LNCS manuscript No.
(will be inserted by the editor)

Quantifying Trust

Mariusz Jakubowski[1], **Ramarathnam Venkatesan**[1], **Yacov Yacobi**[1]

Microsoft Research e-mail: {mariuszj,venkie, yacov}@microsoft.com

The date of receipt and acceptance will be inserted by the editor

Abstract *Trust* is a central concept in public-key cryptography infrastructure and in security in general. We study its initial quantification and its spread patterns. There is empirical evidence that in trust-based reputation model for virtual communities, it pays to restrict the clusters of agents to small sets with high mutual trust. We propose and motivate a mathematical model, where this phenomenon emerges naturally. In our model, we separate trust values from their weights. We motivate this separation using real examples, and show that in this model, trust converges to the extremes, agreeing with and accentuating the observed phenomenon. Specifically, in our model, cliques of agents of maximal mutual trust are formed, and the trust between any two agents that do not maximally trust each other, converges to zero.

We offer initial practical relaxations to the model that preserve some of the theoretical flavor.

Key words Trust, collaboration

1 Introduction

There is empirical evidence that in trust-based reputation model for virtual communities, it pays to restrict the clusters of agents to small sets with high mutual trust [6]. We propose and motivate a mathematical model, where this phenomenon emerges naturally. In our model, we separate trust values from their weights (used in a transitive averaging process). We motivate this separation using real examples, and show that under plausible assumptions,

Send offprint requests to: Offprints Assistant
Correspondence to: Address for offprint requests

in this model trust converges to the extremes, accentuating the empirically observed phenomenon. Specifically, in our model, cliques of agents of maximal mutual trust are formed, and the trust between any two agents that have no common max-trust clique converges to zero.

This work is different from authorization languages, such as DKAL-2 ([7]), where the goal is to create a distributed access control, and trust relations are *given* binary relations.

1.1 Trust model

Customarily, we compute the *local trust* between two agents independent of the behavior of other agents. We then feed it into the computation of *transitive trust*, where we compute an average of the opinions of peers. Examples of both stages of trust computation can be found in [6] and in [10]. The former has a very nuanced definition, with attention to details, such as freshness of the data. Ignoring such important details (which should be put back in a real implementation), we propose a bare-bones Information Theoretic definition of local-trust. It is the most general definition, and can be used when the necessary conditional probabilities are known (even a very rough estimate is sufficient). But when they are not available, then standard less general definitions can be used. Our main result about trust is independent of the definition of local-trust. It is a phenomenon of the transitive trust. Our definition of transitive-trust is a slight change to the definition of transitive trust of e.g. [10]. But this change is sufficient to expose the effect.

The *local trust* depends on the gap between behavior and expected behavior of an ideal agent in that role. When we can model behavior as a random variable, the most general way to quantify the gap between two random variables is their conditional entropy. We later specialize this concept for collaborative Web search (where we expect the gap to be a metric). In this case, one user's trust in another may derive from similarity of their queries and preferred search results.

Transitive trust:

In many cases, we can clearly separate weights from trust levels. For example, it is standard practice to ask reviewers of scientific papers to declare their confidence level in their own conclusions. Namely, they are highly trusted to begin with. They are so much trusted that we leave it to them to decide the weight of their own opinions. The same is true when interviewing candidates for research positions. In the case of collaborative Web search, the user may designate context-dependent weights to some persons whom she knows personally (in addition, we may be able to do some automatic weight allocation). When there is no weight data we assign uniform weights.

A trust matrix of n agents is a $n \times n$ matrix $T = (\alpha_{ij} t_{ij})$, where t_{ij} is the trust of agent i in agent j, and α_{ij} is the relative weight that i assigns to the opinion of j. The common-wisdom (e.g., in [10]) is to engineer T to be stochastic. We do not know of a justification to up-front impose this restriction. In contrast we define only the matrix of weights to be stochastic (this is exactly the meaning of weighted average; a convex combination) and allow trust to have any value in the real interval [0,1]. This change is sufficient to unmask the above phenomenon.

2 Trust

2.1 Local trust

Following [18], we model an information-source as a random variable, and use these two expressions synonymously. The amount of information emanating from a source is the amount of uncertainty that existed before the source released the information. This is the Information Theoretic entropy. Let $\mathbb{M}$ denote a message space, and let x, y be random variables over that space. We use x to denote both a r.v. and the agent associated with it. The *conditional entropy* (also, *equivocation*) of y given x is the average uncertainty of y given x, denoted $H_x(y)$. Normalized per bit, its value is in the [0,1] interval[1].

Main thesis: *Local-trust depends on the information-gap between behavior and the expected behavior of an ideal agent in the same role. When we can represent these behaviors as random variables, the most general measure for this information-gap is conditional-entropy (or simple derivatives thereof).*

Suppose r.v. x gets values $l \in \mathbb{M}$ and y has values $m \in \mathbb{M}$. Let $p(l,m) = \Pr[x = l \cap \; y = m]$, and let $p_l(m) = \Pr[y = m \mid x = l]$. Then, $p(l) = \sum_m p(l,m)$ and $p_l(m) = \frac{p(l,m)}{p(l)}$. The entropy of x is defined [18] as $H(x) = -\sum_l p(l) \log p(l)$, and the equivocation of y given x is $H_x(y) = -\sum_{l,m} p(l,m) \log p_l(m)$, which is the same as saying (perhaps more intuitively) $H_x(y) = H(x,y) - H(x)$. We normalize $H_x(y) \in [0,1]$.

Definition 1 *(Local-Trust): Let x, y, z denote 3 r.v. over $\mathbb{M}$, where y represents an agent in some well defined role, whose trustworthiness we try to evaluate, x correspond to an ideal agent in that role (who doesn't lie nor err), and z corresponds to the evaluator's view of x. Then the absolute trustworthiness of y is $1 - H_x(y)$, and the trust of z in y is $t_{zy} = 1 - H_z(y)$.*

Remark 1 The r.v. z, encompasses whatever the evaluator can efficiently compute about x. For example, if y is a consistent liar, then it is a very reliable source of information (like in IT). But we go beyond just bit flipping (as is the case in the Capacity of a binary symmetric channel in IT).

[1] The entropy function is concave, but the equivocation $H_x(y)$ is not necessarily concave.

Following A.C. Yao [22] we allow any polynomial time computation . This is also closely relates to work on Conditional Computational Entropy [14].

Example-1: Suppose that agent y is a revocation authority, and that there are n agents. Each instance of a revocation list is a binary n-tuple with value 1 corresponding to a revoked agent. The message space $\mathbb{M}$ is the set of all 2^n such n-tuples. Each of the 3 r.v. associates some probability to each message.

2.2 Transitive trust

2.2.1 General: We propose a slight modification to existing mathematical model [10]. In the new model (unlike the old) trust converges to the extremes, agreeing with the empirical evidence. We speculate, that this model can further improve the results.

Trust Matrix: Consider a set $\{1, 2, ...n\}$ of agents. In a $n \times n$ *trust matrix* $T = (\alpha_{ij} t_{ij})$, entry (i, j) is the trust of agent i in agent j, denoted t_{ij}, weighted by some weight factor $0 \leq \alpha_{ij} \leq 1$, where for all i, $\sum_{i=1}^{n} \alpha_{ij} = 1$. α_{ij} can be interpreted as the *relative* relevance of the opinion of a peer (for example, a peer may be fully trusted but claim little confidence about some specific evaluation; the condition $\sum_{i=1}^{n} \alpha_{ij} = 1$ is the usual meaning of weighted average). Occasionally we use the uniform weight $1/n$ as an example, but our claims hold for any convex combination. When the discrete time $\tau = 1, 2, ...$ is necessary for the explanation we write $t_{ik}(\tau)$ instead of t_{ik}. We assume that for all i and τ, $t_{ii}(\tau) = 1$. [2] It is natural to normalize the trust values to the interval $0 \leq t_{ij} \leq 1$, since we expect $0 \leq t_{ij}(\tau) t_{jk}(\tau) \leq 1$. This is similar to EigenTrust [10], but with important difference. We do not engineer T so that for all i, $\sum_{j=1}^{n} \alpha_{ij} t_{ij} = 1$ (in fact, we do not know of any justification for such constraint).

Definition 2 *A* maximal trust matrix *is a trust matrix where every agent has trust=1 in every agent (i.e. for uniform weight, the matrix T is all $1/n$).*

Interpretation of right eigenvector of T:

Consider the $n \times n$ trust matrix, T of agents $1, 2, ..n$. Assume a candidate 0 to this set. The agents $1, 2, ...n$ are existing set members. Each of them interviews the candidate to determine a local trust value. Interviewer i has local trust value t_{i0} in candidate 0. Let $t = (t_{10}, t_{20}, ...t_{n0})^{\mathrm{T}}$. Right multiplying Tt yields the transitive trust values after one iteration. The right eigenvector, corresponding to eigenvalue 1, is the stable transitive trust values of the existing set members in the candidate.

[2] When trust is based on similarity, as is the case with trust-based collaborative Web search, we do not have to assume $t_{ii} = 1$. It follows from the definitions.

Interpretation of left eigenvector of T : Row i represents the trust of agent i in each of the set members, and column j represents the trust of each set member in agent j. Let $t^{\mathrm{T}}(\tau)$ denote a row vector whose entry $j = 1, 2, ...n$, is the aggregate trust of *existing* set members in *existing* agent j at discrete time τ. Then $t^{\mathrm{T}}(\tau+1) = t^{\mathrm{T}}(\tau) \cdot T(\tau)$. Therefore a left eigenvector that corresponds to eigenvalue 1, is a stable trust vector representing the overall trust of the set in each of its *existing* members[3].

2.2.2 Modes of clique build-up Along the time axis, the process is dynamic;

cliques may grow, then split (when facing new data).

Definition 3 *We use the term* gradual *clique build-up when referring to a dynamic process, using right multiplication,* $t(\tau+1) = T(\tau)t(\tau)$, *where current data applies to current candidates to a clique,* $t(\tau)$, *but existing clique members, that were accepted under older data, are not judged again under the newer data. If all the agents represented by* $T(\tau)$ *are evaluated using the data available at time* τ, *then we call it* instantaneous *clique build-up.*

So, $T(\tau)$ has different interpretation, depending on the mode of clique build-up. In the instantaneous clique build-up, the data at time τ is used in all the entries of the matrix, and in the gradual mode of clique build-up, entries are added gradually, and once added they are not re-evaluated under newer data. the left (right) eigenvector corresponding to eigenvalue 1 is the stable solution in the instantaneous (gradual) clique build up.

2.2.3 The Perron-Frobenius Theory The part of the theory that we actually use here appears e.g. in [17], Theorem 1.1, part (e). For a concise summary of the theory see also Th. 1.3.1 in Andries Brouwer's notes[4]. Let T be any matrix over $\mathbb{R}$ (a vector is a special case). $T > 0$ means that every entry of T is positive (the notation $T \geq 0$ should also be interpreted likewise). A matrix $T \in \mathbb{R}^{n\times n}$ is *primitive* if $(\exists k)[T^k > 0]$. It is *irreducible* if $(\forall i,j)(\exists k)[(T^k)_{ij} > 0]$ (the corresponding digraph is strongly connected, i.e. $\exists$ path from any node i to any node j). We present here only the part of the theory that we need now.

Theorem 1 *(Perron-Frobenius): Let* $T \in \mathbb{R}^{n\times n}$ *be irreducible. There exists* $\theta_0 \in \mathbb{R}$ *such that* $\theta_0 = \rho(T)$ *is the spectral radius of* T, *and if* $0 \leq S \leq T$ *and* σ *is any eigenvalue of* S *then* $|\sigma| \leq \theta_0$. *Furthermore,* $|\sigma| = \theta_0$ *if and only if* $S = T$.

[3] If T is a maximal trust matrix, then after one iteration $t(\tau)$ is necessarily a consensus. Since $T^2 = T$, T is also a projection. It projects onto $\mathcal{U}$ along $\mathcal{V}$, where $\mathcal{U}$ is all the consensus vectors, and $\mathcal{V}$ is all the vectors whose components add up to zero. The minimal polynomial of T is $x^2 - x$, whose 2 roots are the eigenvalues $\lambda_0 = 0$ and $\lambda_1 = 1$. Every consensus vector is an eigenvector corresponding to λ_1.

[4] http://www.win.tue.nl/~aeb/srgbk/node4.html

Remark 2 Let T represent a max-trust clique. As such T is stochastic, hence its spectral radius is $\rho(T) = 1$. For any $S < T$, $\rho(S) < 1$ (by the above clause of the Perron-Frobenius Theorem). So, $\lim_{k\to\infty} \rho(S^k) = 0$. This is true not only when using uniform weights 1/n, but for any convex combination.

Remark 3 A trust matrix which is not max-trust has only the zero vector as eigenvector.

Remark 4 Cliques of maximal trust are formed (they may overlap). The trust between two agents that do not maximally trust each other converges to zero (because any matrix S that includes both, is $S < T$).

3 Practical considerations

In the theoretical part of the paper we used the term "clique," while in the practical part, where we allow tolerances, we switch to the murkier term "cluster." Classic clustering distinguish agglomerative (bottom-up) from divisive (top-down) clustering. Both methods are useful even when data is static. In our processes data is dynamic. At each point in time, we can freeze the data and use either one or the other clustering methods, and achieve the same eventual structure.

The theoretical process is all-or-nothing. It forms max-trust cliques, and the trust between any two agents that do not maximally trust each other vanishes. This is useful as a general guideline, but in practice we have to do useful things with less than perfect trust. Allowing tolerance β means accepting a candidate with less than perfect score $t \in [1-\beta, 1]$. We hence switch from the *clique* terminology to the murkier but more realistic *cluster.* In this case (if at least one entry $t_{ij} < 1$) the only eigenvector of the trust matrix of the cluster is the all zero vector (from the Perron-Frobenius Theory). So, we can no longer use eigenvectors. Instead we want to try a related process, which coincides with the theory for zero-tolerance ($\beta = 0$).

Definition 4 *The* cohesion *of a cluster of agents is the smallest trust between a pair of agents in the cluster.*

It is convenient now to consider right-multiplying the trust matrix T of a cluster, by a column vector $t(\tau)$, representing the trust of each of the existing cluster members in a new candidate, at discrete time τ. So,

$$t(\tau + 1) = Tt(\tau).$$

Based on empirical evidence of [6] and in agreement with our mathematical model, we opt to try relatively high cohesion clusters of agents.

In the theoretical case, when T is max-trust, then one iteration of $t(\tau + 1) = Tt(\tau)$ yields a stable solution (an eigenvector corresponding to eigenvalue one). Therefore in the practical approximation we also consider only one iteration. If all the entries of T and of $t(\tau)$ are in $[1-\beta, 1]$, then, as we show in the full paper, all the entries of $t(\tau+1) = Tt(\tau)$ are in $[1-2\beta, 1]$. The fact that we can decide this outcome through a simple inspection, rather than through a more involved computation does not change this error-uncertainty analysis[5].

The algorithm tries to create near max-trust clusters within these tolerances, by accepting or rejecting candidates to existing clusters, with known tolerances, or by splitting existing clusters and accepting to sub-clusters. For $\beta = 0$ this algorithm is consistent with the theoretical model; it gradually creates strict max trust cliques.

Example-2: Usually credit card companies defer to a few credit bureaus to decide the trustworthiness of clients. This corresponds to transitive averaging in which the credit card companies, and merchants concentrate all the weights on the credit bureaus, and assign zero weight to the opinions of users (even when maximally trusted).

Example-3: Trust relations among pieces of code (in the cloud and on a single machine). Software modules establish trust relations based on their vendor's attestation, trust relations between vendors, and trust relations between users and vendors. Near max-trust clusters of modules are allowed to work together. Lower trust vanishes. Clusters are separated from each other, so they cannot interfere.

Example-4: Auction sites such as eBay support a ranking system with which buyers and sellers evaluate one another. Participants score transactions based on aspects such as product quality, shipping duration, timely communication, and overall satisfaction. Long-time, successful buyers and sellers tend to migrate into a pool where everyone has perfect or near-perfect rankings, while others generally fall away and never return. If we equate trust with average ranking, this example shows convergence into max- and min-trust cliques.

Example-5: There is empirical evidence [6] that in trust-based reputation model for virtual communities, it pays to restrict the clusters of agents to small sets with high mutual trust.

Acknowledgement by Yacov Yacobi: I thank Jim Kajiya, who encouraged and supported this research, Yuval Peres, who pointed me to the Perron-Frobenius Theory, and Sara Wyckoff, for educating me about anthropological aspects of trust.

References

1. A. Kraskov, H. Stogbauer, R.G. Andrzejak, and P. Grassberger, Hierarchical Clustering Based on Mutual Information,

[5] The lower bound on error propagation of function f is the minimum over all possible algorithms that compute f.

2. A.N. Langville, and C.D. Meyer, Google's PageRank and Beyond: The Science of search engine ranking, Princeton University Press, 2006, ISBN - 13: 978-0-691-12202-1,
3. R. Cilibrasi and P. M.B. Vitanyi, Clustering by compression, IEEE Trans. IT, Vol. 51, No. 4, April 2005, 1523-1545.
4. E. Balfe, B. Smyth, An Analysis of Query Similarity in Collaborative Web Search, in D.E. Losada and J.M. Fernandez-Luna (Eds.): ECIR 2005, LNCS 3408, pp. 330-344, 2005. Springer-Verlag Berlin Heidelberg 2005.
5. Eric J. Glover, Gary W. Flake, Steve Lawrence, Andries Kruger, David M. Pennock, William P. Birmingham, C. Lee Giles, "Improving Category Specific Web Search by Learning Query Modifications," saint,pp.23, 2001 Symposium on Applications and the Internet (SAINT'01), 2001
6. N. Gal-Oz, E. Gudes, and D. Hendler, A Robust and Knot-Aware Trust-Based Reputation Model, In IFIPTM 2008, Joint iTrust and PST Conferences on Privacy, Trust Management, and Security, June 18-20, 2008, Trondheim, Norway.
7. Y. Gurevich, Itay Neeman: DKAL2—A simplified and Improved Authorization Language. Feb. 2009.
8. A. Josang, An Algebra for Assessing Trust in Certification Chains, NDSS'99.
9. Rainer Hegselmann and Ulrich Krause, Opinion Dynamics and Bounded Confidence Models, Analysis, and Simulation, in *Journal of Artificial Societies and Social Simulation (JASSS) Vol. 5, No. 3, 2002.*
10. S.D. Kamvar, M.T. Schlosser, and H. Garica_Molina: The EigenTrust Algorithm for Reputation Management in P2P Networks,
11. W. Ren, R.W. Beard, E.M. Atkins: A Survey of Consensus Problems in Multi-Agent Coordination, 2005 American Control Conference, June 8-10, 2005. Portland, OR USA, pp. 1859-1864.
12. C.D. Manning, P. Raghavan, and H. Schütze, An Introduction to Information Retrival, Cambridge University Press, 2008.
13. M. Najork, S. Gollapudi, R. Panigraphy, Less is More: Sampling the Neighborhood Graph Makes SALSA Better and Faster. WSDM 2009 Barcelona, Spain.
14. Chun-Yuan Hsiao, Chi-Jen Lu, Leonid Reyzin: Conditional Computational Entropy, or Toward Separating Pseudoentropy from Compressibility. EUROCRYPT 2007: 169-186
15. P. Resnick and R. Zeckhauser, Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System,
16. T. Sander and C.F. Tschudin, Protecting mobile agents against malicious hosts, in G. Vigna (Ed.): Mobile agents and security, LNCS 1419, pp. 44-60, 1998. Springer-Verlag Berlin Heidelberg 1998.
17. E. Seneta: Non-negative Matrices and Markov Chains, Springer-Verlag, 1980, ISBN 0-387-90598-7,
18. C. E. Shannon: A mathematical theory of communication. Bell System Technical Journal, vol. 27, pp. 379–423 and 623–656, July and October, 1948
19. Claude E. Shannon, Communication Theory of Secrecy Systems, Bell Labs Journal 1949.
20. Gilbert Strang, Linear Algebra and its Applications, 4th ed. Thomson, ISBN 0-03-010567-6
21. R.S. Varga: Matrix Iterative Analysis, Prentice Hall, Inc. 1962, Library of Congress Catalog Number 62-21277.

22. A.C. Yao: Computational Information Theory, In Complexity In Information Theory, 1988, pp. 1-15.

4 Appendix:

4.1 The curious incident of metrics "passing through" transitive averaging functions

Metrics that "pass through" transitive-averaging functions cease to be metrics, but if we iterate sufficiently many times (until they converge to the extremes) they again become metrics.

Let $D(u_i, u_j) = g_{ij}$ be the local distance between agent i and agent j. Suppose that g_{ij} is a metric, i.e. $(i)\ \ g_{ij} \geq 0,\quad (ii)\ \ g_{ij} = 0 \Leftrightarrow i = j$, $(iii)\ \ g_{ij} = g_{ji},\quad (iv)\ \ g_{ij} \leq g_{ik} + g_{kj}$. The gaps resulting from a convex combination of trust values based on such direct gaps is not necessarily a metric. For example, plug these numbers into a uniformly weighted consensus function:

$$1 - G_{ik} = \frac{1}{2}[(1 - g_{ik}) + (1 - g_{ij}g_{jk})],$$

In general, G_{ik} is not a metric. For example, $G_{ii} = \frac{1}{2}g_{ij}g_{ji}$ is not necessarily zero. However, when trust converges to the extremes, it is a metric. Trust of i in j is $t_{ij} = 1 - G_{ij}$=1, and as a special case, $G_{ii} = 0$. The trust between two agents that are not in the same clique is zero. In this case, $(G_{ij}, G_{jk}, G_{ki}) \in \{(0,0,0), (1,1,1), (0,1,1)\}$. This is a metric.