative

$= \Diamond(\forall l\colon \mathbf{A}_{il}\circ \Diamond(\forall k\colon \mathbf{B}_{lk}\circ\mathbf{C}_{kj}))$ since $\circ$ distributes over $\Diamond$

$= \Diamond(\forall l\colon \mathbf{A}_{il}\circ (\mathbf{B}\bullet\mathbf{C})_{lj})$ by definition of $\bullet$

This implies that

$\mathbf{X} = \mathbf{A}\bullet(\mathbf{B}\bullet\mathbf{C})$ by definition of $\bullet$.

We have $\mathscr{b}^{(0)} = \mathbf{b}$ and $\mathscr{b}^{(n)} = \mathbf{T}\bullet\mathscr{b}^{(n\text{-}1)}$, so $\mathscr{b}^{(n)} = \mathbf{T}\bullet(\mathbf{T}\bullet(\ldots \bullet(\mathbf{T}\bullet\mathbf{b})))$. Since $\bullet$ is associative,

$$\mathscr{b}^{(n)} = \mathbf{T}^{n}\bullet\mathbf{b} \quad (8)$$

(where $\mathbf{T}^{n}$ means $\mathbf{T}\bullet\mathbf{T}\bullet\mathbf{T}\ldots$ n times, and $\mathbf{T}^{0}$ is the identity matrix). We have $\mathcal{T}^{(0)} = \mathbf{T}$ and $\mathcal{T}^{(n)} = \mathbf{T}\bullet\mathcal{T}^{(n\text{-}1)}$, so $\mathcal{T}^{(n)} = \mathbf{T}\bullet(\mathbf{T}\bullet(\ldots \bullet(\mathbf{T}\bullet\mathbf{T})))$. Hence,

$$\mathcal{T}^{(n)} = \mathbf{T}\bullet\mathbf{T}^{n} \quad (9)$$

Combining Equations 8 and 9, $\mathbf{T}\bullet\mathscr{b}^{(n)} = \mathcal{T}^{(n)}\bullet\mathbf{b}$

Since we run until convergence, this is sufficient to show that $\mathbf{T}\bullet\mathscr{b} = \mathcal{T}\bullet\mathbf{b}$.