

A couple months ago I was reading a paper called [Flow-based reputation with uncertainty: evidence-based subjective logic](#). I think it's a pretty unexamined piece of research for what it presents, which is a reputation algorithm that can compute across arbitrary trust networks

. I'd love to get others opinion of this in the light of Sybil-resistant design - for which PoW-style control mechanisms look to be the only currently effective approach.

To give a brief overview of EBSL:

- Subjective Logic is an algebra for computing "subjective" qualities. It deals with opinions, which are tuples of the form (belief + disbelief + uncertainty) = 1

, and has a couple operators such as discounting (used for transitive trust) and fusion/consensus (for combining opinions).

- SL can be used to model trust networks, where relations are defined as opinions of trustworthiness. Given a set of opinions, a fixed-point convergence algorithm (similar to PageRank) can be used to arrive at a reputation matrix. However the algorithm for computing the flow of trust is flawed (see more in the EBSL paper).
- Evidence-Based SL (EBSL) solves this issue and also introduces a more natural primitive of evidence. I'm not sure of the terminology of domains, but opinion components range 0-1 and evidence is just an integer (so it's an absolute rather than relative metric, if that makes sense). They define a mapping between the two, and also introduce discounting/consensus operators that can converge trust flow over arbitrary network structures.
- You can give evidence or an opinion to the convergence algorithm, although the former I'd argue is more organic. The original simulation used evidence amounts between nodes to compute the reputation, which in modelling P2P downloads for example (as in the original EigenTrust) scales better than adjusting a reputation opinion for every node.

I've reimplemented the algorithm [here](#) (logic in ebsl/lib.py

), based off of the code that the authors kindly forwarded me (also in the repo). In simulations/main.py

, I've done some work simulating Sybil networks and how they can inflate their rep in the eyes of other users. In a highly-connected network of honest nodes and a Sybil network of relations, the honest network's perception of Sybil's reputation scales only linearly

with every honest node you convince.

I'd like to share and encourage others to experiment - I'm not a trained mathematician by any means, more of a hacker - but it's the first model I've seen which seems to be Sybil-resistant and be able to handle reconfigurations of trust networks

in a versatile manner.

What's possible is modularising reputation providers, such as Uber/Lyft. Let's model Uber as a reputation network, wherein we trust them to administer/bootstrap the initial reputation of riders/drivers. Riders get their initial reputation by tying their identity to a phone/credit card no., and drivers also undergo initial license checks etc. Thereafter the usual mutual rating system occurs, and the system can filter out bad actors.

In a decentralized version, we would still have a reputation provider 'dUber', but their role would be reduced to simple attestations of identity, license checks, etc. Riders/drivers could require a proof-of-location for their rating of an exchange to be counted (since it's a physical exchange). Modelling interaction, all drivers/riders in the trust graph are connected by their edge to the dUber org, trust flows and we can get the trust of users we haven't interacted with.

Because dUber is just another edge in the graph, we can replace/augment our worldview with information from other providers/networks (ie. Lyft). If they behave maliciously, we can adapt the network without the usual inertia of large reconfigurations of social relationships.

I think this could be an interesting orthogonal to proof-of-stake/work and general consensus algorithms, and just want to get some discussion going in any case.