PhD Comprehensive Exam

Terrell Russell
University of North Carolina at Chapel Hill
School of Information and Library Science
Thursday, January 28, 2010

Question 4 – Windley

Please do the following:

a.) You describe a number of reputation systems and models. Determine common components of and criteria for classifying reputation systems based on the models and systems you have reviewed. List these with justification for their inclusion.

b.) Use the information in (a) to determine common patterns and to develop a model of reputation systems. Use it to classify and describe at least three different reputation systems. The model should be sufficiently abstract to be useful for describing and analyzing the three systems you choose. Any of a graphical, formulaic, or algorithmic model is acceptable.

c.) In your literature review, you distinguish between the social personas that we put on as we interact with others and outright deception. You mention that society has created legal frameworks–worked out over centuries–to allow the former and restrict the latter. Given this, any automated system is bound to have flaws. Describe at least three failure modes for your model and the reasons that they are present. Can they be mitigated? How?

Response

1 Introduction

Reputation is a multi-headed creature. It exists both as a good thing and as a bad thing. It can be rooted in truth or rooted in falsehood. It is both a collective judgement and an individually derived opinion. As such, modeling reputation is a messy, hard to specify, never-ending process. But I'll give it a try.

In section 2, I'll lay out what I see to be the similarities and facets of reputation systems today. In section 3, I'll be more specific and apply these components to the systems underlying Slashdot, PageRank, and BitTorrent. In section 4, I describe how my proposed system could fail and how to recover from those failures.

2 Components and Criteria

As reputation has been studied and systemized, a few common components and facets have presented themselves as necessary to take into consideration.

First off, any system must be measuring the reputation of a particular unit. What this unit is determines much about how the system will be realized. Some systems are based on people, others on documents, and others on organizations or the reputation claims themselves.

That said, most important, from a design perspective, is whether the system is calculating a global or localized reputation "score." Whatever is being calculated, if it is to be a representation of an object from the perspective of "the system," then it must be globally shared and accessible by the entire system. This is most easily done in a centralized system where a single codebase or algorithm is both determining a score and storing it for further access and distribution. A distributed system is much more complicated to engineer and police, but also, more robust in the case of failure or infiltration.

A second criteria would be the nature of the information being stored – whether it is of a deterministic nature or cognitively generated. This component is important as to the level of interaction humans must have with the system to run the system. As a completely automated, generative algorithm can be run many times, very quickly, it may be missing the ability to change over time in the ways that a human-based system may find natural or easy.

A third component of reputation systems is their level of recursivity and transitivity. Systems that are global may integrate some amount of dampening or multiplication within an algorithm when determining reputation scores, but have little with regards to recursive programming. Distributed systems usually require participating actors to communicate amongst themselves both their scores and some additional information regarding "hops" or "TTL".

Some systems take measurements on one type of item and use those to calculate a score for some other item, as an aggregation or summarization score. Other systems are actually evaluating the item of record directly. Direct systems are much easier to conceptualize and follow algorithmically, but may provide very simple outputs and less insight into the nature of the thing being evaluated. More complex models are harder to get right, but may convey more meaning when they produce output.

Transparency is a component that is key to understanding a reputation system. The double-edged sword, of course, is that, with transparency, insights into the ways that calculations are made comes clarity into how to game the system and provide unfair advantages to some over others. Transparency is usually evaluated on a spectrum, as having full or no transparency is rarely the best option.

Lastly, some amount of internal scoring and ranking may happen between nodes of a system to help calculate confidence or reliability. If a system can be infiltrated, the individual participants must have a way to ignore or punish misbehaving actors or the system will become completely useless very quickly.

These seven elements cover the most interesting and salient features of most reputation systems.

3 Model

The components listed above suggest a straightforward mapping activity between them and any particular reputation system. In this section, I have mapped the seven components to three systems that are well-known and well-studied: Slashdot's Karma system, Google's original PageRank algorithm, and BitTorrent's node selection and performance algorithm. For ease of organization, I have created Table 1 as an overall mapping and will discuss each system subsequently.

Component	Slashdot's Karma	Google's PageRank	BitTorrent
unit	account (person)	document	peer node
global/local	a single score for each account	a single pagerank score for each document	scores are calculated at the node level and shared openly
rational/cognitive	cognitive (human)	strictly algorithmic	strictly algorithmic but with knobs available to node operator (human)
recursivity/transitivity	scores are not propagated between accounts	high value documents provide greater juice	scores are reported during discovery
direct/summary	based on comments	based on links	direct observation
transparency	metamoderators can see moderation data and the underlying code is open	specific code is closed but the basic algorithm is well known	code is open and flowing data is completely visible
reliability/confidence	confidence based on a few trusted humans	based largely on result quality	high confidence based on visibility into the data

Table 1: Mapping of Reputation System components

3.1 Slashdot's Karma System

The technology news and discussion website Slashdot was created in the mid-1990's and was the center of web and nerd technology discussion and expertise for a few years. Due to its popularity, the discussion forum quickly became a place where comment "trolls" began to appear and bring down the level of quality discourse for everyone. A Karma system was devised and rolled out to great fanfare and allowed readers to rate, on a 1-5 scale, the usefulness, interestingness, or humor in each comment. As each comment was written by an account (or by "Anonymous Coward"), those ratings could be aggregated to generate a Karma score for each account. As well, there were moderation points available to users to help police the ratings being issued to comments. As more people participated, the system began to strain as the trolls began to moderate other trolls' comments highly. This generated demand for a role of meta-moderator to be created whereby the moderation evaluations themselves could be policed by a subset of known, well-behaved, and trusted users (accounts). Of course, this system is also open to gaming, but requires a significantly greater effort and subterfuge on the part of the trolls to gain access and then manipulate the meta-moderation system.

As a reputation system, Slashdot Karma can be classified as a global system that evaluates user accounts based on human judgements of comments. Account scores do not affect other account scores, except in the sense that highly rated users are sometimes invited to participate at the moderation and meta-moderation level. Transparency is limited as the system does not report all of its internal scoring and measurement at all times. There are roles for different users and they can see increasing amounts of moderation and rating information. The code that runs Slashdot's Karma system is open source, so the underlying belief that the system is behaving as it was designed is pretty high. As a reliable system for identifying spam and trollish behavior, Slashdot has proven effective. As an early mover in the web technology discussion realm, it fared very well, but has lost some of its cache (and high quality user activity) as newer systems have come online; unfortunately, the technologists'

collective attention span is not very forgiving.

3.2 Google's PageRank

Google's PageRank algorithm was originally developed in the late 1990's at Stanford and was based on the manual work that had been done earlier with citation counts and bibliometrics (Garfield). It served as an automatic way to rank the authority or impact of a document based not on the information within the document, but rather the information stored in the links referring to the document. On the web, these documents were webpages and the hyperlinks between them served as the "votes" of confidence. Links were not capable of carrying positive and negative sentiment, but the system performed well enough to boost the effectiveness of the returns within the Google algorithm to help make it the most popular and profitable search engine today.

The PageRank system today takes into consideration many more elements than just word count and the number and quality of inlinks to a webpage. However, these elements still serve as the backbone of today's algorithm and are worth studying in their own right. PageRank itself is represented as a global score for a document; a page has a PageRank score. This score is determined strictly algorithmically, but is based, of course, on non-affiliated human linking behavior on the open web. The transitive nature of the algorithm is one of the key factors for making it successful. A webpage with high PageRank passes along a portion of its value to every page that it links to. Pages that are pointed to by other high-value pages are themselves, by definition, of high value. As webpages are themselves comprised of links, the directness of the algorithm is pretty straightforward and there is little indirection or summarization going on somewhere else. Transparency regarding the PageRank algorithm is very high as it is well-known and well-studied at this point. Google's specific implementation of it is very secret and they, no-doubt, have many tools in place to detect and punish pages that try to game the system (linkfarms that collectively point to many other pages owned by the same entity). The reliability of the system is innately visible

in the search results that continually come back from issued queries. As long as the quality remains high and the desired results are found, confidence in the system remains high.

3.3 BitTorrent

The BitTorrent file-sharing network is a distributed series of nodes that communicate with one another without a central hub. It was created in response to the takedown of the centralized file-sharing service Napster. Napster also allowed for file-sharing directly between users of the system, whereby the files being shared did not go through the Napster servers themselves. However, all discovery and routing information *did* go through Napster's servers and was therefore a ripe target for litigation and eventual takedown. BitTorrent itself does not have a central server and is technically just a protocol – a specification for how file-sharing nodes can talk directly to one another and share information about files and other nodes. The key for this discussion is the information being shared about the other nodes.

As a reputation system, BitTorrent provides information about peer nodes and their recent performance on the network. By sending information about the recent network performance, other nodes can better make decisions about which nodes to talk to moving forward. As a distributed system, the calculations being made by each node are obviously not global. Each node receives information about other nodes from its own peer nodes and then pools that information and calculates a score for each node it sees. When asked, the node will send that information to other nodes, so they can do the same. This allows for a modular, distributed, flexible, and continuously current system. The calculations are strictly algorithmic but are based on variables set by each node operator. Most operators do not change their defaults as these controls are esoteric and the defaults are tweaked by the software creators as best practices become clear. The system is a transitive system by nature as local scores are propagated to peer nodes and then serve as inputs for other calculations. Since traffic information is directly recorded and then reported, there is little summarization happening that is not empirically derived. The BitTorrent code is open source and highly scrutinized

for malicious attempts to detect or report activity in a manner outside the agreed upon protocol. It is therefore a very trusted system by its users and poorly behaving clients are quickly ignored/blackballed. The confidence in the reputation system between the nodes is based on the public visibility of the data being the source of the calculations. As lying clients are quickly identified by being out of step from their own peers' reportings, gaming the system is very hard and not often successful.

4 Failure Modes

I will answer this part of the essay with respect to my proposed Contextual Authority Tagging (CAT) research. Discussing specifics about potential failure modes is useful only at the specific system level. As I am currently proposing this research as a centralized system, these answers relate to the possibilities inherent in centralized systems. Distributed systems may not have the same means of recovering from failures.

4.1 No Opt-Out

The first failure mode that presents itself when considering a mass system for formalizing and counting gossip/opinion is that participants being discussed have no way to opt-out. We live in a world where humans have free-will and can do and act as they please (within reason). If they want to talk about one another, it is their prerogative and many choose to do so. Valuable information is passed from person to person, and in large part, there is little limitation to the kind of information that can be passed. We do have laws limiting the publishing of false information about others, but opinion information is largely without regulation (as it should be). We do not have anonymity among those we know. With a system like CAT, that lack of anonymity is shared more broadly.

If CAT has no opt-out, then there could be malicious information stored and propagated in the system. As I have proposed, the information coming out of the system during normal

usage is anonymized and not traceable to an individual "tagger". But, in a centralized system, the records themselves could keep track of who said what, and therefore be available for inspection if a claim of wrongfulness was leveled by a victim. If a claim was brought, then the central authority could simply check the database and let the courts answer a question of libel or slander in their usual manner. There is nothing new here except the fact that the publishing of the possible falsehood was carried out by a database rather than in a tabloid or other less-than-reputable information source.

Like most matters before the courts, the cost of bringing the suit compared to the damages incurred by the victim would largely determine whether the claim is taken seriously or makes it onto a docket.

4.2 Mistaken Identity

As a possible failure mode, certainly a system based on identity would need a way to recover from cases of mistaken identity. If someone tags person A while thinking they were tagging person B, there needs to be a way to rollback the action, at minimum, or correctly reassign the tag, at best. While the first scenario dealt with largely malicious activity, this deals more with honest mistakes. Both are bad data, but their recovery paths are significantly different.

With misidentification, the error will probably be detected by the taggee or someone very familiar with the taggee, if it is detected at all. As a system that protects anonymity, the taggee really has no recourse directly. If the information is truly false and potentially damaging, they may have a course of action mentioned earlier. Otherwise, the best hope is that the tag in question does not get much traction among the person's other peers. As the visibility of the tags are ranked by occurrence, tags that are mentioned only once or a few times will probably be seen as noise in any sufficiently large system. With tagging at del.icio.us, there are no typos near the top of the list of commonly used tags. This is because, like an issue surrounding misidentification, these tags are not being used by others and therefore are buried deep within the data (if shown at all). The mitigation for honestly

misapplied tags is largely to have a bigger, more active system.

4.3 Persistence

One more failure mode could be considered the persistence built into the system. As a feature, this allows trending to be visible and analysis to be done regarding the knowledge interactions over time. As a bug, a system with a long memory could hold onto unwanted tagging information and influence future interactions for the taggee.

This is a less offensive failure mode as it is part of living in a *networked public*. We do not yet have good ways to mitigate this other than to "wait it out" and hope that newly generated data will overwhelm and either water down or completely flood older data. Surely one of the only things we can really be sure of is that we will continue to generate more and more data as we move forward.