Informativeness and Incentive Compatibility for Reputation Systems

A Thesis presented

by

Jie Tang

To

Computer Science
and
Economics

in partial fulfillment of the honors requirements

for the degree of

Bachelor of Arts
Harvard College
Cambridge, Massachusetts

April 1, 2008

Abstract

Reputation systems, which rank agents based on feedback from past interactions, play a crucial role in aggregating and sharing trust information online. Reputation systems are used to find authoritative web sites and ensure socially beneficial behavior on auction sites. The main problem faced by reputation system researchers is a lack of good metrics for comparison and evaluation. This thesis defines a novel "informativeness" metric for reputation systems which happens to approximate a crucial economic efficiency metric. This is then applied to the problem of finding optimal reputation systems. We show empirically that this metric enables meaningful comparisons between reputation systems, and present a technique for generating hybrid reputation systems with variable incentive-compatibility and efficiency properties.

Acknowledgments

I thank David Parkes for being a great thesis mentor over the past year. Without his guidance and encouragement this senior thesis could not have happened. As a student, advisee, teaching fellow, and researcher, it has been a privilege to learn from him over the past three years. I thank Sven Seuken for lending me his invaluable technical expertise over the past year, and for reading countless drafts, often at odd hours. I thank Jean Yang for all the feedback and for motivating me to get up and write. I thank Thomas Carriero and Seth Flaxman for reading and editing drafts of this work. I thank my parents for pushing me to be all I can. Finally, I thank Kristina for all the support.

Contents

Chapter 1

Informative Reputation Systems

The world is an increasingly interconnected place; people sell goods online, swap media files, browse social connections, and search for information among the billions of web pages indexed by various search engines. When a user searches for a popular book on eBay or Amazon.com, he or she is often presented with dozens or hundreds of possible sellers to choose from. How can users tell the difference between legitimate businesses and outright frauds? For better or worse, the Internet is inherently an open system, making it difficult to assess the authenticity of a seller on an auction site or the authority of a web page. Reputation systems, which rank agents based on feedback from past interactions, play a crucial role in aggregating and sharing trust information online.

Under eBay's reputation system, for example, buyers are asked to rate the quality of sellers every time an interaction (a sale) occurs. If a seller doesn't ship the promised item, the buyer's negative feedback is recorded for others to see. Over time, buyers begin to avoid sellers with poor reputations and reward sellers with high ones. These two steps capture the essence of what a reputation system is about. Agents interact with each other and send ratings to a reputation system. Absolute reputation scores or relative reputation rankings are computed and exposed to aid the agents in making future decisions. The measure of a good reputation system is "economic efficiency", the extent to which the reputation system information results in the "best" decisions being made by users.

This problem would be significantly easier if the users were not rational agents who will try to cheat the system for their own gain. Malicious sellers may create fake accounts and leave positive feedback, abandon accounts and start over when their reputation dips too low, or leave negative feedback for competitors to drag them down. This issue of incentive compatibility, designing reputation

systems which discourage or prevent cheating, has driven a large amount of the prior work on reputation systems. These approaches have led to interesting impossibility results and characterization theorems for simple reputation systems.

An open challenge for reputation system researchers is the lack of good metrics for "economic efficiency." It is inherently difficult to formally capture the efficiency of a reputation system in a manner which is domain-independent; on a file-sharing network, for example, efficiency might be measured in the ratio of authentic to inauthentic files swapped, while on eBay efficiency might be measured in the number of users making a satisfying purchase. Different approaches to reputation systems research have been more or less ad-hoc, using simulations to estimate efficiency rather than searching for optimal reputation systems.

The goal of this thesis is to bridge the gap between these different approaches (incentive compatibility on one hand, economic efficiency on the other) by first defining a novel metric for reputation systems which approximates efficiency, and second applying this metric to the problem of finding optimal reputation systems. Our metric is based on the accuracy or *informativeness* of the reputation system; intuitively, this correlates with efficiency, because as more information is available to the user the better the decision he/she can make. We present a technique for generating hybrid reputation systems which change their incentive-compatibility and efficiency properties as we vary a weighting parameter. Finally, we show empirically that the informativeness metric enables meaningful comparisons between reputation systems.

1.1 Reputation Overview

Why is the reputation system problem hard? We are forced on a daily basis to judge the reliability of commercial transactions. If we were looking to buy a car from a used-car dealer, we would presumably weigh many different factors to determine the trustworthiness of the dealer. Our personal interaction with the dealer would count for a lot: when we visit, is the dealership clean and professional-looking? Are the cars kept in good condition? Is the lot located in a back alley or off a major street? We might examine the dealer's past transactions by calling past customers and asking about their experiences. We might ask close friends whether they have had experience with this particular dealer.

And we should weight these factors differently: our personal experience may count for more than our friends' experiences, while our friends' opinion may count for more than the opinion of a

stranger. And since such a large amount of money is involved in this purchase, the dealer may be tempted to "game" the reputation system: for instance, he might point us to his business partner as an example of a satisfied customer. Only after weighing these factors appropriately do we decide whether to purchase from this dealer or to move on to another. The study of reputation systems attempts to model this process formally. Given a set of agents and reported trust ratings, how can we rank the agents from most trustworthy to least?

Once we have a ranking, there are a number of questions we might want to ask. First, is the ranking that we get accurate (*i.e.*, are bad dealers exposed as untrustworthy)? If it is, we can use it to rank dealers before making our choices. Next, does it lead us to make the right decisions (*e.g.*, buy or not buy)? This captures the decision making at a higher level: we may not care about the relative rankings of bad dealers, so long as we know to avoid them. Finally, how easy is it for dealers to manipulate the rankings to their advantage? If financial gain is involved, people are sure to seek ways of cheating the system.

Over the course of this thesis, I will formalize the intuition behind these three questions into three metrics for reputation systems: *informativeness*, *economic efficiency*, and *incentive-compatibility*, respectively. While we generally care the most about economic efficiency (we want our reputation system to provide information which leads to good choices) it is a difficult concept to capture in practice. The informativeness metric we want to develop provides a good approximation to efficiency: intuitively, the more information we take into account when deriving our reputation scores, the better agents can make their decisions and the higher the resulting efficiency.

1.2 Motivation

Reputation systems have found applications in a variety of practical domains. As integral components of search engines, file-sharing networks, and shopping sites, reputation systems represent a highly active field of current research.

Web site ranking

The Internet is composed of billions of pages of hypertext, put up by corporations, organizations, and individuals. Because it is open and anonymous, anyone can post information on a web site. However, judging the authenticity of a source is a crucial part of what search engines like Google need to do to generate useful, relevant results. The PageRank algorithm [18], developed by Google's

founders, is one of the reputation systems examined later in this thesis: it can be viewed as a reputation system which models the web as a graph, where each web page is a node and directed edges represent hyperlinks between pages. When particular queries are searched through Google's search engine, the first results shown are those with the best PageRank scores, appropriately weighted by some measure of the relevancy of the page to the search term.

This explosion of information availability has also made us more dependent on search engines like Google for finding and organizing information. These search engines in turn drive the development of reputation systems like Google's PageRank algorithm, which rate the reliability of web sites by examining the hyperlink structure of the web (sites which are linked to more frequently should be thought of as more authoritative). For web site owners, the relative ranking or reputation of a web site can cause huge shifts in the amount of incoming traffic and advertising revenue. Yet the reputation systems underlying search engines are rarely transparent and available: Google relies heavily on secrecy to prevent web site owners from optimizing their sites for higher rankings. For businesses dependent on income from Google-driven traffic, this lack of transparency is unsettling to say the least. This has spurred work on incentive-compatible reputation systems which cannot be "manipulated." The rules for such systems can be published openly, addressing this need for transparency.

Online auction sites

Online auction sites like eBay and Amazon.com's Marketplace have enabled small businesses and individuals to reach thousands of niche markets. Millions of items have been listed and sold online. Yet the relative anonymity of the Internet creates opportunities for criminals to abuse the system and profit from fraud. To combat such behavior, eBay and Amazon implement sophisticated feedback and rating systems to aid their users in making smart buying decisions.

Peer-to-peer networks

Peer-to-peer networks have emerged as a lasting component of the Internet's infrastructure: studies estimate that in 2006 between 50% and 90% of all Internet traffic was P2P-related [1]. Software like Skype, BitTorrent, and Joost enable us to chat over VoIP, swap media files, and enjoy streaming video, while other P2P systems create ad-hoc wireless networks and manage distributed grid computation systems. Such networks are scalable and efficient. With no central server to connect to, there is no single bottleneck or point of failure that can bring the system down.

However, such networks are complicated to understand precisely because there is no central authority to mediate between different self-interested agents. How can we model interactions between multiple rational agents, and how can we incentivize them to behave in socially efficient ways? One solution is to introduce a reputation system. If we can identify users or nodes engaging in beneficial behaviors by assigning them higher reputations, and if those reputations confer some tangible benefit (*e.g.*, faster downloads for peers which share more files), we can get cooperative, collective behavior from self-interested agents.

Need for Security

Online shopping sites like eBay and Amazon process billions of dollars' worth of financial transactions. To help detect and prevent fraud, buyers and sellers have the opportunity to rate each other after every transaction. Ideally, honest dealers are rewarded with high ratings and higher profits, while shady dealers are avoided or removed after enough negative feedback. Such systems do provide incentives to play by the rules: a study of eBay's online auctions by Resnick *et al.* [20] revealed that high reputation sellers earned on average 7% more than sellers with no rating. In a separate, randomized, controlled field experiment by Resnick [21], a high-reputation seller earned 8.1% more on average using an established identity versus using new seller identities. On another level, while it may be inconvenient if a shady dealer on an auction site fails to ship an order, there is real danger whenever personal identity information is available online: if a scammer obtains a billing address or credit card number he can rack up thousands of dollars in fraudulent purchases. Reputation systems address a real need for security: by propagating trust information across the network, these systems prevent malicious agents from repeatedly scamming users.

Need for Transparency

Because of Google's popularity, its algorithms for ranking sites are often the largest drivers of traffic to small and mid-sized commercial web sites. Being ranked on the first (rather than the second) page of Google's search results for a particular query results in orders of magnitude more traffic, which in turn leads to more revenue from advertisements or sales. For businesses which depend heavily on such revenue, understanding Google's reputation systems results in real profits.

Thus, web site owners go to great lengths to ensure good rankings. This has spawned an entire industry centered around search engine optimization (SEO), the art of changing pages and content to generate good rankings for particular queries. Yet this is unsatisfactory: every time Google alters

its ranking algorithms, web site owners are forced to make tweaks and adjustments to maintain their ranking. Conversely, there is no way for site owners to know in advance what will lead to higher rankings. A greater level of openness or transparency with regards to Google's ranking algorithms is to be desired. To address this issue research on reputation systems has also been focused on incentive compatibility: *i.e.,* how to make systems that cannot be "manipulated." If a reputation system cannot be manipulated, the rules and algorithms it follows can be openly disseminated, eliminating the wasteful user opimization underlying current systems.

1.2.1 Related Work

There are two general approaches to the study of reputation systems which differ mainly on the emphasis that is put on theoretic incentive compatibility results versus practical efficiency results. Axiomatic approaches seek to understand and model simple reputation systems by proving strong theoretical results: examples include work by Altman, Cheng, and Chayes [4, 3, 2, 24, 8, 5]. Other, more practically-focused domain-dependent approaches involve focus on simulating and evaluating the efficiency of different reputation systems; see for example the PageRank and EigenTrust papers [18, 15].

Because work on reputation systems draws from many disparate disciplines, each with its own models and histories, comparing different reputation systems in a common framework is a problem which has not been addressed. Neither of these two approaches offers a common framework for evaluating reputation systems. The axiomatic approach, while offering extensive incentive-compatibility results, does not allow for quantitative comparisons between reputation systems, while the domain-dependent approach fails to formally define an acceptable efficiency metric.

1.3 Primary Contributions

Economic efficiency is the primary metric on which reputation systems need to be judged. This thesis introduces a new approach to analyzing reputation systems based on a notion of "informativeness". The core motivation behind this metric is the difficulty of developing good economic efficiency metrics: prior work either attempts to model utility-maximization problem formally, in which case it is intractable to solve, or it attempts to estimate efficiency through simulation, which does not scale well to different problem domains. The informativeness metrice we define is tractable and acts as a good proxy for economic efficiency: the more trust graph information the reputation

system captures, the easier it is for agents to make good decisions based on proper reputation data. Later, we run simulations which yield empirical evidence for our intuition that the informativeness metric correlates well with economic efficiency.

This thesis also examines an intriguing negative correlation between incentive compatibility and informativeness. We develop a technique for combining existing reputation systems into hybrid reputation systems, and characterize general incentive compatibility properties for these hybrid reputation systems. These include both positive and negative results on the preservation of incentive-compatibility as different reputation systems are combined. In order to characterize how incentive compatibility properties may be traded off for informativeness, we introduce natural relaxations of existing incentive-compatibility constructs, and demonstrate their use by proving theorems about two hybrid reputation systems with convenient properties. The hybrid technique allows us to search for the optimal reputation system for a given problem by adjusting a single weighting factor which blends the two reputation systems.

1.4 Paper Outline

The remainder of this thesis is as follows: Chapter 2 lays out background for a general model of reputation systems as computing a function on a "trust" graph, and describes several variations and special cases of the basic model (symmetric vs. asymmetric, binary vs. real-valued trust). We describe the basic game-theoretic framework behind the study of reputation systems, then detail four different classes of reputation system, each based on different graph algorithms: eigenvector-based algorithms like PageRank and EigenTrust, hitting-time-based algorithms, maxflow-based algorithms, and shortest-path-based algorithms. The chapter concludes with a discussion of different reputation-system metrics. Chapter 3 introduces our hybrid reputation system construct, which allows us to create new reputation systems with specific tradeoffs between incentive compatibility and informativeness. We prove general incentive-compatibility properties about this technique, as well as specific results for two convenient hybrid reputation systems. By mixing together reputation systems in this novel way, it is possible to trade off informativeness and efficiency in return for incentive-compatibility. I conclude in Chapter 4 with possible applications of this work, outlining what must be done to connect theory with practice.

Chapter 2

Modeling Reputation

Because of the variety of possible applications, past work on reputation systems has developed from a wide range of disciplines ranging from economic mechanism design to graph theory to computer systems research. In this chapter I provide a brief survey of past work, with the goal of unifying a number of different approaches under a single framework. By focusing on common themes between past approaches, I hope to motivate my perspective on this problem. In the process, we develop a formal framework for talking about reputation systems. This framework will be sufficiently general to capture all the reputation systems I will examine, including maxflow [8], Eigentrust[15], PageRank [18], and shortest paths [4].

2.1 Economic Efficiency

The following list illustrates the steps an end-user of a reputation system might go through:

1. Users form opinions on the trustworthiness of other users
2. Users make reports to the reputation system.
3. The reputation system computes the reputation of each user.
4. Users interact with other users (buying goods, exchanging files), taking into account reputation information.
5. Repeat steps 1-4

Here users, or agents, can refer to humans buying and selling goods through an auction site or to automated software bots transferring files over a peer-to-peer network.

When stated this way, the goal of a reputation system is clear: it must provide information on user reputation that leads agents to make economically efficient, utility-maximizing decisions.

Unfortunately, it is unclear what utility an agent gets from a particular output of the reputation system. While there have been interesting empirical studies which attempt to quantify the value of having higher reputation (*e.g.,* Resnick *et al.* 's study of eBay's reputation system [20]), it is difficult to extend such work to a general agent utility function in a context independent setting. The differences between having a high seller rating on eBay and having a high reputation on a peer-to-peer file-sharing network appear too great to be captured by a single all-encompassing utility model.

2.1.1 Economic Efficiency in Prior Work

There have been two main approaches to the problem of modeling utility in the literature. The first approach, which we will term the axiomatic approach, attempts to sidestep the issue by dealing directly with the final ranking or reputation scores output by the reputation system. Implicit in this formulation is the assumption that utility is directly related the reputation scores / ranking.

Axiomatic Approach and the Theory of Social Choice

Work which takes the axiomatic approach to reputation systems has focused on identifying and analyzing basic axioms about reputation systems (hence the name of the approach). This approach has led to both impossiblity results (*i.e.,* a set of reasonable axioms cannot be satisfied by any reputation system) and representation theorems (*i.e.,* this set of axioms uniquely characterizes a particular reputation system).

The axiomatic approach is closely connected to the classical theory of social choice (see Arrow [6]) — the reputation system problem is modeled as a special case of a social choice problem where the set of agents and the set of alternatives coincide, and the agents have two levels of preferences over the alternatives (*i.e.,* each agent has a set of other agents which it trusts, and another set which it doesn't). Under the classical social choice formulation of this problem, a social choice function is incentive compatible if agents cannot improve the rankings of their preferred alternatives by misreporting their true preferences.

The axiomatic approach's focus on the final ranking may be justifiable in the reputation system setting because an agent's place in the final ranking roughly corresponds to higher overall utility; web pages which appear higher on Google's search results tend to get more traffic and more revenue. However, there are several serious objections to this approach for analyzing reputation systems.

First, it is difficult to formally define the sybil attack, a common manipulation which involves an agent creating fake "sybil" agents which participate in the voting.

Second, this approach is not scalable; axioms for one reputation system must be carefully re-proven if we wish to apply them elsewhere. And each reputation system itself requires carefully developed axioms characterizing its properties.

A domain-dependent utility framework

The second approach to reputation systems, which we will call the domain-dependent approach, strives to model agent utility as precisely as possible. Agents take the output of the reputation system into account when choosing their actions; the output of the reputation system only indirectly influences the final utility through the choice of agent actions. Under this approach, after reports are made to the reputation system, we assume the reputation system returns a set of reputation scores $f_i(v_j)$ representing the trust agent v_i places in agent v_j. Agents must then act upon the reputation information in some way, *e.g.,* by interacting with another agent with high reputation scores or rankings. The utility of an agent is determined entirely by this final interaction.

This motivates the following model for agent utility: each agent v_i has a type θ_i, and a set A_i of possible actions (not to be confused with possible misreport actions of reputation information). Each agent has a decision function which, given another agent v_j, uses the reputation $f_i(G, v_j)$ of v_j to determine an action to take. Finally, each agent has a utility function which takes an action and the type of v_j and determines the utility gained.

Definition 2.1.1. Given a reputation system M, a set of agents $V = \{v_1, ..., v_n\}$, a set of types $\theta = \{\theta_1, ..., \theta_n\}$ s.t. $\forall i, \theta_i \in \theta$, and a trust graph $G = (V, E, w)$, define for each agent $v_i \in V$ an action space $A_i = \{a_1, ..., a_m\}$, and define an action function $a_i : V \times \mathbb{R} \to A_i$, which given an agent to interact with v_j and a reputation score $f_i(G, v_j)$, determines the action $a \in A_i$ that will be taken. Finally, define for each agent a utility function $u_i : \theta \times A_i \to \mathbb{R}$.

Together these definitions define the rules under which agents can interact under a reputation system, but we still cannot determine the utility of v_i without a model of how v_i interacts with other

agents. This motivates the definition of an interaction profile $\mathbb{I}$, which is essentially a list of the other agents which agent v_i interacts with over its entire time in the system.

Definition 2.1.2. Define an interaction profile $\mathbb{I} = (g_1, g_2, ..., g_p) s.t. g_i \in V$. The g_i are not necessarily distinct. p is the number of different interaction opportunities the given agent receives.

Putting all these pieces together,

Definition 2.1.3. Given a set of agents $(v_1, ..., v_n)$, reputation scores for these agents $(f_1, ..., f_n)$, action spaces $A^i = \{a_1, ..., a_m\}$, action functions $a_i : \mathbb{R} \to A^i$, and utility functions $u_i : A^i \to \mathbb{R}$, define the utility of agent i under interaction profile $\mathbb{I}$ as $\sum_{j=1}^{p} u_i(\theta_i, a_i(g_j, f_i(g_j)))$.

For example, in a peer-to-peer filesharing setting, one might use the EigenTrust algorithm as the reputation system; the action space for each agent is $\{Share, NoShare\}$ and $a_i(f_i(v)) = Share$ if $f_i(v) > c$, *i.e.*, if the current agent being considered for an interaction has reputation $> c$ for some constant c. Finally, $u_i(a, \theta) = 1$ if $a = Share$ and $u_i(a, \theta) = 0$ if $a = NoShare$ regardless of θ.

Critiques of the domain-dependent approach

After going through the technical details of the above utility framework, it is important to keep in mind that we were only trying to formalize how one would calculate the utility of an agent in the system. Simply formalizing how utility is calculated under the domain-dependent approach is a daunting challenge; formulating and solving a utility maximization problem is likely to be much harder.

For this reason, work done under the domain-dependent approach tends to involve simulations or actual deployments rather than theoretical results. Each paper develops some measure of social welfare which is highly dependent on the problem context. For example, analyses of reputation systems for peer-to-peer networks often involve simulations of actual networks, using the ratio of authentic to inauthentic files exchanged as a measurement of overall utility. Though this approach has yielded more complex reputation systems, incentive-compatibility issues tend to take a back seat to practical issues. Worse, simulations of reputation systems on peer-to-peer networks do not provide evidence that the same reputation systems can be applied to other domains.

Both the axiomatic and the domain-dependent approaches have their benefits — see Table 2.1 for a comparison of the approaches. The axiomatic approach gives us strong incentive-compatibility theorems, while the domain-dependent approach allows for the comparison of more complex reputation systems in a quantitative way. We will draw from both approaches throughout this thesis

Approach	Focus	Methods	Incentive Compatibility	What is missing
Axiomatic	simple theoretic models	proving characterization theorems	provable properties	standard metrics for comparing reputation systems
Domain-dependent	complex rep. systems	testing through simulation	ad-hoc results	results are domain-specific and difficult to extend
Overall				Ways of comparing the efficiency of existing reputation systems

Table 2.1: This table summarizes the differences between the two main approaches to reputation system research. The "What is missing" column especially highlights the contribution this thesis work aims to make

in order to prove incentive-compatibility properties and to compare existing reputation systems in quantitative ways.

However, both approaches fall short of providing a good metric for the economic efficiency properties of reputation systems. This is troubling because economic efficiency is the motivating reason for using reputation systems. As an extreme example, consider the trivial reputation system which assigns the same constant reputation to each agent irregardless of agent reports. This system is perfectly incentive-compatible because agent reports cannot influence the final reputation score, but at the same time it is adds nothing of value to the system it is deployed in.

2.1.2 Motivation for a novel metric

The domain-dependent approach has taught us that explicitly modeling agent utility is intractable in the general case and unscalable when applied to particular situations. Put another way, solving the general utility model for the utility-maximizing strategy is not feasible, and investigating reputation systems individually through simulation does not enable useful comparisons of economic efficiency.

This gap in usuable metrics provides motivation for our proposed "informativeness" metric.

Intuitively, informativeness tells us how much information our reputation system uses or how accurately our reputation system determines the true reputation of agents. The more information our reputation system takes into account, the more accurate the predicted reputation scores, which results in better decisions and (ideally) higher utility for the user. Thus, it is reasonable to expect informativeness to be a useful proxy for economic efficiency. Since maximizing social welfare is not possible in general, informativeness provides a good way of approximating economic efficiency. We discuss the informativeness metric more thoroughly in the next chapter.

2.2 Game Theory and Incentive Compatibility

Thus far we have glossed over the strategic interactions between agents in a reputation system, talking instead in generalities about incentive-compatibility and utility. We now consider the reputation system problem formally from a game-theoretic perspective.

Agents in a reputation system have many opportunities for strategic interaction, beginning with the trust ratings they report to the reputation system. An agent u making reports about other agents to the reputation system must consider the effect his reports will have on the final reputation scores, and then consider the effect of different reputation scores on the interaction decisions made by users. The effect of making a given report depends heavily on the reports that other agents will make as well as our model for agent actions. How do we determine how self-interested agents are likely to behave?

The field of game theory deals exactly with this problem of predicting the behavior of multiple self-interested agents behavior. In particular, game theory provides powerful *solution concepts* which simplify our analysis. In order to talk meaningfully about this, we first establish some preliminaries, following the model set out by Parkes in [19].

Definition 2.2.1. (Strategy) A *strategy* is a complete contingent plan or decision rule that defines the actions an agent will take.

In our setting, during the stage when trust information is reported to the reputation system agents can play all sorts of unexpected *strategies*, *e.g.*, misreporting their true trust ratings.

Definition 2.2.2. (Game) A *game* defines a set of strategies S_i for each agent and a utility function $u_i(s_1,...,s_n,\theta_i)$ that defines the utility of agent i given the agent's type θ_i and the strategy profile $(s_1,...,s_n)$ being played by all agents.

When modeling reputation systems as games, the strategies of the agents are the reports about other agents that the agent makes to the reputation system. Once these reports are made however, how do we compute the utilities of the different agents in the system?

For our incentive compatibility analysis, we will follow the axiomatic model and assume that utility is directly related to the final reputation scores output by the reputation system. The utility an agent gets increases as its reputation score and/or relative ranking increases.

Once we have this notion of a reputation system as a *game*, we can discuss different *solution concepts* for predicting the outcome of the game: *i.e.,* the expected behavior of participating agents.

Definition 2.2.3. (Nash Equilibrium) Let $s = (s_1, ..., s_n)$ denote the strategy profile for the strategies of all agents, and let $s_{-i} = (s_1, ..., s_{i-1}, s_{i+1}, ..., s_n)$ denote the strategy of every agent except agent i. A strategy profile s is a Nash equilibrium if

$$\forall i, u_i(s_i(\theta_i), s_{-i}(\theta_{-i}), \theta_i) \geq u_i(s_i'(\theta_i), s_{-i}(\theta_{-i}), \theta_i) \forall s_i' \neq s_i$$

The above definition says that in a Nash equilibrium, every agent must be maximizing its expected utility. In games with many agents, this solution concept is unsatisfying because it requires strong assumptions on agents' information and beliefs; namely, that all agents possess perfect information and assume perfect rationality on the part of other agents in computing the expected utility of different strategies.

A more useful concept is that of a *dominant strategy equilibrium*.

Definition 2.2.4. (Dominant Strategy Equilibrium) Strategy s_i is a dominant strategy if it weakly maximizes the agent's expected utility for all possible strategies of other agents,

$$u_i(s_i, s_{-i}, \theta_i) \geq u_i(s_i', s_{-i}, \theta_i) \forall s_i' \neq s_i$$

A dominant strategy for an agent maximizes expected utility no matter what strategies the other agents play. This concept is superior to the standard Nash equilibrium concept because it does not require the agents have any information about each other, and does not require agents believe that other agents will play rationally. If a dominant strategy exists, no matter what the best move for an agent is to play the dominant strategy.

For the remainder of this thesis, we focus solely on characterizing the dominant strategies of the reputation system games being played. We are especially interested in reputation systems where the dominant strategy is to truthfully report your private information (your opinions of other agents in the system) to the reputation system.

Definition 2.2.5. (Dominant Strategy Incentive Compatibility) A game is dominant-strategy incentive compatible or *strategyproof* if truthfully reporting types is a dominant strategy equilibrium.

2.3 Formal Reputation Systems

2.3.1 The Trust Graph

The first step in the process of building a reputation system is modeling the process of agent reports. Given a set of agents V, it is assumed that each agent $v_i \in V$ begins with ratings for some subset of the agents $V_i \subseteq V$. V_i is a subset of the set of all agents because agent v_i hasn't necessarily interacted with every other agent in V. Each agent begins by reporting its ratings to the reputation system.

Definition 2.3.1. (Agent Reports) Given a set of agents $V = \{v_1, ..., v_n\}$, for each i let V_i denote the agents that v_i has trust information about. The agents in V make reports $R = \{(V_1, t_1), ..., (V_n, t_n)\}$ where $t_i : V_i \rightarrow \mathbb{D} \subseteq \mathbb{R}^+$, so $t_i(v)$ represents the trust agent v_i assigns to v.

A natural encoding for this data is a trust graph, in which the nodes represent agents and the edges represent ratings of these agents. For example, in the search engine setting, nodes might represent web sites, and edges might represent hypertext links between these web sites. On an auction site, nodes represent buyers and sellers, and edges represent a rating for each transaction that has occurred (if an edge does not exist between a pair of agents, it indicates that no interaction has occurred between the two).

The set $\mathbb{D}$ is the set from which agents draw their trust ratings of other agents; its exact form depends on the context (*e.g.,* for simple binary trust, $\mathbb{D} = \{0, 1\}$. For a shopping site, it may range from $\mathbb{D} = \{1, ..., 5\}$.).

Next, define the notion of a trust graph constructed on the basis of agent reports following Altman and Cheng [4, 8]:

Definition 2.3.2. (Trust Graph) A trust graph $G = (V, E, w)$ is a set of vertices V and directed edges $(u, v) \in E, u, v \in V$. Each edge $(u, v) \in E$ has an associated weight $w(u, v) \in \mathbb{D} \subseteq \mathbb{R}^+$. (*i.e.,* vertices are individual agents, edges indicate interactions or trust between agents, and edge weights indicate levels of trust).

Some points to note: because of the natural mapping between agents and vertices in the trust graph, I will often refer to them interchangably, using the uniform notation v_i.

Edges in the trust graph can be directed or undirected. Undirected graphs are are appropriate in situations where interactions have a sort of symmetry. For example, consider a social network setting. It is reasonable to expect friendship relationships between users to be reflexive: if I am friends with you, you should be friends with me. On the other hand, in the web page reputation setting, a hyperlink takes visitors from one page to another - the relationship is not symmetric, and is best captured by directed edges. This is the model adopted by the EigenTrust [15] and PageRank [18] papers. In general, the asymmetric case is more common — consider buyers and sellers on an auction site, or downloaders and uploaders on a file-sharing network — and the directed edge model is richer than the undirected model (it is possible to model undirected edges by setting $w(u,v) = w(v,u)$)). The remainder of this thesis uses directed edges exclusively.

The trust values reported by each agent (which eventually become the weights on the edges of the trust graph) can be drawn from different subsets $\mathbb{D}$ depending on the problem domain. Binary trust, in which $\mathbb{D} = \{0,1\}$, is a commonly used model in which an agent either trusts or doesn't trust another agent. This is the simplest model to analyze, and is found predominantly in papers which demonstrate rigorous theoretical results: Cheng's sybilproofness paper [8] and Altman's axiomatic approach [2] are two examples.

However, when users are asked to rate transactions online it can be useful to have a wider range of options than trust/no trust. Shopping sites like eBay and Amazon, for example, allow agents to rate others on an integer scale: $\mathbb{D} = \{0,1,...,K\}$. In other contexts, it is useful to set $\mathbb{D} = [0,1]$, allowing the weights to be interpreted as probabilities (when properly normalized). This approach provides the most expressiveness, though agents may not directly choose each weight - in the Eigentrust reputation system [15], agent reports are normalized s.t. $\sum_{u \in V_i} t_i(u) = 1$, *i.e.*, the agent's reported trust in other agents sums to 1. This forms the actual weights of edges on the trust graph.

Also, the trust graph makes a distinction between an edge (u,v) of weight 0 and the absence of an edge (u,v); it is not a complete graph. The first situation might arise if agent u has had both positive and negative interactions with agent v such that the net trust agent u places in agent v is 0, while the second situation would arise if no interaction has taken place between agents u and v.

Using these definitions, the trust graph is naturally defined on the basis of agent reports:

Definition 2.3.3. (Constructing Trust Graphs) Given a set of agents V and agent reports R, construct a trust graph $G = (V,E,w)$ as follows (note that the agents map exactly to vertices in the graph): for each vertex v_i, given report (V_i,t_i), create a directed edge $(v_i,u) \in E$ for each $u \in V_i$ and

define $w(v_i, u) = t_i(u)$.

Note that the above mapping can be used to generate a graph structure from any given set of agent reports. This gives a way of constructing a model for any trust system.

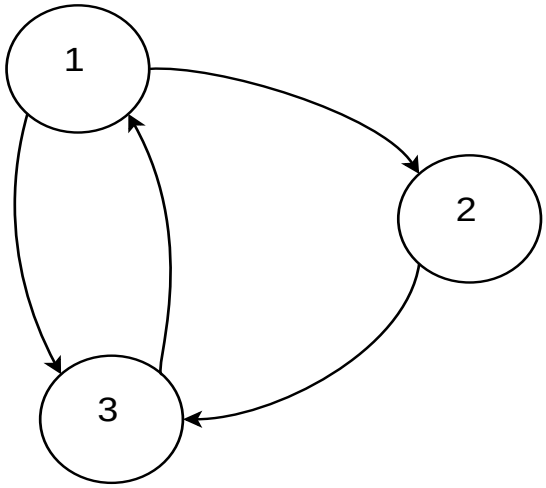


Figure 2.1: Example trust graph induced by a particular pattern of agent interactions. Here agent 1 has had positive experiences buying from agents 2 and 3, agent 2 has bought from agent 3, and agent 3 has bought from agent 1.

For example, consider the trust graph in Figure 2.1. Agent 1 has directed edges to agents 2 and 3, while agent 3 has a directed edge to agent 1, and agent 2 has a directed edge back to agent 3. In the online auction setting (eBay), this situation could arise if agent 1 has positive transactions with agents 2 and 3, agent 2 with agent 3, and agent 3 with agent 1. Intuitively, agent 1 should place more trust in agent 3 than in agent 2, because in addition to having a direct interaction with agent 3 agent 1 also has an indirect relationship through agent 2.

2.3.2 Reputation System

The informal definition of a reputation system given in the first chapter described the problem of a reputation system as deciding how to rank agents based on their own input. Using the trust graph to model agent input, following the definition given by Altman [4], it is possible to formally define a reputation system as a function on a trust graph.

Definition 2.3.4. (Reputation system) A reputation system M is a mapping from a trust graph $G = (V, E, w)$ and its vertices V to $\mathbb{R}^n$ *i.e.*, a function $f : G \times V \to \mathbb{R}^n$. Each $f(G, v_j) = (r_1, ..., r_n)$,

where $f_i(G, v_j) = r_i$ (the ith component of $f(G, v_j)$) can be thought of as the reputation of agent v_j from the perspective of agent v_i. This induces an ordering $\prec_i^G$ over the agents, where $f_i(G,u) < f_i(G,w) \Rightarrow u \prec_i^G w$ indicates that agent w has higher reputation than agent u from the perspective of agent v_i (and should appear higher in the ranking).

Under Definition 2.3.4 we require that f be a completely defined function over V; that is, given a trust graph G and vertices v_i, v_j $f_i(G, v_j)$ always has a value; reputation systems provide predictions for every agent from every other agent's perspective. Even if v_i and v_j are not connected in G, the reputation $f_i(G, v_j)$ must exist. Also note that a higher relative value of $f_i(G,u)$ is "better" in the sense that it implies more trust from agent i in u. An agent u is ranked higher than another agent v from v_i's perspective if $f_i(G,u) \geq f_i(G,v)$.

There remains a fair amount of ambiguity in Definition 2.3.4; this allows for considerable flexibility and variation in applying it to various problem domains.

Symmetric vs. asymmetric reputation

Reputation systems can be divided into symmetric and asymmetric (or alternatively global and local) categories. A *symmetric* reputation system computes a single reputation score for every node in the network. It is useful to think of this as a global system, in which a single reputation score is maintained globally for each agent. On the other hand, *asymmetric* reputation systems keep local reputation information for each agent; each agent has its own reputation score for every other agent.

The symmetric/asymmetric distinction is not to be confused with the undirected/directed edges distinction — the undirected/directed nature of the edges reflect inherent (a)symmetries in the problem formulation, while the differences between symmetric/asymmetric reputation system are more arbitrary. Symmetric and asymmetric reputation systems have different computational efficiency and incentive compatibility characteristics, so neither strictly dominates the other. Thus these two distinctions are fundamentally orthogonal: it is possible to have a symmetric reputation system with directed edges (see PageRank [18]) as well as asymmetric reputation systems with undirected edges (*e.g.*, shortest-path reputation).

More formally, symmetric systems are a special case of the general reputation system formulation.

Definition 2.3.5. A reputation system is symmetric if $\forall i, j$ we have $f_i(G, v_k) = f_j(G, v_k)$.

Symmetric algorithms generally benefit from being more efficient to compute and simpler to analyze. A good way to think about symmetric systems is that there exists a single global reputation score for each agent; thus there are $O(n)$ trust scores to compute as opposed to $O(n^2)$ scores under an asymmetric system. However, later it will be shown that symmetric algorithms necessarily lack a key incentive-compatibility property.

Asymmetric reputation systems may be justified if we agree that each node should trust itself more than any other node in the network, or because each node is in a different position in the trust network. There have been numerous papers analyzing approval voting and max-flow-based algorithms for asymmetric reputation systems [2, 4, 8].

2.4 Manipulations

In this section, it is assumed that an agent's utility is determined directly by its final ranking; thus, agents choose manipulations in order to maximize their relative ranking. We consider several types of manipulation when considering incentive compatibility, but each of the reputation systems are susceptible to some or all of the attacks we describe.

Sybil Manipulation

A sybil manipulation [11] involves an agent creating and inserting a number of fake agents (under the original agent's control) into the network.

Definition 2.4.1. (Sybil manipulation) Given a trust graph $G = (V, E, w)$, a sybil manipulation strategy for node $v \in V$ is a tuple $\sigma = (S, E_S, w_S)$ where $S = \{s_1, ..., s_m\}$ is a set of sybil agents, E_S is a set of edges $E_S = \{(u, w) : u \in S \cup \{v\}, w \in V \cup S\}$ and $w_S : E_S \to \mathbb{D}$ are the weights on the edges in E_S. This results in a modified trust graph $G \downarrow \sigma = G' = (V \cup S, E \cup E_S, w')$, where $w'(e) = w(e)$ for $e \in E$, and $w'(e') = w_S(e')$ for $e' \in E_S$.

Under Definition 2.4.1, under a sybil manipulation an agent can create sybil agents with arbitrary outlinks to any other agent in the trust system. Because it is cheap to create accounts automatically on an auction site or set up hundreds of fake web pages, this type of manipulation is easy to conduct and must be defended against. If an agent can improve its reputation using such an attack, the reputation system is vulnerable to sybil attack.

Under this formulation, an agent can add as many sybil nodes as it wants, and can create whatever graph structure (assuming directed edges and edge weights) it chooses between itself and its sybils in order to create the modified graph G'. However, it cannot alter links from other nodes to point to its sybils, and it cannot redirect incoming links to point to its sybils instead. Consider the web-page example: if I create some sybil pages, I can create hyperlinks from them back to my main site, but I cannot force other sites to link to my sybils.

Before it is possible to compare which reputation systems are vulnerable to sybil manipulation, it is necessary to define the concept of sybilproofness.

Definition 2.4.2. (Rank-sybilproof) A reputation system is rank-sybilproof if given a trust graph $G = (V,E,w)$, for any sybil strategy σ s.t. $G \downarrow \sigma = G' = (V \cup S, E \cup E_S, w')$, for all $u,v \in V$ and $i \in \{1,...,n\}$,

$$f_i(G,v) > f_i(G,u) \;\Rightarrow\; f_i(G',v) > f_i(G',u)$$

.

A reputation system is considered rank-sybilproof if no agent u can increase its reputation and surpass that of another node v that originally had higher reputation than u by employing a sybil manipulation. It is preferable that a reputation system be value- or rank-sybilproof. However, such an attack is very difficult to defend against. Though symmetric reputation systems are easier to characterize and study, Cheng and Friedman have shown the strong negative result that no symmetric reputation system is value-sybilproof [8].

Theorem 1. ***(Cheng and Friedman)*** *There is no symmetric rank-sybilproof nontrivial reputation function. Here nontrivial refers to any reputation function that does not return a constant reputation for all nodes, i.e.,* $\forall i,v \; f_i(G,v) = C$ *for some* $C \in \mathbb{R}$.

Proof. (Sketch) If the reputation system is symmetric, and a malicious node m is not the highest ranked agent, m can create a duplicate copy of the trust graph structure using sybil nodes. The two graphs are connected by node m. Because the sybil copy graph is symmetric, there is some sybil node s that has a higher reputation than node m; since m controls s it has successfully increased its reputation. □

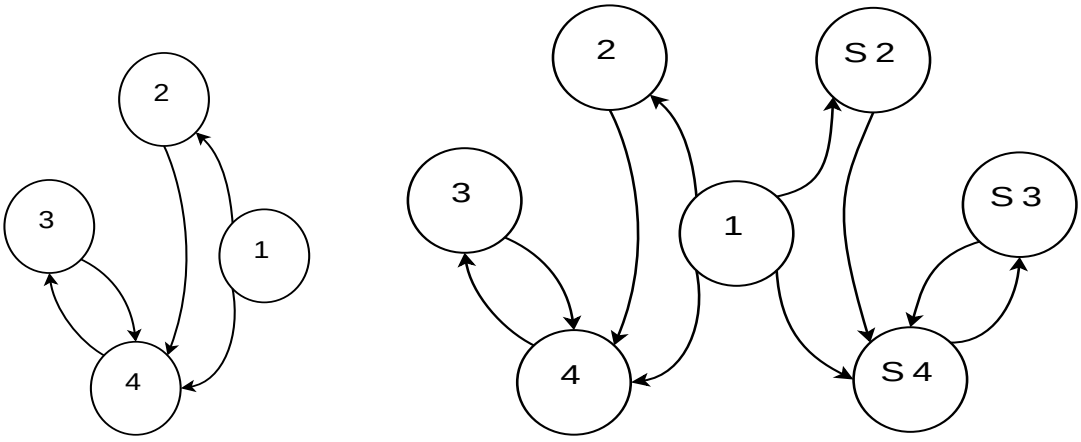


Figure 2.2: (a) Example graph G with four nodes and four edges. Perhaps agent 4 has the highest reputation (b) On the right, after the sybil manipulation by node 1, we have a duplicate copy of the graph starting at the manipulating node 1. By symmetry one of these nodes (S4) must have higher reputation than node 1.

Misreports

Under a misreport manipulation, an agent u may simply lie about its interaction with another agent v. Reporting a bad interaction about agent v can lower v's reputation, and if u was originally ranked below v this can improve u's relative ranking.

Definition 2.4.3. (Misreport Strategy) Given a trust graph $G = (V,E,w)$, define the set $E_{-v} = \{(u,x) : (u,x) \in E, u \neq v\}$ (*i.e.*, the set of all edges in G that do not start at v). A misreport strategy for $v \in V$ is a tuple $\sigma = (V,E_v,w_v)$ where $E_v = \{(v,u) : u \in V\}$ and $w_v : E_v \to \mathbb{D}$. Applying the strategy σ to G results in a modified trust graph $G \downarrow \sigma = G' = (V,E_{-v} \cup E_v, w')$ where $w'(e) = w(e)$ for all $e \in E_{-v}$, and $w'(e') = w_v(e')$ for all $e' \in E_v$

Note that the edges in the set E_v do not necessarily have to exist in the original graph; it is allowable for a node to make up an edge. Definition 2.4.3 mandates that all edges and edge weights not originating from node v must remain the same in the modified graph. Thus, an agent v can misreport the weights on any of its outlinks (*i.e.*, edges originating from v), but it cannot affect the reports other agents make about it.

Whitewashing

Another class of manipulation is the whitewashing attack, in which an agent removes itself from the reputation system and adds a new node under its control (*e.g.*, a new account). Because such an attack is cheap to conduct when dealing with a peer-to-peer network or other computerized system, a viable reputation system needs some sort of initiation cost for new members. Such an effect has been analyzed by Resnick and colleagues in their empirical analysis of eBay's reputation system [14, 20]; it has also been examined theoretically in work by Feldman and colleagues [13]. However, modeling this attack requres a model of the dynamics of the system (when do agents enter and exit) we will not deal with it now.

2.4.1 Strategyproofness

The sybil and misreport manipulations are often studied together; Altman has shown that the asymmetric shortest-path algorithm, in which the reputation of an agent u from the perspective of an agent $v \in V$ is given by the length of the shortest path between u, v, is in some sense resistant to these manipulations [3]. Sheldon *et al.* demonstrate a manipulation-resistant hitting-time based approach to reputations based on PageRank [23]. I will discuss both these algorithms in detail later.

For the remainder of this thesis, I consider any combination of sybil and misreport strategies as possible manipulations of the system. To handle combining these strategies, define the composition of two manipulations as follows:

Definition 2.4.4. Given manipulations σ_1 and σ_2, for any graph G, define the composite manipulation $\sigma = \sigma_1 \circ \sigma_2$ s.t. $G \downarrow \sigma = G'$ iff $\exists G''$ s.t. $G \downarrow \sigma_2 = G''$ and $G'' \downarrow \sigma_1 = G'$.

Since each type of manipulation takes a graph as input and returns a modified graph, composing these two operations is well-defined.

A system that cannot be manipulated is called strategyproof, because an agent cannot earn a higher utility by applying a manipulation. This is a dominant-strategy equilibrium formulation: telling the truth must always result in higher utility using Assumption 1. For now, we follow the axiomatic approach and assume the utility of an agent to be directly related to the agent's relative ranking. This implies that a reputation is strategyproof if no agent can increase its rank by applying a manipulation.

Definition 2.4.5. (Rank-strategyproof) A reputation system is rank-strategyproof if given a trust graph $G = (V, E, w)$, for every $w \in V$ and for every manipulation strategy σ for node w s.t. $G \downarrow \sigma =$

G', for all $v_i, u \in V$, $u \neq w$,

$$f_i(G,w) < f_i(G,u) \Rightarrow f_i(G',w) < f_i(G',u)$$

This definition states that a reputation system is rank-strategyproof if we cannot increase our relative ranking by applying a sybil and/or misreport manipultion. If $w \prec_i^G u$, then on trust graph G' we must still have $w \prec_i^{G'} u$. Rank-strategyproofness turns out to be very difficult to guarantee. A different strategyproofness concept is based on the absolute repuation score of each agent.

Definition 2.4.6. (Value-strategyproof) A reputation system is value-strategyproof if given a trust graph $G = (V,E,w)$, for all $u \in V$ and for all manipulation strategies σ for u s.t. $G \downarrow \sigma = G'$, for all $v_i \in V$, $f_i(G,u) \geq f_i(G',u)$.

Value-strategyproofness guarantees that an agent cannot increase its own reputation in the eyes of another agent. Why do we introduce this concept? It seems that the relative ranking of an agent should determine its final utility, especially in settings like web search. However, in other domains it is often useful in practice to set absolute cutoffs for reputation scores which confer some benefit to the agent. For example, a peer-to-peer file sharing network might be set up so that all agents with reputation higher than some cutoff c get to download twice as fast as other agents.

A reputation system may be value-strategyproof but not rank-strategyproof because under a value-strategyproof system it may still be possible for an agent to reduce the reputation of a higher reputation agent; this may lower the rank of the higher-reputation agent to below that of the manipulating agent.

However, the reverse is also true: a rank-strategyproof system may not be value-strategyproof. It may be possible for an agent to increase its reputation score (*e.g.,* $f_i(G,u)$ for agent u from agent v_i's perspective) under a rank-strategyproof system, so long as the reputations of higher-ranked agents increase as well. Thus rank-strategyproofness does not strictly dominate value-strategyproofness (and vice versa). In most reputation systems studied in the literature, however, rank-strategyproof reputation systems are also value-strategyproof, so in general rank weakly dominates as an IC concept.

2.4.2 Relaxations

Unfortunately, both value- and rank-strategyproofness are difficult to achieve. It is useful to introduce two relaxations of the value- and rank-strategyproofness concepts that attempt to ensure that

the final ranking / reputation scores remain "close" to the true ranking / scores (as opposed to ensuring that scores remain the same). This is done by introducing a parameter ε, which determines exactly how "close" to the true ranking a reputation system must be under any manipulation strategy.

Definition 2.4.7. (ε-value-strategyproof) A reputation system is ε-value-strategyproof for $\varepsilon \geq 0$ if given a trust graph $G = (V, E, w)$, for all $u \in V$ and for all manipulation strategies σ for u giving $G' = G \downarrow \sigma$, for all $v_i \in V$, $f_i(G, u) + \varepsilon \geq f_i(G', u)$.

This states that under an ε-value-strategyproof system an agent u cannot increase its reputation score by more than ε (as viewed from any other agent v_i under any manipulation strategy σ for a trust graph G). An additive factor ε (rather than a multiplicative factor) is appropriate for this relaxation for the same reason value-strategyproofness is a useful concept: in some domains reputation scores above a certain fixed threshold might confer some benefits to the agent.

Definition 2.4.8. (ε-rank-strategyproof) A reputation system is ε-rank-strategyproof for $\varepsilon \geq 0$ if given a trust graph $G = (V, E, w)$, for all $u \in V$ and for all manipulation strategies σ for u s.t. $G' = G \downarrow \sigma$, for all $v_i \in V$, $w \in V$,

$$f_i(G, u) + \varepsilon \leq f_i(G, v) \Rightarrow f_i(G', u) \leq f_i(G', v)$$

This states that an agent u whose reputation under trust graph G is not within ε of another agent v (*i.e.*, it is ranked below agent v under G), then it cannot become ranked higher than agent v after applying a manipulation under an ε-rank-strategyproof system.

These relaxations have some precedent in prior work. Altman and Tennenholtz [3] quantify the incentive compatibility of different reputation systems by weakening the rank-strategyproofness incentive compatibility concept. Because they define strategyproofness solely in terms of the final ranking (and not the absolute reputation score) the relaxation they develop, k-worst-case-rank-strategyproofness, is stated in terms of rank rather than value. Under a reputation system which is k-worst-case rank-strategyproof, an agent can increase its ranking by at most k places by executing a misreport attack. Since this thesis takes into account both sybil and misreport manipulations, the concept of rank-strategyproofness uses the absolute reputation score rather than the relative ranking.

2.5 Strategyproofness of existing reputation systems

Using the strategyproofness concepts developed in the previous section, this section presents several different reputation systems, along with a discussion of their incentive-compatibility properties. The

goal is to exhibit an interesting link between incentive compatibility and the amount of information in the trust graph used by the reputation system. This is in essence what I want to quantify with my informativeness metric.

2.5.1 Eigenvector-Based Methods

Eigenvector-based reputation systems can be thought of as random walks on weighted, directed trust graphs $G = (V, E, w)$, where the edge weights $w(v_i, v_j)$ on edges $e_{ij} = (v_i, v_j)$ leaving any vertex v_i have been normalized so that $\sum_{j=1}^{n} w(v_i, v_j) = 1$ - *i.e.*, so that we can treat each $w(v_i, v_j)$ as the probability of moving from vertex i to vertex j.

This suggests the following random-walk based algorithm:

Definition 2.5.1. (Random-walk-based algorithm) Given a trust graph $G = (V, E, w)$, begin from a random node $v_i \in V$. At each step, with probability $w(v_i, v_j)$, jump to a random neighbor v_j of v_i.

Such a process can be modeled as a Markov process with transition matrix $T = [e_{ij}]$, where $0 < i \leq n$, $0 < j \leq n$.

The reputation of a given node $v \in V$ is given by the weight on v in the stationary distribution π of this process (the distribution satisfying $\pi = T\pi$). This is given by the principal eigenvector of the matrix T, and the reputation of a node v is defined as $f(G, v) = (\pi_v, \pi_v, ..., \pi_v)$, where π_v is the probability of finding the random walk at node v in the stationary distribution. Under the random-walk interpretation of this process, the reputation of a node is the probability the random walk is at node v as the number of timesteps $t \to \infty$.

The motivation behind this kind of algorithm is simple yet elegant: the reputation of a node or agent is quickly approximated by the time a web surfer or file-sharer would spend interacting with the given agent assuming it randomly transitioned (*e.g.*, clicked links or shared files) with different neighbors of the current agent.

Two examples of this class of reputation system are PageRank [18] and EigenTrust [15].

PageRank

PageRank was originally developed by Page and Brin [18] to analyze the reputation of hypertext documents; the nodes of the trust graph represent web pages, while the edges represent hyperlinks between pages. Because each hyperlink is symmetric, the weight on any given edge from a vertex u is given by $w(u, v) = \frac{1}{\text{out-degree(u)}}$, where out-degree(u) is the number of edges (hyperlinks) leaving

u. By definition, this results in $\sum_{j=1}^{n} w(v_i, v_j) = \text{out-degree}(v_i)\frac{1}{\text{out-degree}(v_i)} = 1$, so the weight vectors are properly normalized. Under the random-walk interpretation of this algorithm, the random walk has an equal probability of transitioning to any of the neighbors of a given vertex.

PageRank also introduces a dampening factor d: with probability d, at each step the random walk pauses at its current state. If we let $\vec{x_k}$ be a vector $(p_k^1, ..., p_k^n)$ where p_k^i represents the probability of the random walk being at node i after k time steps, we can represent the transition function for this walk as follows:

$$\vec{x_k} = (1-d)T\vec{x_{k-1}} + d\vec{x_{k-1}}$$

Finally, if the random walk reaches a node with no outgoing links, PageRank randomly jumps to another node in the trust graph with uniform probability.

EigenTrust

The EigenTrust algorithm as originally described by Kamvar *et al.* in [15] is very similar to PageRank in that it involves computing the stationary distribution of a random walk over a trust graph. However, it uses more complicated edge weights than PageRank: in EigenTrust, the weight $w(v_i, v_j)$ is the difference between the number of positive and negative interactions between individual agents in a peer-to-peer network. These weights are then normalized so that $\sum_{j=1}^{n} w(v_i, v_j) = 1$. This is in some sense more appropriate for this setting because it captures more of the available trust information: there can be multiple download/upload interactions, not just the existence or absence of a hyperlink.

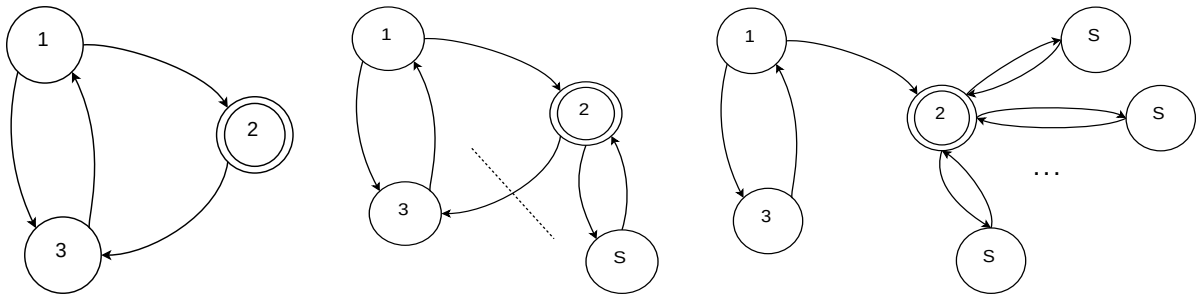
EigenTrust does not use a dampening factor in the random walk, but it does use a random jump factor β. The difference with PageRank is that EigenTrust jumps back to a random pre-trusted node rather than to a random node in the network. More formally, let $p = (p_1, ..., p_n)$, where p_i is the prior probability of randomly jumping to vertex v_i. The global trust vector x is updated by the following process:

$$\vec{x_k} = (1-\beta)Tx_{k-1} + \beta p$$

where β is a suitably chosen probability between 0 and 1.

The motivation for using pre-trusted nodes rather than random nodes is to improve the incentive-compatibility properties of the algorithm. The authors argue that PageRank is more susceptible to sybil attack because by creating a large number of sybil nodes a malicious attacker can control where

the random walk jumps to. However, assuming the existence of pre-trusted nodes is unsatisfying; it is unclear where such pre-trusted nodes originate, and how the system guarantees that no malicious nodes are included.



(a) (b) (c)

Figure 2.3: (a) Given this simple unweighted trust graph, PageRank returns [0.39,0.21,0.40] (b) After adding a single sybil and cutting his outlink, agent 2 changes the PageRank vector to [0.11,0.41,0.08,0.39] (c) If agent 2 is allowed to add an arbitrary number of sybils, its PageRank goes to 0.5 while the PageRank of every other node goes to 0.

Bianchini *et al.* [7] have shown that the optimal sybil manipulation for a node v under the PageRank algorithm is to create sybil nodes in a star formation.

Theorem 2. ***(Bianchini** et al. **)** Given any trust graph G, vertices v_i, v_j, and for any manipulation* σ *resulting in* $G' = G \downarrow \sigma$,

$$f_i(G', v_j) \leq 0.5$$

Proof. **(sketch)** For the PageRank reputation system, the optimal manipulation for a given node u is to create N sybils, where N is as large as possible. Each sybil links to node u, and u links to each of its sybils. Furthermore, u cuts all outlinks to non-sybil nodes.

Any random walk that reaches either node u or any of its sybils cannot escape, and hits node u on every other step. Thus, in the limit as $N \to \infty$, the stationary probability of u approaches 0.5.

If a node u has a PageRank greater than 0.5, $E[X_t = u] > 0.5$ which implies that $P(X_{t+1} = u | X_t = u) > 0$, so there must exist a self loop from u back to itself; however, this is explicitly disallowed by construction.

Therefore, 0.5 is the optimal PageRank manipulation. □

This manipulation extends naturally to analyses of other eigenvector-based algorithms like

EigenTrust. Figure 2.3 illustrates the sybil attack on the PageRank algorithm: even a single sybil can greatly improve the reputation of a malicious node.

Thus, PageRank and EigenTrust do not satisfy rank-strategyproofness nor value-strategyproofness; by applying this manipulation an agent can raise both the absolute reputation and the relative rank of a given node. In fact, work by Cheng and Friedman [9] has bounded the increase in ranking that is possible under PageRank given a fixed number of sybils.

2.5.2 Hitting Time Reputation

In 2007, Sheldon and Hopcroft [23] proposed a manipulation-resistant reputation system based on the hitting time of a random walk over a trust graph. This system, described more formally below, builds on the PageRank reputation system by creating a ranking which is close to the PageRank ranking, but is value-strategyproof against sybil attacks. Because it privileges a set of pre-trusted nodes, this algorithm is asymmetric in nature.

Defining the hitting time algorithm first requires a definition of the hitting time of a node v under a kind of random walk on a trust graph $G = (V, E, w)$. Define a starting distribution q; the starting node is chosen randomly from this distribution. At each step, with probability α, the random walk jumps to a node chosen randomly from the starting distribution q. With probability $1-\alpha$, the random walk randomly follows an outgoing edge of the current node. Because the parameter α can vary, I refer to these random walks as α-random walks. More formally,

Definition 2.5.2. let $(X_t)_{t\geq 0}$ be the sequence of nodes visited by this walk. Then $P(X_0 = v) = q(v)$ and

$$P(X_t = v | X_{t-1} = u) = \begin{cases} \alpha q(v) + \frac{1-\alpha}{out-degree(u)} & \text{if } (u,v) \in E \\ \alpha q(v) & \text{if } (u,v) \notin E \end{cases}$$

The definition of an α-random walk is analogous to the model of the random surfer in the PageRank algorithm. However, where the PageRank algorithm uses the stationary probability of the random walk as the reputation of each node, this algorithm focuses on the hitting time of each node.

Definition 2.5.3. The *hitting time* of a node v is $H(v) = min\{t : X_t = v\}$, $E[H(v)]$ is the expected number of steps before a given α-random walk first arrives at node v.

Definition 2.5.4. The *jump time* of an α-random walk is given by J a geometric random variable with parameter α; J should be interpreted as the first time the α-random walk jumps to a node from the starting distribution instead of randomly following an edge from the current node.

Under the hitting-time based algorithm, the reputation of a node $u \in V$ from the perspective of node v is the probability that a random walk on the trust graph hits u without randomly jumping.

Definition 2.5.5. Given a trust graph G, the reputation of node u from v_i's perspective is $f_i(G,u) = Pr(H(u) < J)$

Because J is simply a geometric random variable, the reputation of a node actually correlates closely with the hitting time of the node. Though this reputation system computes a single global trust value for each agent, because it uses a predefined trust distribution q it can still be value-strategyproof without contradicting Theorem 1.

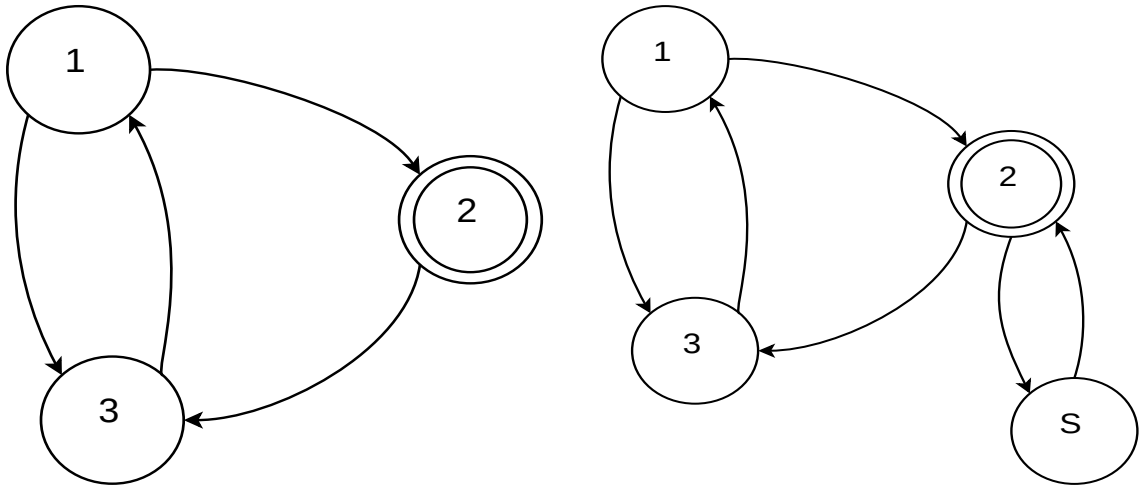


Figure 2.4: (a) The reputations under the hitting time algorithm for this small graph approximate PageRank: [0.39,0.21,0.40] (b) Adding sybils does not shorten the hitting time for agent 2 because the only way to get to the sybil node is to go through agent 2

The star-shaped PageRank manipulation is not effective under the hitting time algorithm. Because sybils and other manipulations are only allowed to create outlinks from malicious nodes, it is impossible to shorten the path to node v. Similarly, agent v cannot decrease its hitting time by adding additional out-links (either to sybils or to other nodes) because before such outlinks are considered the random walk must already have arrived at v.

Note that an agent can still improve its ranking in the system by reducing the reputation of other agents. It can do this by removing outlinks to higher-reputation nodes, thus lengthening the hitting time by forcing random walks to take alternate paths. Thus this does not satisfy rank-strategyproofness. However, Shelden and Hopcroft are able to bound the effectiveness of such manipulations:

Theorem 3. ***(Sheldon and Hopcroft)*** *Under the hitting time algorithm, node $u \in V$ cannot surpass a node $w \in V$ that is at least twice as reputable: i.e., given any manipulation σ and any trust graph G, let $G \downarrow \sigma = G'$. Then for all v_i,*

$$2 f_i(G,u) \leq f_i(G,w) \Rightarrow f_i(G',u) \leq f_i(G',w)$$

This result is interesting because it ties together value- and rank-strategyproofness. Under the hitting time reputation system, a node which is twice as reputable as another node (on an absolute, value-based scale) cannot be surpassed in rank by the lower-valued node. We will use this result later to show desirable incentive compatibility properties of a hybrid hitting-time / shortest-path algorithm.

However, the design of this reputation system is troubling because of the initial pretrusted distribution q. It is unclear how one would find and designate such pre-trusted nodes, and how one would verify that no pre-trusted node was malicious.

Asymmetric hitting-time based algorithms, which generally do not depend on the existence of such an initial distribution, have also been defined and studied in the literature [5]. Like in the symmetric hitting-time mechanism, the reputation of agent u is equal to the probability that certain α-random walks hit node u before jumping. However, under asymmetric hitting-time algorithms, only random walks which start from node v_i are considered when computing $f_i(G,u) = Pr(H_i(u) < J)$, where $H_i(u) = min(t : X_t = u, X_0 = v_i)$.

This asymmetric hitting-time algorithm can be shown to be value-strategyproof but not rank strategyproof; the argument proceeds analogously to the proof for the symmetric hitting-time algorithm.

2.5.3 Maxflow-based Algorithms

Another family of reputation system which is value-strategyproof is the asymmetric maxflow-based family studied by Cheng and Friedman [8] and Altman *et al.* [4] from an axiomatic incentive-compatibility perspective. To understand this reputation system it is necessary to define what is meant by the flow through the graph.

Definition 2.5.6. Define a flow $flow(v_i, v_j)$ between a sink v_i and a source v_j in a trust graph $G = (V, E, w)$ to be a mapping $F : E \to \mathbb{R}$ satisfying the following properties:

1. for all $e \in E$, $F(e) \leq w(e)$
2. for any $v \in V$, let $I(v) = \{(u,v) \in E : u \in V\}$ be the set of incoming edges and let $O(v) = \{(v,u) \in E : u \in V\}$ be the set of outgoing edges. Then $\sum_{e \in I(v)} F(e) = \sum_{e \in O(v)} F(e)$.

The first condition is a capacity constraint; it ensures that the flow across any edge does not exceed the capacity of the edge. The second condition is a flow constraint; the flow entering a vertex must be equal to the flow leaving a vertex. In order to define what the maximum flow through a graph is, define the value of a flow as follows:

Definition 2.5.7. Let $I(v) = \{(u,v) \in E : u \in V\}$. The value of a flow $flow(v_i, v_j)$ is defined as $\sum_{e \in I(v_i)} F(e)$

This is exactly the flow leaving the source vertex, and would be equivalent to the flow entering the sink vertex.

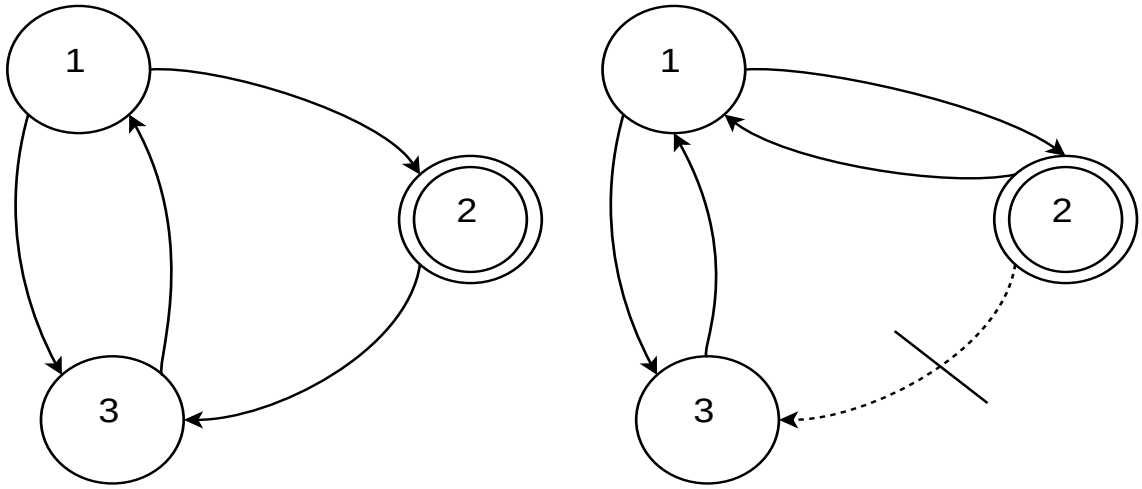
Definition 2.5.8. The maximum flow $MF(v_i, v_j)$ is the flow of maximum value between v_i and v_j.

For more information about maximum flow algorithms, see a reference book such as [10].

The maxflow reputation system sets agent v_j's reputation as viewed from agent $v_i \in V$ to be the value of the maximum flow from v_i to v_j.

Definition 2.5.9. Given a trust graph G and vertices v_i, v_j, the maxflow reputation system sets $f_i(G, v_j) = MF(v_i, v_j)$.

The intuition behind this system is that each trust relationship from agent i to agent j is indicative of the maximum amount of trust or utility that agent i would lend to agent j; thus, when considering any path in the trust graph between two agents, the smallest weight (smallest capacity) edge



(a) (b)

Figure 2.5: (a) Under the max-flow system, the reputation of agent 2 from agent 1's perspective is 1, while the reputation of agent 3 from agent 1's perspective is 2 because of the direct path from 1 to 3 and the indirect path from 1 to 2 to 3. (b) Maxflow is not rank-sybilproof because by removing the link to agent 3, agent 2 causes the reputation of agent 3 from agent 1's perspective to drop to 1, making the agents tied in rank.

determines how much trust the path contributes. Mobius *et al.* [17] demonstrate that the maxflow reputation system falls out naturally when modeling a borrowing game on a social network.

This algorithm is value-strategyproof. To see this, consider a trust graph $G = (V, E, w)$, and arbitrary agents $v_i, v_j \in V$. The reputation of v_j from v_i's perspective is $f_i(G, v_j) = MF(v_i, v_j)$. Playing a sybil strategy cannot increase the maximum flow between v_i, v_j (and thus cannot increase the reputation of agent v_j) because no links can be added to the sybil nodes from nodes already in the trust graph, and so no additional flow can be sent through the sybils. Also, v_j misreporting its outlinks cannot increase the maximum flow, because any flow crossing an edge leaving v_j must already have entered v_j.

However, the maxflow algorithm is not rank-strategyproof: it is possible for an agent *i* to remove an outlink to an agent *j* with higher reputation. This potentially lowers the reputation of agent *j*, increasing the relative ranking of agent *i*. A simple example of this manipulation can be found in Figure 2.5.

2.5.4 Shortest-Path Algorithms

Finally, the asymmetric shortest-path algorithm, described in detail by Altman [4], deserves mention as both the simplest and the most manipulation-resistant reputation system. Given a trust graph G, let $SP(v_i, v_j)$ denote the length of the shortest path between agents v_i and v_j. In the unweighted edges setting, this is simply the number of hops between v_i and v_j on graph G.

Definition 2.5.10. For a given trust graph G, the asymmetric shortest-path algorithm sets agent v_j's reputation as viewed from agent $v_i \in V$ to be $f_i(G, v_j) = \frac{1}{SP(v_i, v_j)}$.

Intuitively, we should trust an agent which is 5 steps away less than an agent we have a direct trust relationship with.

The difference between our definition and the definition of the shortest-path reputation system due to Altman is that Altman never defines the reputation score for an agent, instead working directly with the relative rankings of the agents. Under his definition of the shortest-path system,

$$SP(v_i, u) < SP(v_i, v) \Leftrightarrow u \prec_i^G v$$

It is simple to verify that the function we have chosen for $f_i(G, v_j)$ satisfies the above property.

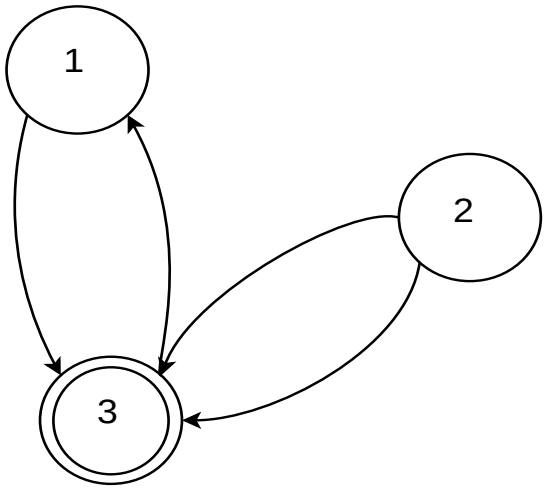


Figure 2.6: The reputations of agents 2 and 3 from agent 1's perspective are 1/2 and 1, respectively. Agent 2 can do nothing to increase his reputation by adding outlinks, because by the time the shortest path algorithm considers outlinks from agent 2 it must already have arrived at agent 2.

This algorithm is rank-strategyproof in addition to being value-strategyproof. It is value-strategyproof because an agent j cannot change the shortest path from agent i to j by either creating sybils or misreporting outlinks; the path must already have arrived at agent j before it can include outlinks or

Reputation System	Directed?	$\mathbb{D}$	Symmetric?	Incentive Compatibility
EigenTrust	directed	[0,1]	symmetric	no
PageRank	directed	{0,1}	symmetric	no
Hitting Time	directed	{0,1}	both	value-strategyproof
Max-flow	both	{0,...,C}	asymmetric	value-strategyproof
Shortest Path	both	{0,1}	asymmetric	value and rank strategyproof

Table 2.2: This table summarizes the properties of the different reputation systems we have discussed in this chapter. Directed? refers to the use of directed vs. undirected edges; $\mathbb{D}$ is the set from which edge weights are drawn. As we move from top to bottom, we tend to find asymmetric reputation systems with better incentive-compatibility properties.

edges to sybils.

By misreporting edges, an agent j can potentially decrease the reputation of other agents. However, if the shortest path from agent i to agent k goes through agent j, then the reputation of agent k must be strictly less than the reputation of agent j (the length of the shortest path is at least 1 greater). Thus, agent j cannot decrease the reputation of any agents which are higher in rank, and so cannot improve its reputation score. See Figure 2.6 for a simple example.

2.5.5 Summary

Table 2.5.5 summarizes the setup and approaches used for analyzing each of the reputation systems discussed thus far, as well as the incentive compatibility properties. As we move from top to bottom, we tend to find asymmetric reputation systems with good incentive compatibility properties. However, we also move from algorithms which potentially use every edge in the trust graph to algorithms which use very few edges (shortest-path). This trend gives motivates our definition of informativeness: how much information does the reputation system make use of?

2.6 Informativeness

We wish to find a metric which tells us how accurately our reputation system predicts true types θ_i. This measure is useful because the more information our reputation system takes into account, the better the predicted reputation scores, which results in better decisions and higher utilities for

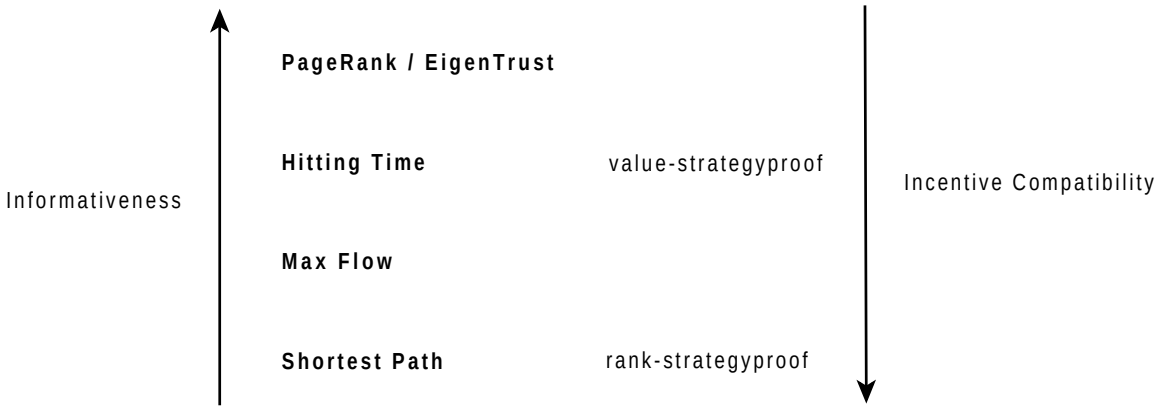


Figure 2.7: This summarizes the tradeoffs between informativeness and incentive compatibility. Incentive compatibility properties get better as we go from top to bottom: the shortest-path algorithm has the best incentive-compatibility properties. On the other hand, as we go from bottom to top the reputation systems take into account more information about the structure of the graph.

the users. Thus, our new informativeness metric may act as proxy for approximating economic efficiency, which is still our overriding standard when it comes to evaluating reputation systems.

The intuition for this metric comes from the observation that the shortest-path based reputation system ignores most of the information contained in the trust graph. Two agents that are a distance 2 away from a given node v have the same reputation, even if one agent is connected to all other agents that are a distance 1 away from v, while the other agent is connected to just 1 other agent a distance 1 away from v. As we move from the shortest-path system to the maxflow system and then the eigenvector-based systems, we take into account more and more of the trust graph information. This is desirable because trust information is being propagated more quickly across the network, leading to fewer instances of abuse. But as more of the trust graph is taken into account, incentive compatibility properties are lost. As shown in Figure 2.7, the shortest-path system is in some sense the "most" incentive-compatible (rank-strategyproof). Maxflow algorithms are value-strategyproof, while hitting time algorithms are merely "resistant" to manipulation.

Unfortunately, like economic efficiency, informativeness is a difficult metric to formalize; we choose to deal with it in an empirical fashion through simulation of a real problem domain. However, our metric is still general enough to allow multiple existing reputation systems to be compared on the same scale.

Definition 2.6.1. (Informativeness) Given a trust graph G, a set of agents $\{v_1,...,v_n\}$ with types

$\{\theta_1, ..., \theta_n\}$, define the informativeness of a reputation system M as $I = \sum_{i=1}^{n} \sum_{j=1}^{n} (f_i(G, v_j) - \theta_j)^2$

Since each agent in the system is given a type which determines the probability of malicious behavior, the measure of a reputation system's informativeness is the squared error between an agent's real type and its predicted reputation score.

Chapter 3

Hybrid Reputation Systems

This chapter presents the novel theoretical contributions of this work. The first idea is a way of combining two different reputation systems into a hybrid reputation system. I then demonstrate several incentive-compatibility properties of the resulting systems.

Since there are reputation systems like PageRank which are informative and economically efficient but possess poor incentive-compatibility properties, and reputation systems like maxflow with poor informativeness and efficiency properties but good incentive-compatibility, a natural thing to do is to take a convex combination of an agent's reputation score under different reputation systems.

3.1 Theoretical Properties of Hybrid Reputation Systems

Definition 3.1.1. The α-hybrid of two reputation systems M_1 and M_2 is defined as a reputation system $M_\alpha(M_1, M_2)$: given a trust graph $G = (V, E, w)$, let $f_i^1(G, v_j)$ denote the reputation of node $v_j \in V$ from $v_i \in V$'s perspective under reputation system M_1, and let $f_i^2(G, v_j)$ be similarly defined for M_2. The reputation of v_j from v_i's perspective under $M_\alpha(M_1, M_2)$ is given by

$$f_i^\alpha(G, w) = \alpha f_i^1(G, v_j) + (1 - \alpha) f_i^2(G, v_j)$$

There are a few issues related to normalization when combining absolute reputation scores in this way. Most of our reputation systems (*e.g.,* PageRank) output reputation scores in the range $[0, 1]$, but maxflow could output flows in the range $[0, 10]$, $[0, 1000]$, etc. Combining maxflow with another reputation system in a naive way clearly biases the resulting hybrid.

However, the raw reputation scores are meaningless; a maximum flow of 50 on one trust graph cannot be compared to a maximum flow of 1000 on another graph. Only the relative ranking

amongst agents is significant. Thus, we are free to normalize the output of the maxflow algorithm to fall in $[0,1]$. If we let M denote the maximum capacity of any edge, and $|E|$ denote the number of edges, the maximum flow between any pair of vertices is bounded above by $M|E|$. Whenever we use the maxflow reputation system in a hybrid system, we can normalize the output of maxflow to lie between $[0,1]$ by divide the raw reputation scores output by $M|E|$.

3.2 Theoretical bounds

There are a few simple bounds on the incentive compatibility properties of certain hybrid reputation systems that can be rigorously demonstrated. These bounds can be more sophisticated the more that is known about the reputation systems M_1 and M_2.

3.2.1 General properties

It is reasonable to expect certain properties to follow directly if M_1 and M_2 possess the same strategyproofness properties ; *i.e.,* , if M_1 and M_2 are both value-strategyproof we expect M_α to be value-strategyproof as well.

Lemma 4. *If M_1 and M_2 are value-strategyproof on all trust graphs G, then M_α is value-strategyproof on all graphs G.*

Proof. This follows from the value strategyproofness of M_1, M_2. For given nodes v_i, v_j, under any manipulation G', neither the contributions from the first nor the second component of $f_i^\alpha(G',v) = \alpha f_i^1(G,v_j) + (1-\alpha) f_i^2(G,v_j)$ can increase the reputation score of v. □

However, the analogous lemma does not hold for rank-strategyproofness. The proof (by counterexample) provides intuition for an additional condition that is needed.

Lemma 5. *If M_1 and M_2 are rank-strategyproof on all trust graphs G, then M_α is not necessarily rank-strategyproof on all graphs G.*

Proof. By counterexample.

Assume a trust graph G with two agents, 1 and 2. There is a link from agent 1 to agent 2.

Reputation system M_1 assigns a reputation of 1 to agent 2 and a reputation of 0 to agent 1 (and all other agents). This is trivially rank-strategyproof.

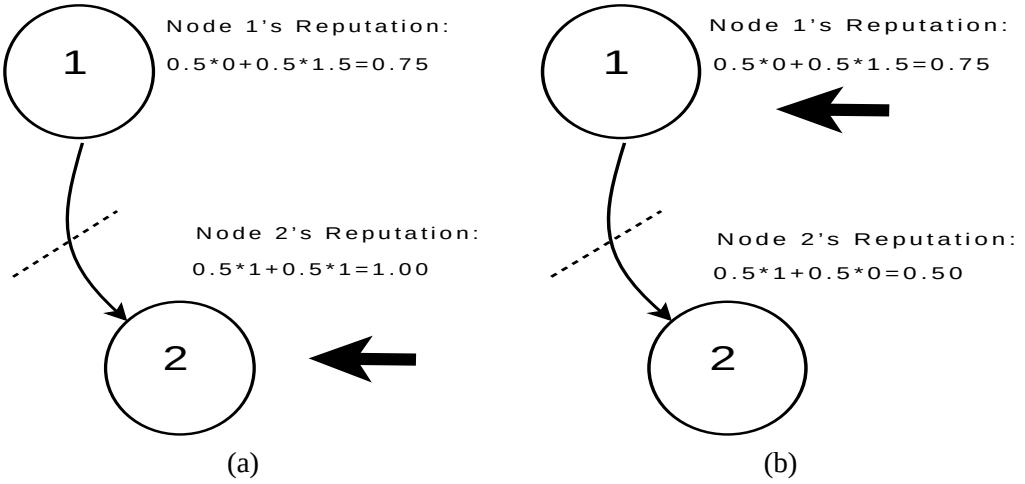


Figure 3.1: counterexample for Lemma (5). (a) On the left, we have the normal reputation values under M_1, M_2. Note that agent 2 has a repuatation lower than that of agent 1 under M_2. (b) On the right, agent 1 has cut the link, and so agent 2's reputation score contribution from M_2 drops to 0.

Reputation system M_2 assigns a reputation of 1.5 to agent 1, and assigns reputation 1 to agent 2 if an edge exists from agent 1 to agent 2 and reputation 0 otherwise. This is still rank-strategyproof because agent 1 has no agents ranked higher than it, and agent 2 cannot affect the final ranking.

Now, for $\alpha = 0.5$, agent 1 has reputation 0.75 while agent 2 has reputation 1. If agent 1 removes the link to agent 2, then agent 2's reputation is lowered to 0.5, and agent 1 becomes ranked higher than agent 2. See Figure 3.1 □

This counterexample is possible because there is a failure of monotonicity between M_1 and M_2: because the relative rankings of agent 1 and 2 are different under M_1 and M_2, lowering the reputation of a lower-ranked agent can cause rankings to flip under the composite M_α. So, if it is known that the relative ranking under M_1 is the same as the relative ranking under M_2, then the theorem about rank-strategyproofness theorem does hold.

Lemma 6. *If M_1 and M_2 are rank-strategyproof on all trust graphs G, and for all v_i, v_j, v_k $f_i^1(G, v_j) < f_i^1(G, v_k) \Leftrightarrow f_i^2(G, v_j) < f_i^2(G, v_k)$, then M_α is rank-strategyproof on all graphs G.*

Proof. Given any trust graph G and any agent v_i, consider two agents u, w. WLOG assume that under M_α $u \prec_i w$.

If we had that $f_i^1(G, u) > f_i^1(G, w)$, by the monotonicity condition $f_i^2(G, u) > f_i^2(G, w)$. Taking

convex combinations of these, we have

$$\alpha f_i^1(G,u) + (1-\alpha) f_i^2(G,u) > \alpha f_i^1(G,w) + (1-\alpha) f_i^2(G,w)$$

and then $u \succ_i u$, a contradiction.

Therefore, both $f_i^1(G,u) < f_i^1(G,w) \Leftrightarrow u \prec_i^1 w$ and $f_i^2(G,u) < f_i^2(G,w) \Leftrightarrow u \prec_i^2 w$. Since both M_1 and M_2 are rank-strategyproof, under any manipulation σ s.t. $G' = G \downarrow \sigma$ we must still have $f_i^1(G',u) < f_i^1(G',w)$ and $f_i^2(G',u) < f_i^2(G',w)$, so under the hybrid system

$$\alpha f_i^1(G,u) + (1-\alpha) f_i^2(G,u) < \alpha f_i^1(G,w) + (1-\alpha) f_i^2(G,w)$$

and we have $u \prec_i w$ as desired. □

Finally, if M_1 is rank-strategyproof while M_2 is not, it should not be surprising that M_α is not necessarily rank-strategyproof.

Lemma 7. *If M_1 is rank-strategyproof but M_2 is not rank-strategyproof on all trust graphs $G = (V,E)$, then M_α is not necessarily rank-strategyproof.*

Proof. This follows from the fact that M_2 is not rank-strategyproof: let M_1 be the trivial reputation system that assigns a score of 0 to all agents. The reputations of two agents i, j are now entirely determined by M_2, and since M_2 is not rank-strategyproof M_α cannot be strategyproof. □

A similar lemma holds for value-strategyproofness:

Lemma 8. *If M_1 is value-strategyproof but M_2 is not value-strategyproof on all trust graphs $G = (V,E)$, then M_α is not necessarily value-strategyproof.*

Proof. Similar to the above lemma. □

Finally, the equivalent lemmas exist for the ε-rank and ε-value-strategyproof concepts from Chapter 2.

Lemma 9. *If M_1 is ε_1-value-strategyproof and M_2 is ε_2-value-strategyproof on all trust graphs $G = (V,E)$, then if $\varepsilon = \varepsilon_1 + \varepsilon_2$ M_α is ε-value-strategyproof.*

Proof. This follows from the ε_i-value-strategyproofness of M_1, M_2. For given nodes v_i, v_j, under any manipulation G', we have

$$\begin{aligned} f_i^1(G, v_j) + \varepsilon_1 &\leq f_i^1(G', v_j) \\ f_i^2(G, v_j) + \varepsilon_2 &\leq f_i^2(G', v_j) \\ \Rightarrow \alpha f_i^1(G, v_j) + \alpha\varepsilon_1 + (1-\alpha) f_i^2(G, v_j) + (1-\alpha)\varepsilon_2 &\leq \alpha f_i^1(G', v_j) + (1-\alpha) f_i^2(G', v_j) \\ f_i^\alpha(G, v_j) + \alpha\varepsilon_1 + (1-\alpha)\varepsilon_2 &\leq f_i^\alpha(G', v_j) \end{aligned}$$

Since α is at least 0 and at most 1, the constant term is bounded by $\varepsilon_1 + \varepsilon_2$, so this is ε-value-strategyproof. □

Lemma 10. *If M_1 is ε_1-rank-strategyproof and M_2 is ε_2-rank-strategyproof on all trust graphs $G = (V, E)$, then M_α is not necessarily ε-rank-strategyproof if $\varepsilon = \varepsilon_1 + \varepsilon_2$.*

Proof. Set $\varepsilon_1, \varepsilon_2$=0 and apply Lemma 5 □

Lemma 11. *If M_1 is ε_1-rank-strategyproof and M_2 is ε_2-rank-strategyproof on all trust graphs $G = (V, E)$, if $\varepsilon = \varepsilon_1 + \varepsilon_2$, and the following monotonicity condition holds*

$$\forall v_i, v_j, v_k, f_i^1(G, v_j) < f_i^1(G, v_k) \Leftrightarrow f_i^2(G, v_j) < f_i^2(G, v_k)$$

then M_α is ε-rank-strategyproof

Proof. This proof proceeds similarly to Lemma 9 □

The last few lemmas are interesting because they allow us to chain hybrid reputation systems together; our hybrid algorithms generally end up with a relaxed strategyproofness formulation, so this lets us combine two ε-value strategyproof hybrids.

We can now pick any two existing reputation systems (provided they are normalized to output reputation scores in the range $[0, 1]$) and combine them using the α-hybrid technique outlined in Definition 3.1.1. There are however two reputation system hybrids that illustrate interesting incentive-compatibility properties and deserve special analysis: first, the PageRank/normalized maxflow hybrid reputation system is ε-value-strategyproof; second, the hitting-time/shortest-path hybrid reputation system is ε-rank-strategyproof.

3.2.2 Theoretical Properties of the PageRank/Maxflow Hybrid Reputation System

Let M_{PageRank} be the PageRank reputation system, and let M_{maxflow} be the maxflow reputation system. Let $M_\alpha(M_{\text{PageRank}}, M_{\text{maxflow}})$ be the α-hybrid of $M_{\text{PageRank}}, M_{\text{maxflow}}$. M_{PageRank} is neither rank- nor value-strategyproof, while $M_{\text{max-flow}}$ is value-strategyproof but not rank-strategyproof.

By immediate application of Lemma 4 above, M_α is not necessarily value-strategyproof. However, using specific information about PageRank and max-flow, it can be shown that the relaxed form of value-strategyproofness applies to $M_\alpha(M_{\text{PageRank}}, M_{\text{maxflow}})$.

Theorem 12. *$M_\alpha(M_{PageRank}, M_{maxflow})$ is 0.5α-value-strategyproof on all trust graphs $G = (V, E, w)$.*

Proof. Since M_{maxflow} is value-strategyproof, an agent u cannot improve its reputation score from agent v_i's perspective in the contribution from M_{maxflow} under any manipulation σ. If we let $G' = G \downarrow \sigma$,

$$(1-\alpha) f_i^{\text{maxflow}}(G', u) \leq (1-\alpha) f_i^{\text{maxflow}}(G, u)$$

.

By Theorem 2, the optimal manipulation for M_{PageRank} cannot increase the reputation score of any agent u above 0.5 (this involves creating an infinite number of sybils linking back to the manipulating node). For any such node u, the increase in reputation contributed by M_{PageRank} to the final reputation under manipulation σ yielding $G' = G \downarrow \sigma$ is

$$\alpha f_i^{\text{PageRank}}(G', u) - \alpha f_i^{\text{PageRank}}(G, u) \leq \alpha(0.5 - 0) = 0.5\alpha$$

Rearranging and summing these equations yields

$$\alpha f_i^{\text{PageRank}}(G, u) + (1-\alpha) f_i^{\text{maxflow}}(G, u) + 0.5\alpha \geq \alpha f_i^{\text{PageRank}}(G', u) + (1-\alpha) f_i^{\text{maxflow}}(G', u)$$

$$f_i(G, u) + 0.5\alpha \geq f_i(G', u)$$

which is the definition of 0.5α-value-strategyproof □

This is the type of relationship that was expected: by quantifying the incentive compatibility tradeoff as a function of α, it is possible to gradually improve the incentive-compatibility properties of this reputation system by decreasing α. If it can be shown that informativeness increases as α increases, there may exist an optimal value of α which trades off the "right amount" of incentive compatibility for informativeness.

3.3 Theoretical Properties of the Hitting-Time/Shortest-Path Hybrid Reputation System

The previous hybrid reputation system demonstrated how value-strategyproofness can be quantified. It is also possible to create a hybrid reputation system with rank-strategyproofness properties. Let M_{hitting} be the hitting-time based reputation system, and let M_{shortest} be the shortest-path based reputation system. Let $M_\alpha(M_{\text{hitting}}, M_{\text{shortest}})$ be the α-hybrid of $M_{\text{hitting}}, M_{\text{shortest}}$. M_{shortest} is rank- and value-strategyproof, but intuitively appears to use the least information about the trust graph. M_{hitting} is in some sense as informative as PageRank, but retains value-strategyproofness. Thus, it is reasonable to expect M_α might have better incentive-compatibility properties than M_{hitting} and better informativeness and efficiency than M_{shortest}.

By immediate application of Lemma 4 and Lemma 7, M_α is value-strategyproof but not necessarily rank-strategyproof. But, like the PageRank-maxflow hybrid, an additional result can be shown using the relaxed form of rank-strategyproofness.

Theorem 13. *$M_\alpha(M_{hitting}, M_{shortest})$ is α-rank-strategyproof on all trust graphs $G = (V, E, w)$.*

Proof. Given a trust graph G, an agent u's reputation from agent v_i's perspective under M_α has two components:

$$f_i(G,u) = \alpha f_i^{\text{hitting}}(G,u) + (1-\alpha) f_i^{\text{shortest}}(G,u)$$

Since M_{shortest} is rank-strategyproof and value-strategyproof, for any agent v and under any manipulation σ yielding $G' = G \downarrow \sigma$,

$$f_i^{\text{shortest}}(G,u) \leq f_i^{\text{shortest}}(G,v) \Rightarrow f_i^{\text{shortest}}(G',u) \leq f_i^{\text{shortest}}(G',v) \tag{3.1}$$

For the hitting time componnet of the reputation, by Theorem 3, under any manipulation σ yielding $G' = G \downarrow \sigma$, agent u cannot surpass an agent w whose reputation is twice that of agent u. More formally,

$$2 f_i^{\text{hitting}}(G,u) \leq f_i^{\text{hitting}}(G,w) \Rightarrow f_i^{\text{hitting}}(G',u) \leq f_i^{\text{hitting}}(G',w)$$

Multiply the above relation by α to get:

$$2\alpha f_i^{\text{hitting}}(G,u) \leq \alpha f_i^{\text{hitting}}(G,w) \Rightarrow \alpha f_i^{\text{hitting}}(G',u) \leq \alpha f_i^{\text{hitting}}(G',w) \tag{3.2}$$

Next, multiply equation (3.1) by $(1-\alpha)$ to get

$$(1-\alpha)\, f_i^{\text{shortest}}(G,u) \leq (1-\alpha)\, f_i^{\text{shortest}}(G,v) \Rightarrow (1-\alpha)\, f_i^{\text{shortest}}(G',u) \leq (1-\alpha)\, f_i^{\text{shortest}}(G',v) \quad (3.3)$$

Finally, we sum equations (3.2) and (3.3) to get:

$$\begin{aligned} 2\alpha f_i^{\text{hitting}}(G,u) + (1-\alpha) f_i^{\text{shortest}}(G,u) &\leq \alpha f_i^{\text{hitting}}(G,w) + (1-\alpha) f_i^{\text{shortest}}(G,w) \\ \Rightarrow \alpha f_i^{\text{hitting}}(G',u) + (1-\alpha) f_i^{\text{shortest}}(G',u) &\leq \alpha f_i^{\text{hitting}}(G',w) + (1-\alpha) f_i^{\text{shortest}}(G',w) \end{aligned}$$

or alternatively,

$$\begin{aligned} \alpha f_i^{\text{hitting}}(G,u) + f_i^{\alpha}(G,u) &\leq f_i^{\alpha}(G,w) \\ \Rightarrow f_i^{\alpha}(G',u) &\leq f_i^{\alpha}(G',w) \end{aligned}$$

Since hitting time reputation is a probability on a random graph, for any G $f_i^{\text{hitting}}(G,u) \leq 1$. Plugging this into the above equation yields the relation

$$\alpha + f_i^{\alpha}(G,u) \leq f_i^{\alpha}(G,w) \Rightarrow f_i^{\alpha}(G',u) \leq f_i^{\alpha}(G',w)$$

which is exactly the definition of α-rank-strategyproofness. □

Again, this relationship allows incentive compatibility of the M_α reputation system to be adjusted. Depending on the informativeness properties of this algorithm under different values of α, there may exist an optimal tradeoff between incentive-compatibility and informativeness.

Chapter 4

Experimental Results

Next, we empirically analyze the informativeness and economic efficiency properties of the shortest-path / hitting time hybrid reputation system in the problem domain of peer-to-peer file sharing. Our goal is to show that the informativeness metric is closely related to the economic efficiency of the system.

4.1 Experimental Setup

Following the model of the Eigentrust paper [15], for the simulations we created a model of a file-sharing system with a collection of well-behaved agents (which always exchange authentic files) and a collection of malicious agents which share inauthentic files with some probability.

We model this by initializing each agent with a type p, which determines the probability of sharing inauthentic files. Well-behaved agents have $p = 1$, while malicious agents were initialized with some probability of sharing inauthentic files $p \in [0, 1]$. The profile of malicious agents is generated once for each set of test parameters and shared across different initial graph topologies. This serves to reduce noise. Information on the structure of the network is drawn from real-world studies of such networks and will be discussed in the next section [22, 16].

4.1.1 Graph Topology

Real peer-to-peer networks display a power-law degree distribution with a few highly connected nodes and many poorly-connected nodes [22]. To create a realistic model of the structure of a peer-to-peer graph, I used the preferential attachment model (see [16]) to construct graphs that obey

power-law-like degree distributions. Starting with a pair of nodes that are connected to one another, new nodes are incrementally added. Each new node is connected to exactly one of the old nodes. The exact node it is connected to is randomly chosen, where the probability of choosing a given node *v* is equal to the indegree of *v* divided by the number of edges in the graph. This makes it more likely for nodes with large numbers of edges to become more highly connected.

This graph defines the initial topology of the network. As the simulation progresses, and as agents interact with one another, the degree distribution of the graph will change. This process is dependent on the decision framework used by the individual agents.

4.1.2 Decision Framework

At each time step, with constant probability, an agent *i* chooses to download a file. A random set of responding agents *K* is chosen from the set of all agents. Agent *i* then calculates the trust value of each agent in *K* using the reputation system being tested.

We initially tested two different rules for determining whether an interaction takes place. The first rule is the deterministic δ-greedy rule ($\delta = 0.1$). Under this rule, an agent chooses to download from a random responding agent j with probability δ. This helps agents discover new connections on the trust graph. With probability $1-\delta$, the agent downloads from the highest-reputation agent j that responds (note that the reputations all lie in $[0,1]$).

The second rule is the reputation-weighted-random decision rule. Under this rule, the probability of interacting with any of the responding agents is weighted by its current reputation score. Agents which have not been interacted with previously are given a default probability of 0.10, after which the probabilities are normalized. Agents with higher reputation are more likely to be chosen for download.

Preliminary tests of the dynamics of this indicated that the δ-greedy update rule more appropriately maintained the structure of the peer-to-peer network, so all simulations were run with the δ-greedy rule. Appendix A has a brief word about the preliminary testing.

If agent *i* chooses to interact, and agent j is malicious, agent j sends an inauthentic file with probability p_j.

Following the interaction, agent *i* applies an update rule to the trust graph. If a malicious file was sent from agent j, agent *i* severs its link entirely with agent j if one exists. If a good file was sent, agent *i* creates a link to j if one did not already exist (*i.e.,* if this occured through random exploration). This grim trigger update rule is severe and leads to fast isolation of malicious nodes.

4.2 Measurements: efficiency and informativeness

The measure of economic efficiency falls out naturally in the file-sharing domain: use the ratio of the number of authentic files versus inauthentic files exchanged over the network. This is appropriate as a measure of general social welfare because it aligns with the purpose of the filesharing network – allow users to transfer files quickly.

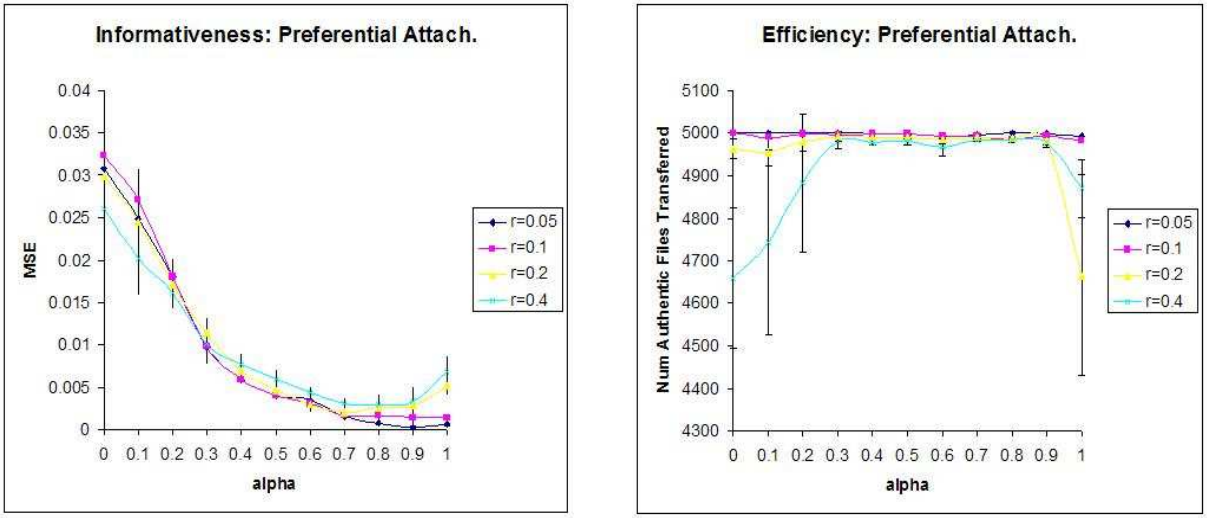
For informativeness, agent types are determined by the probability p of sending a malicious file. For each reputation system, the final reputations of each agent are normalized so that the agent with the highest reputation has a reputation of 1. This ensures that the range of the reputation system coincides with the possible agent types. The informativeness metric is the mean squared error between the reputation score for each agent and the type of each agent (more precisely, since the reputation system is asymmetric, I compare the reputation score for each agent from the perspective of every other agent against the true type).

4.3 Experimental Results

These tests were run on a system with 50 agents for 100 timesteps. We varied both the weighting factor of the α-hybrid reputation system $\alpha \in \{0.0, 0.1, ..., 1.0\}$ as well as the proportion of malicious agents $\beta = \{5\%, 10\%, 20\%, 40\%\}$. Because of the stochastic nature of the simulations 10 trials were run at each combination of α and β and the data averaged across these trials. To further reduce noise, ten different random initial graph topologies were generated under each graph generation method and reused.

For each trial the data gathered includes the efficiency in terms of authentic files transferred as well as the informativeness, calculated according to Definition 2.6.1 as the mean-squared-error between actual types and reputation score. The following results show MSE and authentic/inauthentic ratio for different proportions of well-behaved vs. malicious agents and different values of α. $\alpha = 0$ corresponds to the pure shortest-path algorithm, while $\alpha = 1$ corresponds to the pure hitting time algorithm. These tests were all run using graphs generated with a preferential attachment model, using the δ-greedy decision rule and grim-trigger update rule.

Figure 4.1 (a) shows an interesting trend. The mean square error starts out high under the shortest-path reputation system and drops as it moves towards the hitting-time algorithm (Note: since mean squared error is being graphed, the lower the error the more informative the algorithm



(a) (b)

Figure 4.1: (a) Mean squared error (informativeness) after running the simulation for different values of α (b) Number of authentic files (measure of efficiency) transferred over the course of the simulation. Standard error bars are plotted.

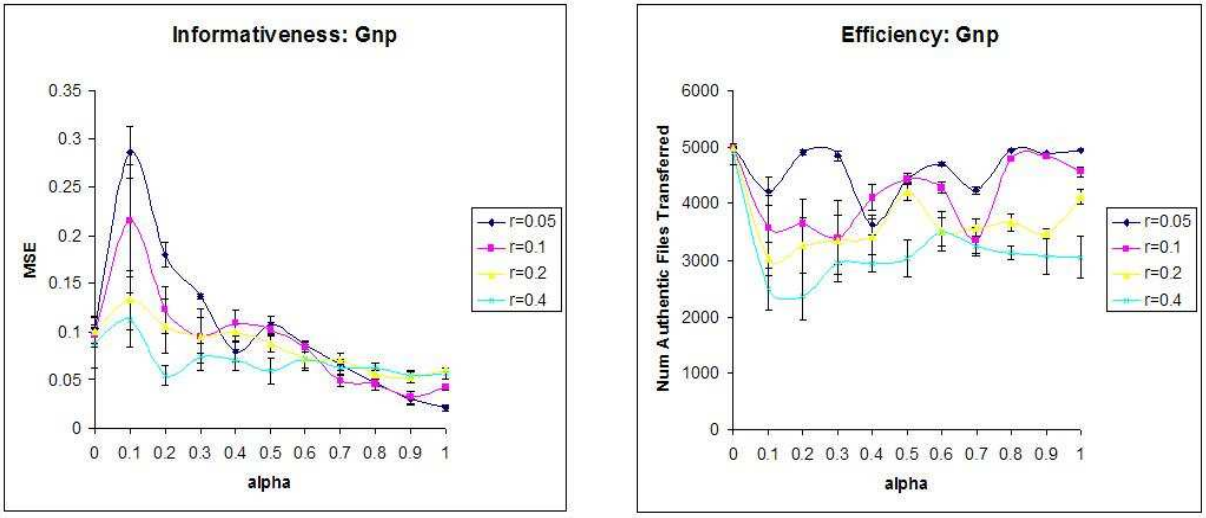
is). This is the expected result; since the hitting-time reputation system takes more information about the trust graph into account than the shortest-path system, we expect to see informativeness increase as α increases and more weight is put on hitting time.

It is also interesting that informativeness appears to be maximized somewhere in between the pure hitting time and pure shortest-path algorithms, with $\alpha = 0.9$, but this pattern falls within standard error. One possibility direction for future work is investigating whether the optimal informativeness does indeed occur somewhere between $\alpha = 0$ and $\alpha = 1$.

There is a similar pattern in the efficiency ratings of the different algorithms: both the hitting time and shortest-path algorithm perform worse than the hybridization of the two when the proportion of malicious agents is sufficiently large (*i.e.*, when $r = 0.2$ or $r = 0.4$).

This is a promising result: as informativeness rises, so does the economic efficiency of the reputation system. However, since efficiency is always relatively high, the trend is not entirely compelling.

The data for the graphs generated under the $G_{n,p}$ model is shown in Figure 4.2. These tests also used the δ-greedy decision rule and grim-trigger update rule. Though the data is noisier, the same general trends apply: as α increases, and as informativeness increases, we see an increase in efficiency as well.



(a) (b)

Figure 4.2: (a) Mean squared error (informativeness) after running the simulation for different values of α (b) Number of authentic files (measure of efficiency) transferred over the course of the simulation.

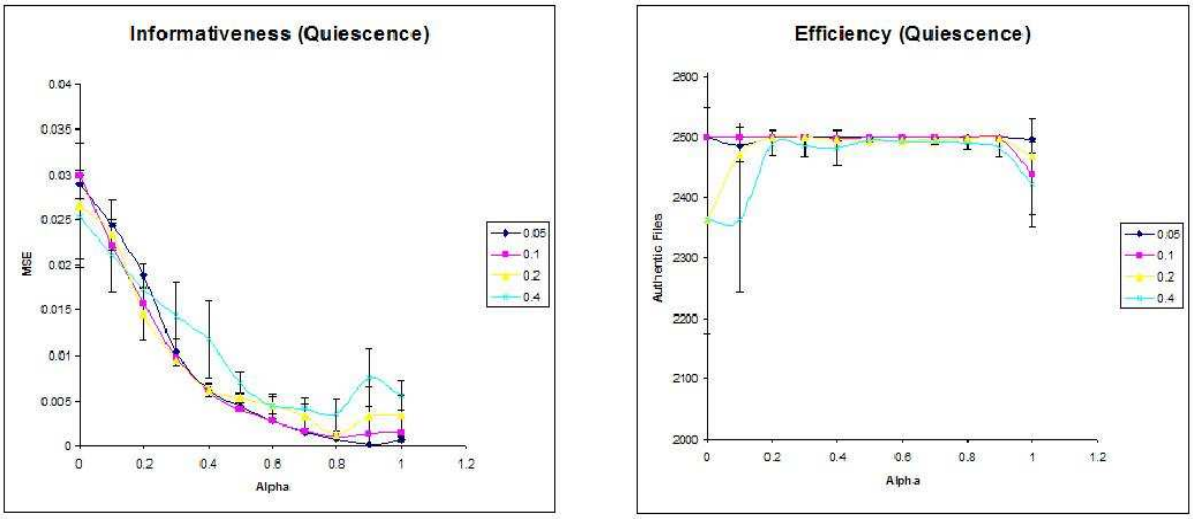
4.3.1 Quiescence Tests

The first few timesteps of any simulation are generally chaotic as the reputation system slowly converges towards the true reputations of each agent – this introduces noise into the final data. Instead, in general the metric is least noisy when evaluated on the steady-state behavior of any given reputation system.

To this end, we developed the following empirical criterion to determine when a reputation system has reached steady-state (quiescence): for each agent *v* and *w*, compare the change in reputation of agent *v* from agent *w*'s perspective against a fixed threshold ($\delta = 0.1$). If the change exceeds this threshold, count it as an absolute change. Finally, compute the average number of changes from one round to the next. When this dips below another threshold $(\delta = 0.005)$, we assume the system has reached a steady-state.

From the steady-state, the simulation is then run for a specified number of time-steps. The following simulations were run with 50 agents for 50 timesteps after quiescence. Each trial consisted of 5 separate and independent runs.

The general trends under the quiescence tests closely mirror the results from the standard simulation. These results are included primarly because the standard error (plotted as the standard error bars) is considerably lower in both the informativeness and efficiency measurements. This gives



(a) (b)

Figure 4.3: (a) Mean squared error (informativeness) after running the simulation as a function of α (b) Number of authentic files (measure of efficiency) transferred over the course of the simulation.

confidence that the trends observed in Figure 4.3 are not a by-product of the randomness of the process.

4.4 Informativeness metric evaluation

The informativeness metric I developed appears to effectively mirror the economic efficiency metric. As informativeness increases, the general trend is for efficiency to increase as well. Though more simulation is necessary for this to be convincing, the fact that this result is robust under the quiescence test and under different initial graph topologies is highly suggestive.

As a side note, the informativeness metric is effective at separating the relatively uninformative shortest-path reputation system from the hitting time reputation; the informativeness rises sharply as we increase α to bias towards hitting time.

4.5 Hybridization evaluation

The M_α construct is able to deliver different levels of informativeness in exchange for sacrificing incentive compatibility. The results suggest that a locally optimal reputation system configuration exists somewhere between our two chosen reputation systems. Interesting questions for future work

include testing different hybrid algorithms to see if we also find locally optimal performance between $\alpha = 0$ and $\alpha = 1$.

This provides validation for the empirical approach: my formulation allows for quantitative predictions. Starting with a reputation system with known properties, it is possible to trade off incentive compatibility in return for informativeness, and by extension economic efficiency. This suggests interesting potential applications for this method in designing custom reputation systems. By creating and tuning a α-hybrid, we can creating a novel reputation system which is tuned for a particular context.

Chapter 5

Conclusions

5.1 Summary

The overarching concern of a reputation system is to provide users with information that allow for good decisions, *i.e.*, economic efficiency. In order to do this, a reputation system must be both incentive compatible (IC) and informative. Incentive compatibility ensures that wasteful optimization and manipulation of the reputation system does not occur, while informativeness ensures that each agent has accurate information to make its decision. In Chapter 1, we discussed past work in this field which tended to focus on one of these reputation system properties over the other, *i.e.*, focusing either on proving formal incentive compatibility properties or on evaluating the performance of individual reputation systems.

One primary contribution of this thesis was recognizing the tension between incentive compatibility and economic efficiency, two desirable reputation system properties. As reputation systems move towards stronger IC, they tend to become less efficient. This trend motivated the definition of a range of different incentive-compatibility concepts, including value-strategyproofness, rank-strategyproofness, and the novel ε-rank and ε-value-strategyproof relaxations, in order to better characterize this inherent tradeoff. Reputation systems like shortest paths and maxflow satisfying rank- or value-strategyproofness performed poorly in simulation against reputation systems with no IC guarantees. This pattern was revealed only after analyzing both the IC properties and the empirical performance of a wide variety of existing reputation systems, including shortest-paths, maxflow, hitting time, and eigenvector algorithms.

In order to investigate the nature of the aforementioned tension, we developed a method for

trading off the incentive compatibility properties of a reputation system against informativeness and efficiency properties. The α-hybridization technique discussed in Chapter 3 allows for the creation of new reputation systems by taking a convex combination of two different reputation systems. If chosen carefully, by adjusting the weighting parameter α it is possible to tuned for informativeness or IC. Following the model of Altman and others [4, 3, 24], we theoretically characterize the IC properties of our hybrid construct.

The next major contribution was the development of an informativeness metric for reputation systems which was empirically shown to correlate well with the economic efficiency of the reputation system. The intuition behind this is reasonable; reputation systems that use more trust graph information to generate their reputation scores generally encode more information about the trust graph; this additional information gives the agents in the system a better chance of making socially beneficial decisions. And because reputation systems are used in so many different contexts, it is difficult to develop economic efficiency metrics which are generalizable or even tractable. The informativeness metric is simple in concept and enables useful comparisons among many existing reputation systems.

Our experiments under a peer-to-peer file-sharing domain served to validate the informativeness metric as both practically usable and as a good proxy for efficiency. Using reasonably faithful models of real peer-to-peer systems, we showed that both the informativeness and the efficiency of the hitting-time/shortest-path hybrid increase dramatically as we change the weighting factor α to emphasize the hitting-time reputation system. This result remains robust under changes in the initial graph topology and under steady-state quiescence testing.

5.2 A Recommendation System Application

Peer-to-peer networks benefit generally from the application of reputation systems; these benefits are somewhat tangential to the primary contributions of this thesis – *i.e.*, ways of measuring informativeness and ways of constructing informative reputation systems. Recommendation systems, however, offer a compelling application domain for this work.

Reputation systems in online systems are often accompanied by systems for making personalized recommendations to end users. Systems like Amazon's book-recommendation service or NetFlix's movie-rental service suggest new products that shoppers may enjoy based on their past history of purchases. Because online retailers have information on the purchasing patterns of so

many different buyers, they can make well-informed suggestions that customers are willing to investigate. There is real financial incentive to construct these systems properly; recommendations can uncover material that customers are willing to pay for but did not know existed.

While much of the existing literature on recommendation systems is based on collaborative filtering techniques, work by Andersen, Chayes *et al.* [5] has investigated building a recommendation system on top of an existing trust graph, leveraging the information contained in social relationships. Following Andersen's model, a *voting network* is built on top of an existing trust graph by annotating a subset of nodes with "votes" (either $+$ or $-$).

Definition 5.2.1. (Voting Network) A voting network is a directed, annotated graph $G = (N, V_+, V_-, E)$ where N is a set of nodes, $V_+, V_- \subseteq N$ are disjoint subsets of positive and negative voters, and $E \subseteq N \times N$ is a set of edges. Let $V = V_+ \cup V_-$ denote the set of voters, and let $V' = N \, V$ denote the set of nonvoters.

Definition 5.2.2. (Recommendation system) Given a voting network G and a specific nonvoter $s \in V'$, a recommendation system outputs a recommendation $R(G,s) \in \{-,0,+\}$.

The authors state and prove axioms characterizing several different recommendation system algorithms. However, under the above formulation, I will demonstrate that a recommendation system problem can be reduced to a standard reputation system problem.

Definition 5.2.3. (Recommendation-reputation reduction) Given a voting network $G = (N, V_+, V_-, E)$, define its reputation-reduction trust graph $G' = (N \cup \{-,+\}, E \cup E_T, w)$ where $e = (n,-) \in E_T$ iff $n \in V_+$, and $e = (n,+) \in E_T$ iff $n \in V_-$. The weight function $w(e) = 1$ for $e \in E \cup E_T$.

In other words, a new good-node is inserted in the trust graph for each alternative. For each voter $v \in V$ that voted for a particular alternative a an edge $(v,a) \in E_T$ is created. Once we have our trust graph, given any agent $n \in N$, we can run any reputation system to get a ranking of all the agents from agent n's perspective; the highest ranked alternative is returned as the recommendation.

There are several advantages to the proposed reduction. First, under the reputation system formulation it is easy to extend the recommendation problem to situations with more expressive preferences and multiple goods. We can add weights on edges from agent-nodes to good-nodes to represent more expressive preferences, and we can add more goods by simply adding additional good-nodes.

By encoding both trust information and opinion information into the same trust graph, we simplify the problem and allow reputation system techniques to be applied to this new domain of

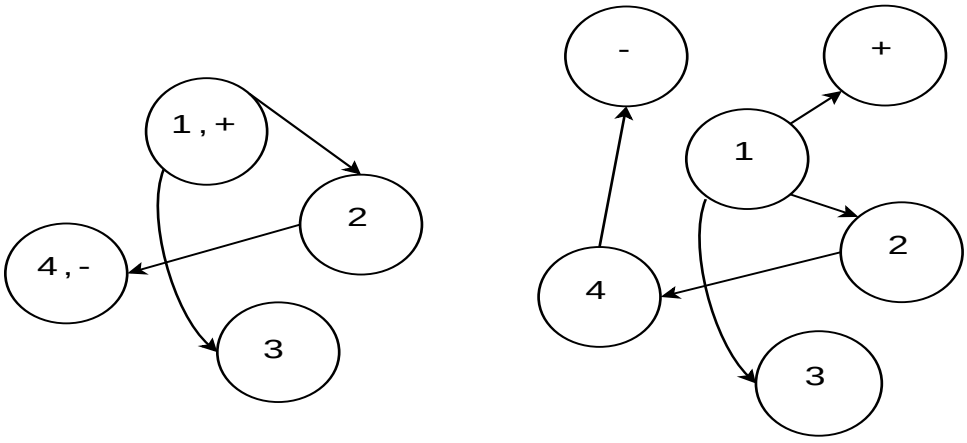


Figure 5.1: The image on the left shows a voting network complete with annotated nodes. On the right, we show the reputation-reduced version of the same voting network; the alternatives now have their own nodes, and each node that was annotated has a link to the corresponding alternative.

trust-based recommendation. The work presented in this thesis on measuring informativeness and constructing informative reputation systems is especially well-suited for the problem of accurately predicting which goods appeal to which users. Because the end user either likes or dislikes the end recommendation, informativeness is almost exactly equivalent to economic efficiency in this domain. This makes trust-based recommendation systems a promising open topic for future work in this space.

5.3 Open Problems

The real promise of reputation systems lie in their application to real, large-scale systems. There remain significant challenges in scaling up the techniques outlined in this thesis to real-world problems. We have so far ignored issues of scalability and computational efficiency in order to develop compelling metrics for evaluating and tuning reputation systems; these issues must be addressed before the benefits of our reputation system work are fully realized.

The applications we have discussed in this paper depend on analyzing and drawing trust information from the activities of thousands or hundreds of thousands of individual users and agents. I have not considered computational efficiency issues in this work, focusing instead on issues facing the end-user: informativeness, economic efficiency, and incentive compatibility; however, a reputation system which takes too long to run or too much space to compute is of little practical use.

Polynomial time algorithms do exist for both the hitting-time and shortest-path algorithms; however, for datasets on the scale of Amazon's book rating database more significant optimization may be necessary.

These issues are even more complicated when the setting changes to a distributed, peer-to-peer domain. Throughout my analysis I have implicitly assumed the existence of a trusted center which can gather agent reports and run the reputation computation. In a peer-to-peer network this computation must either be duplicated at each node (probably prohibitively expensive) or distributed across the network. However, distributed computation raises a number of challenging incentive-compatibility issues which do not arise in the centralized setting; see [12] for a more in-depth description of decentralized mechanism design. If computation of the reputation system is distributed, agents may be able to influence their ranking not only by misreporting interactions but by deviating from the preprogrammed reputation computation algorithm, opening up new classes of manipulations that have not been defined in this thesis. Developing manipulation-resistant distributed algorithms remains a difficult open problem.

5.4 Conclusion and Outlook

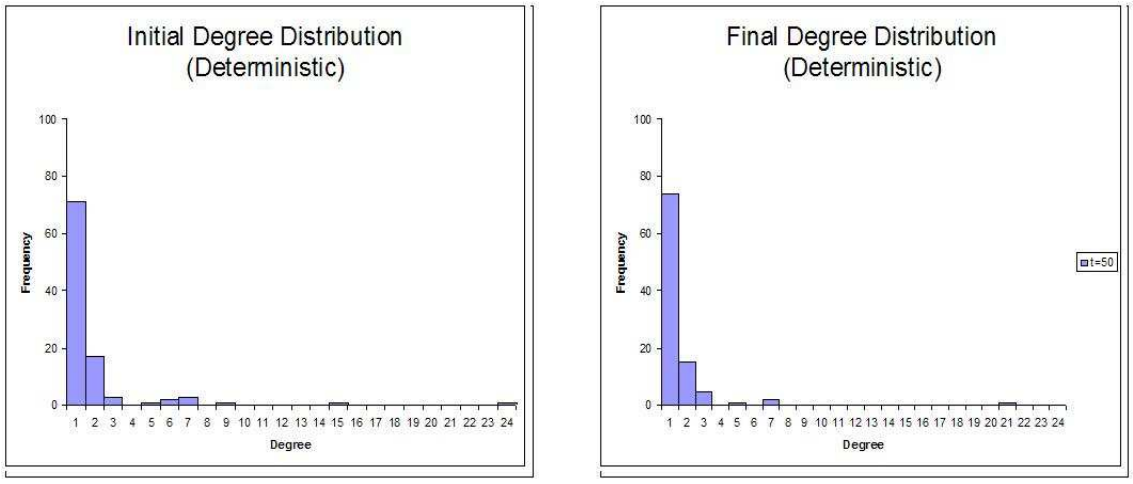
Because of the wide range of potential applications, reputation systems are likely to play an integral role in the evolution of the Internet. Reputation systems drive commerce on online retailers. They control download speeds on peer to peer networks. They extract information from the hyperlink structure of the web itself. And as the amount of trust and interaction information grows, so does the demand for reputation systems that can take advantage of this information. The work we have done on informativeness metrics will make it simple for users to compare different reputation systems and pick the one that achieves the highest level of economic efficiency.

But at a higher level, our work on α-hybrid reputation systems allow the construction of reputation systems with exactly the right tradeoff between informativeness and incentive compatibility. A web site which offers houses for sale probably merits a reputation system with stronger incentive compatibility properties than a web site for children's toys. The ability to build exactly the right reputation system for the task is sure to benefit the users of such systems. Investigating the applications and the limits of this approach is an exciting area for future research.

Appendix A

Decision Rule Selection

The two decision rules yield vastly different dynamics in terms of the evolving graph topology. Preliminary simulations with 25 agents were run for 50 rounds; using the data I generated the following plots of the degree distribution of the peer-to-peer network over time under the reputation-weighted-random and δ-greedy decision rules.

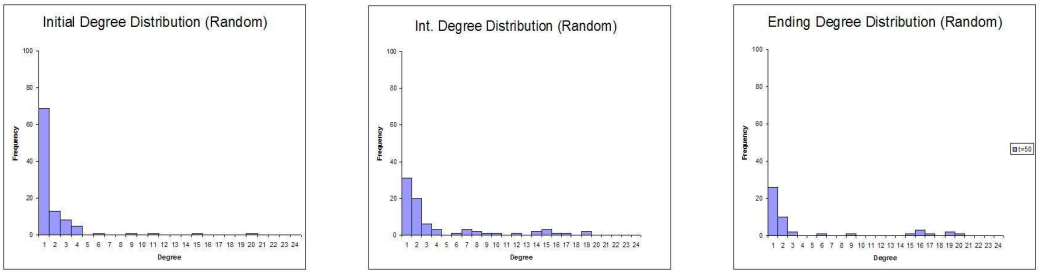


(a) (b)

Figure A.1: (a) Initial degree distribution for δ-greedy rule (b) Ending degree distribution.

Figure A.1 shows the δ-greedy results. A few agents gained or lost links; one agent ended up fully connected to other nodes in the network. The δ-greedy method preserves the initial power-law like distribution.

Behavior under the random algorithm (see A.2) is much different. By emphasizing exploration,



(a) (b) (c)

Figure A.2: (a) Initial degree distribution under the reputation-weighted-random rule (b) Intermediate degree distribution (c) Ending degree distribution.

this causes good nodes to gain links with one another very quickly; bad nodes are isolated near the bottom. However, because the distribution quickly moves away from the initial power-law degree distribution, I chose to do most of the testing using the δ-greedy update rule.

Bibliography

[1] P2P Survey 2006 - Extended Abstract. http://www.ipoque.com/userfiles/file/P2P-Survey-2006.pdf, 2006.

[2] Alon Altman and Moshe Tennenholtz. On the Axiomatic Foundations of Ranking Systems. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 917–922, 2005. Submitted to Journal of the ACM.

[3] Alon Altman and Moshe Tennenholtz. Quantifying Incentive Compatibility of Ranking Systems. In *Proc. 20th International Joint Conference on Artificial Intelligence*, 2006.

[4] Alon Altman and Moshe Tennenholtz. An Axiomatic Approach to Personalized Ranking Systems. Submitted to Journal of the ACM, 2007.

[5] Reid Andersen, Christian Borgs, Jennifer Chayes, Uriel Feige, Abraham Flaxman, Adam Kalai, Vahab Mirrokni, and Moshe Tennenholtz. Trust-based recommendation systems: an axiomatic approach.

[6] K. Arrow. *Social Choice and Individual Values*. Yale University Press, 2nd edition, 1963.

[7] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside Pagerank. *ACM Transactions on Internet Technology*, 5(1), 2005.

[8] Alice Cheng and Eric Friedman. Sybilproof Reputation Mechanisms. In *Proceedings of the ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems (P2PECON)*, pages 128–132, Philadelphia, PA, August 2005.

[9] Alice Cheng and Eric Friedman. Manipulability of PageRank under Sybil Strategies. In *Proceedings of hte First Workshop of Networked Systems (NetEcon06)*, 2006.

[10] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2003.

[11] J. Douceur. The Sybil Attack. In *IPTPS02 Workshop*, 2002.

[12] Joan Feigenbaum and Scott Shenker. Distributed Algorithmic Mechanism Design: Recent Results and Future Directions. In *6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 1–13, Atlanta, Georgia, September 2002.

[13] Michal Feldman, Christos Papadimitriou, John Chuang, and Ion Stoica. Free-Riding and Whitewashing in Peer-to-Peer Systems. *IEEE Journal on Selected Areas in Communications*, 24:5:1010–1019, May 2006.

[14] E. Friedman and P. Resnick. The Social Cost of Cheap Pseudonyms. *Journal of Economics and Management Strategy*, 10 (1), 2001.

[15] Sepandar Kamvar, Mario Schlosser, and Hector Garcia-Molina. The Eigentrust Algorithm for Reputation Management in p2p Networks. In *Twelfth International World Wide Web Conference*, pages 640–651, 2003.

[16] Michael Mitzenmacher. A brief history of generative models for power law and lognormal distributions, 2001. M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. Manuscript.

[17] Markus M. Mobius and Adam Szeidl. Trust and Social Collateral. NBER Working Paper No. W13126, May 2007.

[18] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[19] David Christopher Parkes. *Iterative combinatorial auctions: achieving economic and computational efficiency*. PhD thesis, University of Pennsylvania, 2001.

[20] P. Resnick and R. Zeckhauser. Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. Technical report, University of Michigan, 2001.

[21] Paul Resnick, Richard Zeckhauser, John Swanson, and Kate Lockwood. The value of reputation on eBay: A controlled experiment. *Experimental Economics*, 9:79–101, 2006.

[22] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A Measurement Study of Peer-To-Peer File Sharing Systems. In *Proceedings of Multimedia Computing and Networking 2002 (MMCN '02)*, San Jose, CA, January 2002.

[23] Dan Sheldon and John Hopcroft. Manipulation-Resistant Reputations Using Hitting Time. In *5th Workshop on Algorithms for the Web-Graph*, 2007.

[24] Moshe Tennenholtz. Reputation systems: An axiomatic approach. In *Proceedings of the 20th conference on uncertainty in Artificial Intelligence (UAI-04)*, 2004.