Modeling Trust in Multi-Agent Systems

Eli Stickgold
Sam Mahoney
Jonathan Pfautz
Joseph Campolongo
Erik Thomsen

Charles River Analytics Inc.
625 Mt. Auburn St.
Cambridge MA 02138
617.491.3474
estickgold@cra.com, jpfautz@cra.com, smahoney@cra.com, jcampolongo@cra.com, ethomsen@cra.com

Keywords:
Human Behavior Models, Belief Management, Agents, Multi-Agent Models, Human-in-the-Loop Modeling, Trust

ABSTRACT: In order to extract information from a complex, multi-agent situation, where hard facts are not readily known, it is necessary to determine how much trust a user should place in each information source report. In order to usefully aid a human in making this assessment, an algorithm must be able to rapidly assign trust values to multiple agents, taking into account a priori assessments by the user based on data that cannot be accurately represented in the model. To this end, we have designed a graph-theoretic algorithm for assessing appropriate trust to place in each agent of a multi-agent situation, based on information provided by the agents, the scale of agreement between this information, using the Katz centrality metric. This algorithm takes user beliefs into account while simultaneously allowing for rapid recalculation to test alternate interpretations and helps users avoid information order bias by computing metrics that remain consistent irrespective of the order in which data is entered. This method serves to make the problem of assigning trust to sources more tractable for analysts, and has the potential to greatly accelerate the process of making accurate decisions.

1. Introduction

In many real-world situations, no information is ever entirely trustworthy and beyond scrutiny. In such situations, making useful action requires the ability to determine what information to trust (Pfautz et al., 2009). This task is generally performed first by analysts, and later by other operational users. As the volume of information available grows, however, it has become increasingly infeasible for this task to be performed without artificial assistance (Guarino et al., 2009; Pfautz et al., 2009).

There are several difficulties that any system for providing this assistance must overcome. First, the basic problem is significantly difficult – in a situation without any known hard facts, it is non-trivial to assess the trustworthiness of an unknown source. Secondly, in order to be of use to an analyst, it must be able to fit within the user’s workflow. Many algorithms that provide only simple, packaged report with a series of assessments of reliability are discounted because they fail to take into account any a priori beliefs the analyst may have that cannot easily be entered into the algorithm, and they do not provide the user with the reasoning by which the algorithm arrived at its conclusions. Finally, if the algorithm achieves rapport with the user by taking user beliefs into account, it needs to be efficient enough to support rapid recalculation to allow the user to test various beliefs and assumptions. The ability to do this allows the user to consider many possible theories and see how they match up to other pieces of known information. Finally, if there is an unreasonable computational cost to altering user beliefs, the algorithm will be untenable.

In this paper, we summarize the existing state of the trust management field, and describe a new method of modeling trust in a multi-agent situation. Our graph-theoretic algorithm is based on centrality metrics on a graph of sources generated by searching reports for instances of conflict and corroboration. It uses a two-step algorithm that pre-calculates a relational matrix that can be applied to any set of a priori beliefs to rapidly calculate overall trust levels, resulting in more efficient recalculation. As an added benefit, the centrality-based method is order-invariant, and does not succumb to the information order bias, and therefore offers significant augmentation to analyst reasoning in information dense situations.

To demonstrate the effectiveness of our new technique, we have implemented it in Charles River Analytics Connect™ network analysis software. We include in this paper a sample analysis of a simple example. Connect provides a platform that allows easy display of
the assessments our algorithm makes, providing additional support for establishing two-way information flow between the user and the algorithm. Finally, we discuss potential future work on our algorithm, both to improve its accuracy and efficiency and to expand it to be able to model internal trust in a social network.

2. Background

In this section, we provide an extended view of the problem space, followed by a review of some previously developed approaches to assigning trust.

2.1 Requirements

In a growing section of modern information gathering, almost all available information comes from information sources of varying reliability – such as sensors, databases, news outlets or individuals. Occasionally it is possible to obtain hard facts, but the presence of such data cannot be assumed. In these situations, it is the responsibility of an analyst to encode information with reliability assessments in the face of inaccurate, padded, or deliberately deceiving reports.

In most of these situations, an analyst will be required to sift through reports from dozens or hundreds of information sources. When action is required, it may be based on information available from only a single source, making it impossible to double-check. This makes it vital to be able to assess the trustworthiness of the source based on prior information that overlaps with subject matter obtained from other sources.

Over time, an analyst must be able to identify information sources whose reliability assessments become key modeling trust. These are sources such that the trusted status of a large number of other sources would be changed if theirs were. If these sources are identified ahead of time it’s possible to make deliberate attempts to test their veracity, improving the accuracy of future trust assessments.

Complicating any attempt to automate this process is the fact that any analyst is likely to have formed their own opinion of each source. These opinions may or may not be based on any concrete facts that can be entered into an algorithm, but nevertheless provide valuable input. Any trust algorithm that entirely discounts these opinions is not only likely to be less accurate, but will also find it difficult to gain the user’s trust, and thus may be discounted entirely.

2.2 Previous Approaches to Determining Trust

There already exist other possible approaches to the issue of determining appropriate trust levels, but they have issues that make them unlikely as candidates for use in this situation. The most common form is Dempster-Schafer Evidence Theory (Shafer, 1976), which arrays possible hypotheses and assigns an evidence mass to each hypothesis, then calculates an upper and lower bound to the probability of each hypothesis based on the mass of evidence explicitly for and against it. Unfortunately, our situation does not lend itself to this method, as Dempster-Schafer and its extensions generally either require a base concept of evidence, which we lack, or only provide trustworthiness values as they relate to each agent in the network (Yu & Singh, 2002). This means that we could know how much source A trusts source B, but not how much the analyst as an outside observer should.

Other, simpler techniques come from fields where algorithms must account for potential inaccuracies of forecasts, rather than of reports. Robotics is the most common source of such algorithms (Gspandl et al., 2011). These algorithms, however, are not easily applicable to our situations for several reasons. Firstly, they generally assume at least some hard truth, gained from the robot’s sensors (which are implicitly trusted). While this assumption is sometimes accurate in real-world situations, it cannot be counted on. Second, inherently, any algorithm intended to account for simple inaccuracies will have difficulties dealing with the possible case of deliberate deception, which is a strong possibility in the real world. In contrast, trust-management algorithms for computer network security (Yan & Holtmanns, 2008) are common, but have the opposite problem: they deal only in deliberate deception, and do not generally consider the problems of inaccuracy or unreliability.

Finally, Bayesian Belief Networks (BBNs) have been used to manage trust in distributed systems (Wang et al., 2006). We considered these methods extensively, as Charles River Analytics has a very mature BBN creation and analysis tool, but ultimately, found them insufficient. While BBNs are able to manage the prospect of deliberate malicious misinformation, they ultimately require at least a kernel of posted evidence to produce trust values. Distributed systems can use reviews or tests to overcome this, but this approach is not necessarily available in a real-world situation.

3. A Centrality-Based Approach to Trust Management

In this section, we present our novel approach to trust management in a multi-agent situation. First, we lay out our reasoning behind using a graph-theoretic
representation of our data. Then, we describe our algorithm of choice for determining trust.

3.1 Graph-Theoretic Representations of Conflict and Corroboration

The nature of the multi-agent trust problem suggests a graph-based approach. A graph approach will give us the bidirectionality inherent in conflict and corroboration (it is not possible to have one-source conflict with another without the reverse being true). As an added benefit, many graph-theoretic algorithms are general enough to have already been implemented efficiently, and also ported to massively parallel architectures. Due to the nature of social networks, one of the driving forces behind graph research, these methods are also generally well adapted to sparse networks. This is valuable, as by the nature of the problem, overlaps between information are relatively rare.

In order to create a network representation of the problem, we must decide what information to represent in vertices and edges. As we delineate in our requirements description, one of the most important tasks this algorithm must be able to assist with is the assessment of a report that has no overlap with other data. In order to determine the veracity of such reports, it is vital that our algorithm be able to determine the trustworthiness of an information source as a whole. As such, we elect to use information sources rather than reports as the vertices of our graph. Each vertex will thus be annotated with a subjected trustworthiness value, which is initialized with the user's subjective assessment. In order to support the analyst's ability to test the effects of altering that subjective assessment, this value must be mutable. To accommodate this, we incorporate the subjective assessment during the trust calculation step itself.

Given this choice, the natural extension is to use edges to denote pairs of information sources that overlap in the bodies of information they have provided. This will result in a sparse network, but as we discussed earlier in this section, that is not an unprecedented difficulty in graph-theoretic algorithms. Because overlaps in information are by nature symmetrical, these edges will be undirected, making our entire graph undirected. We further will need to track the degree of conflict or corroboration represented by each edge, for which we will use edge weights.

There are many possible choices for weighting schema. We chose to use a method based on ratios, rather than absolute values. Considering total conflict or corroboration could potentially have some value in identifying reliability, but runs into difficulties when dealing with malicious adversaries. It is possible, if the analyst uses a scheme like this, that multiple hostile information sources could skew results by submitting large numbers of reports that corroborate with each other. In doing so, they would artificially inflate their own apparent trustworthiness. Instead, we calculate an edge weight, limited to being between -1 and +1, by using the following formula:

$$W_{ij} = \frac{A_{ij} - D_{ij}}{A_{ij} + D_{ij}}$$

Where \(W_{ij}\) is the weight of the edge between vertices \(i\) and \(j\), \(A_{ij}\) is the number of agreements between the two sources, and \(D_{ij}\) is the number of disagreements. This value will be -1 when all overlapping reports from two sources are in disagreement and +1 when they are in perfect agreement.

3.2 Centrality as a Measure of Trust

The concept of centrality (Ferligoj & Kramberger, 1993) is a strong candidate for the type of trust management we require in this situation. In particular, most measures of centrality capture the idea that the centrality of a node in a network should be proportional to the centrality of those nodes it is connected to, weighted by the weights of the edges that connect them. Similarly, when dealing with an algorithm where our information comes primarily in the form conflicts and agreements between sources, we want the trustworthiness of a source to reflect the combined trustworthiness of those sources it agrees with, degraded by an amount proportional to the trustworthiness of those sources it disagrees with.

Of the four most common measures of centrality, we can discard two of them – PageRank and Degree Centrality – as immediately unsuitable, as they discount the contribution of an overlapping source based on how many other sources it overlaps with, which makes sense in the context of centrality in a social network, but not for trust (Newman, 2010). Of the remaining two, we chose Katz Centrality over Eigenvector Centrality, as Katz Centrality allows the user to give an initial weight to each node, which lets us incorporate an analysts underlying belief in the trustworthiness of each source.

In its general form, Katz Centrality assigns each node \(i\) a centrality weight \(x_i\), satisfying the property

$$x_i = a \sum_j (A_{ij}x_j) + \beta_i$$

where \(a\) is scalar representing how strongly we weight the network component of the centrality (dependent on adjacent vertices), as compared to the a priori belief in
the source’s trustworthiness. $A_{ij}$ is the weight the edge between $i$ and $j$, and $\beta_i$ is the inherent centrality of node $i$.

Written for the network as a whole, this may be expressed by the equation:

$$\tilde{x} = \alpha A \tilde{x} + \tilde{\beta}$$

Where $\tilde{x}$ is the vector consisting of the centralities of each node in the network, $A$ is the adjacency matrix of the network, and $\tilde{\beta}$ is a vector consisting of the inherent centrality of each node in the network. Solving for $\tilde{x}$ gives us:

$$\tilde{x} = (I - \alpha A)^{-1} \tilde{\beta}$$

Where $I$ is the identity matrix. If we represent our information as a network of sources each linked by an edge that has a value between -1 and 1 depending on the degree to which the sources conflict or agree, then the Katz Centrality calculated by the above equation will give us a trust vector that is weighted based on trust in conflicting and corroborating sources and integrates an a priori evaluation of trustworthiness. As an added benefit, the most computationally-intensive part of the calculation (taking the inverse) need not be re-computed every time we change the a priori beliefs.

4. Sample Calculation

Using Connect™, we have visualized an example of using Katz Centrality to calculate trust in a simple network with varying values for $\alpha$ and $\beta$.

Consider the following network of sources:

![Figure 1](image1.png)  
**Figure 1, a network of sources with partial overlap in domain.**

Beginning with $\alpha$ set to 90% of its maximum value ($\alpha$ is bounded by the largest eigenvalue of the adjacency matrix) and all a priori values set to 1, running our version of the Katz Centrality Algorithm produces the following result:

![Figure 2](image2.png)  
*Figure 2, trust with equal a priori weights. Shading is on a gradient that goes from trust -5 to trust +5.*

As expected, the algorithm rates Tom, who disagrees with everyone he overlaps with, as untrustworthy. Bill, who has the most overlapping agreement, is the most trustworthy, and George is slightly more trustworthy than Sam, because while he has less agreement with Bill, he also has strong disagreement with a source that has been deemed untrustworthy.

But suppose we had strong reason to believe that Tom is a trustworthy source. Rather than simply setting his trustworthiness high, we can adjust our a priori belief in his trustworthiness upwards and watch what happens to the trustworthiness of the other sources:

![Figure 3](image3.png)  
*Figure 3, Tom’s inherent trust is set to 5, other values are unchanged.*

As expected, Tom is now highly trustworthy, but more importantly, his trustworthiness has resulted in those sources in conflict being rated untrustworthy. Even Sam, who has no direct conflict with Tom, has low trustworthiness, as he agrees completely with Bill. We can also modify the value of the $\alpha$ parameter, which will change how much weight the algorithm gives to the network-based component of the trust value, as compared to the a priori component. For example, in the next two figures, we lower our a priori assessment of Tom’s trustworthiness to 2, leaving the others at one, but in the second we also lower the value of $\alpha$.

![Figure 4](image4.png)  
*Figure 4, Tom’s trustworthiness is now greater than zero, but still less than that of the other sources, as despite our higher a priori belief in his reliability, the*
higher degree of agreement between the other three sources makes them considered more trustworthiness.

![Figure 5](image.png)

**Figure 5.** with $\alpha$ lowered to 50% of its maximum value, Tom is now more trustworthy than the other sources, as the algorithm weights our prior beliefs higher in relation to network-based results than before.

5. Conclusions

Ultimately, we believe that the Katz Centrality-based approach to trust management is extremely well-suited to applications in a multi-agent problem-space. The advantages of this approach are numerous. In this section, we will reiterate these advantages, and discuss potential future research directions that could build off our results.

The first major advantage of our algorithm is that it requires no hard facts, and computes trust values entirely from relative information. It is, however, not limited to including only relative information. The Katz Centrality algorithm allows us to also include information that modifies our a priori trust in certain sources. This also serves to help bring the analyst into the loop, allowing them to test the effects of less concrete beliefs on the resultant analysis.

The method of calculating Katz Centrality offers another important advantage. As all the computationally intensive steps are computed universally for the relative structure of a network, regardless of any analyst beliefs that are subsequently factored in. This allows rapid recalculation of trust values with different a priori beliefs. This further helps integrate the analyst into the calculation. Rather than simply “firing and forgetting,” the analyst is able to see easily how their assumptions impact the algorithm’s results. This allows them to revise these assumptions, and also to identify areas where they are most polarizing, which highlights potential targets for verification.

A final added benefit of our algorithm is its time-invariance. The graph-theoretic approach processes all conflicts and corroborations simultaneously, preventing any false primacy or recency biases. This is doubly important due to the variable speed at which reports may be received depending on how they are being delivered. This has the disadvantage that it requires complete recalculation of the relative trust values whenever we wish to add new information to the network, but there exist many efficient and parallelized matrix inversion algorithms (Galil & Pan, 1989; Hager, 1989), and the added calculation is ameliorated by the aforementioned savings in calculation due to the reusability of relative trust values across multiple sets of a priori assumptions.

There are several directions for future research suggested by this work. Firstly, this algorithm is primarily designed to model trust an outside observer places on various nodes in a social network. A simple extension would be to adapt the algorithm to apply to generating values for how much trust each node has for the others. To do this, we would have to generate a priori beliefs for each node based on their personal conflicts and corroborations. Then a subgraph could be generated without the node in question, and the algorithm could be applied to it.

In the interest of improving the algorithm’s ability to manage a constant data stream, another improvement would be the integration of approximation algorithms allowing low-cost simple recalculations of the relative trust matrix after minor changes to the underlying graph model. Due to the ratio-based edge-weighting, incremental new conflicts and corroborations should cause only minor changes to edge weights. Linear or near-linear approximations (Hager, 1989b) should allow the algorithm to remain nearly current while only performing full recalculations in larger batches.

To allow the algorithm to make more fine-grained decisions, we could expand it to consider multiple categories of reports. This would allow us to assess a trustworthiness value for a given source in a given domain. This approach would require maintaining multiple graphs and potentially increase calculation load, but could provide more accurate assessments in the long run.

Finally, in the domain of providing additional support to the analyst, another valuable service to provide would be sensitivity analysis (Saltelli et al., 2004). This would streamline the process of identifying where verification could be most usefully applied. The current algorithm supports this use, but the analysis must be performed manually. An algorithmic approach would streamline this process.

In conclusion, we believe that the algorithm outlined in this paper provides a novel and valuable tool for calculating appropriate trust levels in multi-agent situations lacking in concrete data.

6. References


**Author Biographies**

**ELI STICKGOLD** is a Scientist at Charles River Analytics Inc., and was the Technical Lead for this effort. His expertise includes machine learning techniques, cognitive models, and network modeling. He has experience developing intuitive model construction tools, and has worked on developing a number of extensions to our graphical agent development environment, AgentWorks.

**JONATHAN PFAUTZ** is a Principal Scientist and Vice President – Cognitive Systems at Charles River Analytics Inc. in Cambridge, MA. His major research interests include cognition and perception-based visualization and user interface design, cognitive work analysis, social network analysis, human behavior modeling, and the applications of cognitive engineering to software system development across a variety of military and non-military domains. He received his PhD from Cambridge U.

**SAM MAHONEY** is a Senior Scientist at Charles River Analytics Inc. Mr. Mahoney’s areas of expertise include system functional decomposition, data processing architectures, and computational knowledge representations. Mr. Mahoney is leading an effort developing a data-driven world model, capable of automated deduction of complex conditions based on incoming sensor information. Mr. Mahoney graduated from Stonehill College with a B.S. in Computer Science.