Manipulation-resistant Reputations using Hitting Time*

John Hopcroft
jeh@cs.cornell.edu

Daniel Sheldon
dsheldon@cs.cornell.edu

July 3, 2007

Abstract

Popular reputation systems for linked networks can be manipulated by spammers who strategically place links. The reputation of node v is interpreted as the world's opinion of v's importance. In PageRank [7], v's own opinion can be seen to have considerable influence on her reputation, where v expresses a high opinion of herself by participating in short directed cycles. In contrast, we show that expected hitting time — the time to reach v in a random walk — measures essentially the same quantity as PageRank, but excludes v's opinion. We make these notions precise, and show that a reputation system based on hitting time resists tampering by individuals or groups who strategically place outlinks. We also present an algorithm to efficiently compute hitting time for all nodes in a massive graph; conventional algorithms do not scale adequately. Our algorithm, which applies to any random walk with restart, exploits a relationship between PageRank and hitting time in random walks with restart. This relationship also provides novel insights into spam detection and PageRank computation.

*This material is based upon work supported by the National Science Foundation under Grant No. 0514429. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

1 Introduction

Reputation and ranking systems are an essential part of web search and e-commerce. The general idea is that the reputation of one participant is determined by the endorsements of others; for example, one web page endorses another by linking to it. However, any web user knows that not all participants are honorable — spammers will do their best to manipulate a search engine's rankings. A very basic requirement for a reputation system is that individuals should not be able to improve their own reputation using simple self-endorsement strategies, like participating in short cycles to boost PageRank. Many popular systems, including PageRank, fail in this regard. Since PageRank enjoys many nice properties, it is instructive to see where things go wrong.

Let $G = (V, E)$ be a directed graph (e.g, the web). PageRank assigns a score $\pi(v)$ to each node v, where π is defined to be the stationary distribution of a random walk on G, giving the pleasing interpretation that the score of page v is the fraction of time a web surfer spends there if she randomly follows links forever. For technical reasons, the random walk is modified to restart in each step with probability α, jumping to a page chosen uniformly at random. This ensures that π exists and is efficient to compute. Then a well-known fact about Markov chains [1] says that $1/\pi(v)$ is equal to the expected *return time* of v, the number of steps it takes a random walk starting at v to return to v. A heuristic argument for this equivalence is that a walk returning to v every r steps on average should spend $1/r$ of all time steps there.

Despite its popularity, one can easily manipulate return time with a strategy that changes *only outlinks.* Intuitively, a node v should only link to nodes from which a random walk is expected to return to v very quickly. By partnering with just one other node to form a 2-cycle with no other outlinks, v ensures a return in two steps — the minimum possible without self-loops — unless the walk jumps first. In this fashion, v can often boost its PageRank by a factor of 3 to 4 for typical settings of α [9]. However, this strategy relies on manipulating the portion of the walk before the first jump: the jump destination is independent of v's outlinks, and return time is determined once the walk reaches v again, so v's outlinks have no further effect. This suggests eliminating the initial portion of the walk and measuring reputation by the time to hit v following a restart, called the *hitting time* of node v from the restart distribution. This paper develops a reputation system based on hitting time that is provably resistant to manipulation. Our main contributions are:

- In Theorem 1, we develop a precise relationship between expected return time and expected hitting in a random walk with restart, and show that the hitting time of v is completely determined by the probability that v is reached before the first jump.
- In Theorem 2, we prove that a reputation system based on hitting time resists manipulation, using natural definitions of reputation and influence derived from Theorem 1. For example, node v has a limited amount of influence that depends on her reputation, and she may spread that influence using outlinks to increase others' reputations. However, node v cannot alter her own reputation with outlinks, nor can she damage w's reputation by more than her original influence on w. Furthermore, the advantage that v gains by purchasing new nodes, often called *sybils* of v, is limited by the restart probability of the sybils.
- To realize a reputation system based on hitting time we present an efficient algorithm to simultaneously compute hitting time for all nodes. In addition to one PageRank calculation, our algorithm uses Monte Carlo sampling with running time that is linear in $|V|$ for given accuracy and confidence parameters. This is a significant improvement over traditional algorithms, which require a large-scale computation for each node.[1]

[1]Standard techniques can simultaneously compute hitting time from all possible sources to a single target node

We mention a few additional consequences of our work in less detail. For example, we describe a Monte Carlo algorithm to find the most reputable nodes in time that is independent of the size of the graph. We give an example showing that Theorem 1 can be used to easily reason about optimal strategies for the placement of links to maximize reputation. Finally, we show how to pre-compute a small number of values per node to quickly and accurately estimate the hitting time or PageRank for *any choice of restart probability* α that is not too small.

The rest of the paper is structured as follows. In section 2 we discuss related work. In section 3 we present Theorem 1, giving the characterization of hitting time that is the foundation for the following sections. In section 4 we develop a reputation system using hitting time and show that it is resistant to manipulation. In section 5 we present algorithms for computing hitting time. In section 6 we discuss additional consequences of Theorem 1.

2 Related Work

PageRank [7] and HITS [23] were early reputation systems used for web search. Since then, PageRank has been adapted to a variety of applications: personalized web search [30], web spam detection [18], and trust systems in peer-to-peer networks [22]. Each of these uses the same general formulation and our work applies to all of them. Recently, personalized PageRank was used as the basis of a graph partitioning algorithm that runs in nearly-linear time [3].

Much work has focused on the PageRank system itself, studying computation methods, convergence properties, stability and sensitivity, and, of course, implementation techniques. See [24] for a survey of this wide body of work. Computationally, the Monte Carlo methods in [12] and [4] are similar to our algorithms for hitting time. They use a probabilistic formulation of PageRank in terms of a *short* random walk [21] that permits efficient sampling. In particular, we will use the same idea as [12] to efficiently implement many random walks simultaneously in a massive graph, without requiring random access.

Manipulability of PageRank falls under the umbrella of stability under perturbation. [29] gave an early stability result: the total change in PageRank can not be too great if only low PageRank pages change their links; their bounds were improved by [5]. [10] showed that PageRank is *monotonic* — the addition of an inlink cannot decrease a page's PageRank. [6] defined *rank-stability*, which captures the idea that small changes to the link structure should not dramatically change the rankings given by an algorithm, and [25] showed that PageRank is not rank-stable. Recent works have cast the stability question in a strategic light: how can a group of selfish nodes place outlinks to optimize their PageRank, and how can we detect such nodes [5, 9, 15–17, 27, 31]? In particular, [5, 9, 16] all describe the manipulation strategy mentioned in the introduction.

For a more general treatment of reputation systems in the presence of strategic agents, [13] is a nice overview with some specific results from the literature. [2] gives an axiomatic treatment of reputation systems and incentive compatibility, showing possibility and impossibility results under different combinations of axioms. [8] proves an impossibility result that relates to our work — a wide class of reputation systems (which includes ours) cannot be resistant to a particular attack called the *sybil attack* [11]. However, their definition of resistance is very strong, and our results can be viewed as positive results under a relaxation of this requirement. We will discuss sybils in section 5. [22] uses a central mechanism to make EigenTrust (essentially PageRank applied to peer-to-peer systems) provably non-manipulable; the mechanism prohibits links between certain nodes.

---

using a system of linear equations. However, what is desired for reputation systems is the hitting time from one source, or in this case a distribution, to all possible targets.

Hitting time is a classical quantity of interest in Markov chains. See chapter 2 of [1] for an overview. The exact terminology and definitions vary slightly: we define hitting time as a random variable, but sometimes it is defined as the expectation of the same random variable. Also, the term *first passage time* is sometimes used synonymously. Hitting time has appeared in contexts related to ours: in [26], hitting time between pairs of nodes is used as a measure of proximity to predict link formation in a social network. [20] uses hitting time to measure similarity between nodes; in particular, their *expected-f meeting distance* is the hitting time of a set of nodes in a random walk (with restart) on a product graph. [10] uses hitting time to analyze the monotonicity of PageRank.

Finally, the relationship between hitting time and return time in a random walk with restart is related to work on *regenerative stochastic processes* — processes that restart at certain intervals. In fact, Theorem 1 can be derived as a special case of a general result about such processes, where the process is Markovian and the regeneration times are geometric. See equation (15) in [19] and the references therein for a more general treatment.

3 Characterizing Hitting Time

Our goal is to build a reputation system using hitting time that is both informative and manipulation-resistant. This section will pave the way towards both goals by stating and proving Theorem 1. Part (i) of the theorem relates hitting time to return time — then we argue that hitting time is informative by comparison with PageRank — the two are essentially the same *except* for nodes where the random walk is likely to return before jumping, a known sign of manipulation. Part (ii) proves that the hitting time of v is completely determined by the probability that v is reached before the first jump; this will lead to precise notions of manipulation-resistance in section 4.

3.1 Preliminaries

Let $G = (V, E)$ be a directed graph. Consider the *standard random walk* on G, where the first node is chosen from starting distribution q, then at each step the walk follows an outgoing link from the current node chosen uniformly at random. Let $(X_t)_{t\geq 0}$ be the sequence of nodes visited by the walk. Then $\Pr[X_0 = v] = q(v)$, and $\Pr[X_t = v \mid X_{t-1} = u] = 1/\text{outdegree}(u)$ if $(u, v) \in E$, and zero otherwise. Here, we require $\text{outdegree}(u) > 0$.[2] Now, suppose the walk is modified to restart with probability α at each step, meaning the next node is chosen from the starting distribution (henceforth, *restart distribution*) instead of following a link. The new transition probabilities are:

$$\Pr[X_t = v \mid X_{t-1} = u] = \begin{cases} \alpha q(v) + \frac{1-\alpha}{\text{outdegree(u)}} & \text{if } (u, v) \in E \\ \alpha q(v) & \text{otherwise} \end{cases} .$$

We call this the *α-random walk* on G, and we parametrize quantities of interest by the restart probability α. A typical setting is $\alpha = 0.15$, so a jump occurs every $1/.15 \approx 7$ steps in expectation. The *hitting time* of v is $H_\alpha(v) = \min\{t : X_t = v\}$. The *return time* of v is $R_\alpha(v) = \min\{t \geq 1 : X_t = v \mid X_0 = v\}$. When v is understood, we simply write H_α and R_α. We write H and R for the hitting time and return time in a standard random walk.

3.2 Theorem 1

Before stating Theorem 1, we make the useful observation that we can split the α-random walk into two independent parts: (1) the portion preceding the first jump is the beginning of a standard

[2] This is a technical condition that can be resolved in a variety of ways, for example, by adding self-loops to nodes with no outlinks.

random walk, and (2) the portion following the first jump is an α-random walk independent of the first portion. The probability that the first jump occurs at time t is $(1-\alpha)^{t-1}\alpha$, i.e., the first jump time J is a geometric random variable with parameter α, independent of the nodes visited by the walk. Then our model for the α-random walk is: (1) start a standard random walk, (2) independently choose the first jump time J from a geometric distribution, and (3) at time J begin a new α-random walk. Hence we can express the return time and hitting time of v recursively:

$$R_\alpha = \begin{cases} R & \text{if } R < J \\ J + H_\alpha & \text{otherwise} \end{cases}, \qquad H_\alpha = \begin{cases} H & \text{if } H < J \\ J + H'_\alpha & \text{otherwise} \end{cases}. \tag{1}$$

Here H'_α is an independent copy of H_α. It is convenient to abstract from our specific setting and state Theorem 1 about general random variables of this form.

Theorem 1. *Let R and H be independent, nonnegative, integer-valued random variables, and let J be a geometric random variable with parameter α. Define R_α and H_α as in (1). Then,*

(i) $E[R_\alpha] = \Pr[R \geq J]\left(\frac{1}{\alpha} + E[H_\alpha]\right)$, *(ii)* $E[H_\alpha] = \frac{1}{\alpha} \cdot \frac{\Pr[H \geq J]}{\Pr[H < J]}$, *(iii)* $E[R_\alpha] = \frac{1}{\alpha} \cdot \frac{\Pr[R \geq J]}{\Pr[H < J]}$.

Part (i) relates expected return time to expected hitting time: $\Pr[R \geq J]$ is the probability that the walk does not return before jumping. On the web, for example, we expect $\Pr[R \geq J]$ to be close to 1 for most pages, so the two measures are roughly equivalent. However, pages attempting to optimize PageRank can drive $\Pr[R \geq J]$ much lower, achieving an expected return time that is much lower than expected hitting time.

For parts (ii) and (iii), we adopt the convention that $\Pr[H < J] = 0$ implies $E[H_\alpha] = E[R_\alpha] = \infty$, corresponding to the case when v is not reachable from any node with positive restart probability. To gain some intuition for part (ii) (part (iii) is similar), we can think of the random walk as a sequence of independent explorations from the restart distribution "looking" for node v. Each exploration succeeds in finding v with probability $\Pr[H < J]$, so the expected number of explorations until success is $1/\Pr[H < J]$. The expected number of steps until an exploration is terminated by a jump is $1/\alpha$, so a rough estimate of hitting time is $\frac{1}{\alpha} \cdot \frac{1}{\Pr[H<J]}$. Of course, this is an overestimate because the final exploration is cut short when v is reached, and the expected length of an exploration conditioned on not reaching v is slightly shorter than $1/\alpha$. It turns out that $\Pr[H \geq J]$ is exactly the factor needed to correct the estimate, due to the useful fact about geometric random variables[3] stated in Lemma 1. We stress that the hitting time of v in the α-random walk is completely determined by $\Pr[H < J]$, the probability that a given exploration succeeds; this will serve as the foundation for our reputation system.

Lemma 1. *Let X and J be independent random variables such that X is nonnegative and integer-valued, and J is a geometric random variable with parameter α. Then $E[\min(X, J)] = \frac{1}{\alpha}\Pr[X \geq J]$.*

Lemma 1 is proved in the appendix. Interestingly, Lemma 1 uniquely characterizes a geometric random variable, so the conclusions of Theorem 1 do not hold for arbitrary J. This is also proved in the appendix.

Proof of Theorem 1. We rewrite $R_\alpha = \min(R, J) + I\{R \geq J\}H_\alpha$, where $I\{R \geq J\}$ is the indicator variable for the event $R \geq J$. Note that $I\{R \geq J\}$ and H_α are independent. Then, using linearity of expectation and Lemma 1,

$$E[R_\alpha] = E[\min(R, J)] + \Pr[R \geq J]\, E[H_\alpha]$$

[3] We mentioned that Theorem 1 can be derived from a result about regenerative stochastic processes [19]. In fact, Theorem 1 captures most of the generality; to write recurrences as in (1), the process need not be Markovian, it is only necessary that the process following a restart is a replica of the original. The only non-general assumption made is that J is a geometric random variable; this simplifies the conclusions.

$$= \frac{1}{\alpha} \Pr [R \geq J] + \Pr [R \geq J] E [H_\alpha]$$
$$= \Pr [R \geq J] \left(\frac{1}{\alpha} + E [H_\alpha] \right).$$

This proves part (i). For part (ii), we take R to be a copy of H in part (i), giving

$$E [H_\alpha] = \Pr [H \geq J] \left(\frac{1}{\alpha} + E [H_\alpha] \right).$$

Solving this expression for $E [H_\alpha]$ gives (ii). Part (iii) is obtained by substituting (ii) into (i). □

4 Manipulation-resistance

In this section we develop a reputation system based on hitting time, and quantify the extent to which an individual can tamper with reputations. It is intuitively clear that u cannot improve her own hitting time by placing outlinks, but we would also like to limit the damage that u can cause to v's reputation. Specifically, u should only be able to damage v's reputation if u was responsible for v's reputation in the first place. Furthermore, u should not have a great influence on the reputation of too many others. To make these ideas precise, we define reputation using $\Pr [H < J]$ instead of $E [H_\alpha]$. By Theorem 1, either quantity determines the other — they are roughly inversely proportional — and $\Pr [H < J]$ is convenient for reasoning about manipulation.

Definition 1. *Let $f(v) = \Pr [H(v) < J]$ be the* reputation *of v.*

In words, $f(v)$ is the probability that a random walk hits v before jumping. Of all walks that reach v before jumping, an attacker u can only manipulate those that hit u first. This leads to our notion of influence.

Definition 2. *Let $g(u, v) = \Pr [H(u) < H(v) < J]$ be the* influence of u on v.

Definition 3. *Let $g(u) = \sum_v g(u, v)$ be the* total influence of u.

To quantify what can change when u manipulates outlinks, suppose G' is obtained from G by the addition or deletion of edges originating at u. Let f' and g' be the reputation and influence functions in G'. It is convenient to formalize the intuition that u has no control over the random walk until it hits u for the first time.

Definition 4. *For an event A, we say that A is u-*invariant *if $\Pr_G [A] = \Pr_{G'} [A]$. If A is u-invariant, we also say that the quantity $\Pr [A]$ is u-invariant.*

Lemma 2. *An event A is u-invariant if the occurrence or non-occurrence of A is determined by time $H(u)$.*

Lemma 2 is proved in the appendix. With the definitions in place, we can quantify how much u can manipulate reputations.

Theorem 2. *Let v be any node in V. Then*

(i) $f'(u) = f(u)$ (cannot alter own reputation),

(ii) $g(u, v) \leq f(u)$ (influence cannot exceed reputation),

(iii) $f'(v) \geq f(v) - g(u, v)$ *(cannot damage v's reputation by more than influence on v),*

(iv) $f'(v) \leq f(v) + f(u) - g(u, v)$ *(cannot boost v's reputation by more than reputation minus current influence),*

(v) $g(u) \leq \frac{1}{\alpha} f(u)$ *(total influence cannot exceed* $\frac{1}{\alpha}$ *times reputation).*

Proof. For part (i), we have $f'(u) = \Pr_{G'}[H(u) < J] = \Pr_G[H(u) < J] = f(u)$, since $H(u) < J$ is u-invariant. For part (ii), a walk that hits u then v before jumping contributes equally to u's reputation and u's influence on v: $g(u, v) = \Pr[H(u) < H(v) < J] \leq \Pr[H(u) < J] = f(u)$. For parts (iii-iv), we split walks that hit v before jumping into those that hit u first and those that don't:

$$\begin{aligned} f(v) &= \Pr[H(v) < J] \\ &= \Pr[H(u) < H(v) \;\wedge\; H(v) < J] + \Pr[H(u) \geq H(v) \;\wedge\; H(v) < J] \\ &= g(u, v) + \Pr[H(u) \geq H(v) \;\wedge\; H(v) < J] \end{aligned}$$

The term $\Pr[H(u) \geq H(v) \;\wedge\; H(v) < J]$ is u-invariant, and equal to $f(v) - g(u, v)$ by the above, so repeating the calculation for G' gives $f'(v) = g'(u, v) + f(v) - g(u, v)$. By parts (i) and (ii), $0 \leq g'(u, v) \leq f'(u) = f(u)$, and substituting these bounds into the expression for $f'(v)$ gives (iii) and (iv).

Part (v) uses the observation that not too many nodes can be hit after u but before the first jump. Let $L = |\{v : H(u) < H(v) < J\}|$ be the number of all such nodes. Then,

$$E[L] = E\left[\sum_v I\{H(u) < H(v) < J\}\right] = \sum_v \Pr[H(u) < H(v) < J] = g(u).$$

But L cannot exceed $J - \min(H(u), J)$, so

$$\begin{aligned} g(u) = E[L] &\leq E[J] - E[\min(H(u), J)] \\ &= E[J](1 - \Pr[H(u) \geq J]) \\ &= E[J] \Pr[H(u) < J] \\ &= \frac{1}{\alpha} f(u). \end{aligned}$$

The equality in the second line is due to Lemma 1. □

4.1 Manipulating the Rankings

Theorem 2 quantifies how much a node u can manipulate reputation *values*, but often we are more concerned with how much u can manipulate the ranking, specifically, how far u can advance by manipulating outlinks only. The limited influence of u in parts (ii) and (v) of the theorem yield two immediate corollaries, proved in the appendix. If $f(u) < f(v)$, we say that u *meets* v if $f'(u) = f'(v)$, and u *surpasses* v if $f'(u) > f'(v)$.

Corollary 1. *Node u cannot surpass a node that is at least twice as reputable.*

Corollary 2. *Node u can meet or surpass at most* $\frac{1}{\alpha\gamma}$ *nodes that are more reputable than u by a factor of at least* $(1 + \gamma)$*.*

4.2 Reputation and Influence of Sets

We have discussed reputation and influence in terms of individual nodes for ease of exposition, but all of the definitions and results generalize when we consider the reputation and influence of sets of nodes. Let $U, W \subseteq V$, and recall that $H(W) = \min_{w \in W} H(w)$ is the hitting time of the set W. Then we define $f(W) = \Pr[H(W) < J]$ to be the reputation of W, we define $g(U, W) = \Pr[H(U) < H(W) < J]$ to be the influence of U on W, and we define $g(U) = \sum_{v \in V} g(U, \{v\})$ to be the total influence of U. With these definitions, exact analogues of Theorem 2 and its corollaries hold for any $U, W \subseteq V$, with essentially the same proofs. Note that U and W need not be disjoint, in which case it is possible that $H(U) = H(W)$. We omit further details.

4.3 Sybils

In online environments, it is often easy for a user to create new identities, called *sybils*, and use them to increase her own reputation, even without obtaining any new inlinks from non-sybils. A wide class of reputation systems is vulnerable to sybil attacks [8], and, in the extreme, hitting time can be heavily swayed as well. For example, if u places enough sybils so the random walk almost surely starts at a sybil, then adding links from each sybil to u ensures the walk hits u by the second step unless it jumps. In this fashion, u can achieve reputation almost $1 - \alpha$ and drive the reputation of all non-sybils to zero. We'll see that this is actually the *only* way that sybils can aid u, by gathering restart probability and funneling it towards u. So an application can limit the effect of sybils by limiting the restart probability granted to new nodes. In fact, applications like Personalized PageRank [30] and TrustRank [18] are already immune, since they place all of the restart probability on a fixed set of known or trusted nodes. Applications like web search that give equal restart probability to each node are more vulnerable, but in cases like the web the sheer number of nodes requires an attacker to place many sybils to have a substantial effect. This stands in stark contrast with PageRank, where one sybil is enough to employ the 2-cycle self-endorsement strategy and increase PageRank by several times [9].

To model the sybil attack, suppose $G' = (V \cup S, E')$ is obtained from G by a sybil attack launched by u. That is, the sybil nodes S are added, and links originating at u or inside S can be set arbitrarily. All other links must not change, with the exception that those originally pointing to u can be directed anywhere within $S \cup \{u\}$. Let q' be the new restart distribution, assuming that q' diverts probability to S but does not redistribute probability within V. Specifically, if $\rho = \sum_{s \in S} q'(s)$ is the probability alloted to sybils, we require that $q'(v) = (1 - \rho)q(v)$ for all $v \in V$. Let f' be the new reputation function.

Theorem 3. *Let $U = \{u\} \cup S$ be the nodes controlled by u, and let v be any other node in V. Then*

(i) $f'(u) \leq f'(U) = (1 - \rho)f(u) + \rho$,

(ii) $f'(v) \geq (1 - \rho)(f(v) - g(u, v))$,

(iii) $f'(v) \leq (1 - \rho)(f(v) + f(u) - g(u, v)) + \rho$.

Compared with Theorem 2, the only additional effect of sybils is to diminish all reputations by a factor of $(1 - \rho)$, and increase the reputation of certain target nodes by up to ρ.

Proof of Theorem 3. We split the attack into two steps, first observing how reputations change when the sybils are added but no links are changed, then applying Theorem 2 for the step when only links change. Let $G^+ = (V \cup S, E)$ be the intermediate graph with sybils but no new links, and let f^+ and g^+ be the reputation and influence functions in G^+. Assume the sybils have self-loops so the transition probabilities are well-defined. We can compute $f^+(U)$ by conditioning on whether X_0 lands in V (probability $1-\rho$) or S (probability ρ).

$$\begin{aligned} f^+(U) &= (1-\rho)\cdot \Pr_{G^+}\left[H(U) < J \mid X_0 \in V\right] + \rho \cdot \Pr_{G^+}\left[H(U) < J \mid X_0 \in S\right] \\ &= (1-\rho)\cdot \Pr_G\left[H(u) < J\right] + \rho \\ &= (1-\rho) f(u) + \rho. \end{aligned}$$

In the second step, $\Pr_{G^+}[H(U) < J \mid X_0 \in V] = \Pr_G[H(u) < J]$ because hitting U in G^+ is equivalent to hitting u in G; all edges outside U are unchanged, and all edges to U originally went to u. Also the conditional distribution of X_0 given $X_0 \in V$ is equal to q, by our assumption on q'. The term $\Pr_{G^+}[H(U) < J \mid X_0 \in S]$ is equal to one, since $X_0 \in S$ implies $H(U) = 0 < J$. A similar calculation gives

$$f^+(v) \;=\; (1-\rho) f(v) + \rho \cdot \Pr_{G^+}\left[H(v) < J \mid X_0 \in S\right] \;=\; (1-\rho) f(v).$$

The term $\Pr_{G^+}[H(v) < J \mid X_0 \in S]$ vanishes because S is disconnected, so a walk that starts in S cannot leave. Similarly, $g^+(U,v) = (1-\rho) g(u,v)$. Finally, we complete the sybil attack by modifying the edges originating in U, and apply Theorem 2 (the version generalized to deal with sets) to f^+ and g^+. Parts (i), (ii), and (iii) of this theorem are obtained by direct substitution using parts (i), (iii) and (iv) of Theorem 2. □

Theorem 3 can also be generalized to deal with sets.

5 Algorithms

To realize a reputation system based on hitting time, we require an algorithm to efficiently compute the reputation of all nodes. Theorem 1 suggests several possibilities. Recall that $\pi(v)$ is the PageRank of v. We have the following tools in our toolbox, which can be combined to obtain algorithms satisfying different requirements for accuracy and running time: (1) $E[R_\alpha(v)] = 1/\pi(v)$ can be computed efficiently for all nodes using a standard PageRank algorithm, (2) $\Pr[R(v) \geq J]$ can be estimated efficiently by Monte Carlo sampling, and (3) $\Pr[H(v) < J]$ can be estimated efficiently by Monte Carlo sampling *when this quantity is not too small*. The distinction between the last two items is important. We can get one sample of $\Pr[H(v) < J]$ or $\Pr[R(v) \geq J]$ by running a random walk until it first jumps, which takes about $1/\alpha$ steps. However $\Pr[H(v) < J]$ may be infinitesimal, requiring a huge number of independent samples to obtain a good estimate. On the other hand, $\Pr[R(v) \geq J]$ is bounded below by α since the walk has α chance of jumping in the very first step. If self-loops are disallowed, we obtain a better lower bound of $1-(1-\alpha)^2$, the probability the walk jumps in the first two steps . For now we focus on $\Pr[R(v) \geq J]$.

5.1 Algorithm for Hitting Time

In this section we describe an efficient Monte Carlo algorithm to simultaneously compute hitting time for all nodes. To obtain accuracy ϵ with probability at least $1-\delta$, the time required will be $O(\frac{\log(1/\delta)}{\epsilon^2\alpha^2}|V|)$ in addition to the time of one PageRank calculation. The algorithm is:

1. Compute π using a standard PageRank algorithm.[4] Then $E[R_\alpha(v)] = 1/\pi(v)$.

[4] PageRank algorithms are typically iterative and incur some error. Our analysis bounds the additional error incurred by our algorithm.

2. For each node v, run k random walks starting from v until the walk either returns to v or jumps. Let $y_v = \frac{1}{k} \cdot (\#$ of walks that jump before returning to $v)$.

3. Use y_v as an estimate for $\Pr[R(v) \geq J]$ in part (i) or (iii) of Theorem 1 to compute $E[H_\alpha(v)]$ or $\Pr[H(v) < J]$.

We suggested that a low value of $\Pr[R(v) \geq J]$ may be a good indicator of PageRank manipulation. To estimate this quantity for use in spam detection, steps 1 and 3 may be omitted. How many samples are needed to achieve high accuracy? Let $\mu = \Pr[R(v) \geq J]$ be the quantity estimated by y_v. We call y_v an (ϵ, δ)-approximation for μ if $\Pr[|y_v - \mu| \geq \epsilon\mu] \leq \delta$. A standard application of the Chernoff bound (see [28] p. 254) shows that y_v is an (ϵ, δ)-approximation if $k \geq (3\ln(2/\delta))/\epsilon^2\mu$. Using the fact that $\mu \geq \alpha$, it is sufficient that $k \geq (3\ln(2/\delta))/\epsilon^2\alpha$. Since each walk terminates in $\frac{1}{\alpha}$ steps in expectation, the total expected number of steps is no more than $\frac{3\ln(2/\delta)}{\epsilon^2\alpha^2}|V|$.

For massive graphs like the web that do not easily fit into main memory, it is not feasible to collect the samples in step 2 of the algorithm sequentially, because each walk requires random access to the edges, which is prohibitively expensive for data structures stored on disk. We describe a method from [12] to collect all samples simultaneously making efficient use of disk I/O.

Conceptually, the idea is to run all walks simultaneously and incrementally by placing tokens on the nodes recording the location of each random walk. Then we can advance all tokens by a single step in one pass through the entire graph. Assuming the adjacency list is stored on disk sorted by node, we store the tokens in a separate list sorted in the same order. Each token records the node where it originated to determine if it returns before jumping. Then in one pass through both lists, we load the neighbors of each node into memory and process each of its tokens, terminating the walk and updating y_u if appropriate, else choosing a random outgoing edge to follow and updating the token. Updated tokens are written to the end of a new unsorted token list, and after all tokens are processed, the new list is sorted on disk to be used in the next pass.

The number of passes is bounded by the walk that takes the longest to jump, which is not completely satisfactory, so in practice we can stop after a fixed number of steps t, knowing that the contribution of walks longer than t is nominal for large enough t, since $\Pr[R \geq J \wedge J > t] \leq \Pr[J > t] = (1-\alpha)^t$, which decays exponentially.

5.2 Finding Highly Reputable Nodes Quickly

Earlier we noted that $f(v) = \Pr[H(v) < J]$ is troublesome to estimate because it may be small. However, sampling $f(v)$ has the very nice property that a single random walk gives a sample of $f(v)$ for *all nodes*, and it may well be the case that we don't care about nodes for which $f(v)$ is small. For example suppose we want to label nodes as high-reputation if $f(v)$ exceeds some fixed threshold c, and low-reputation otherwise. It will be very difficult to classify nodes with reputation almost exactly c, so we relax the problem slightly and allow either classification for some small interval $[a, b]$ containing c. Let $\epsilon = \frac{b-a}{b}$. Then we have the following result, proved in the appendix.

Theorem 4. *Using $O(\log(1/\delta)/a\epsilon^2)$ Monte Carlo samples, we can label all nodes as high or low reputation, such that the expected number of mislabeled nodes is at most $\delta|V|$. With $O(\log(|V|/\delta)/a\epsilon^2)$ samples, we can classify all nodes correctly with probability at least $1-\delta$.*

The first result *does not depend on the size of the graph*, only on the threshold parameters a and ϵ. For graphs with highly skewed reputation distributions, a can be set to a high value to find the most reputable nodes very quickly. It is likely that real-world graphs will have skewed reputation distributions: for example, PageRank on the web graph has been observed to follow a power-law distribution [9]. Also, the thresholds need not be set in advance, so Theorem 4 can be used to give on-the-fly confidence intervals for the discovery of reputable nodes.

6 Additional Consequences of Theorem 1

Dissecting the quantities in Theorem 1 provides some additional insights into PageRank and hitting time. To start, the quantities $\Pr[R < J]$ and $\Pr[H < J]$ — which completely determine expected return time and hitting time — are polynomials in $1-\alpha$, with coefficients given by the distributions of R and H:

$$\Pr[H < J] = \sum_{t=0}^{\infty} \Pr[H = t] \Pr[J > t] = \sum_{t=0}^{\infty} \Pr[H = t] (1-\alpha)^t. \tag{2}$$

Similarly, $\Pr[R < J] = \sum_{t=0}^{\infty} \Pr[R = t] (1-\alpha)^t$. An immediate consequence is that $E[R_\alpha]$ and $E[H_\alpha]$ do not depend particularly on the link structure of G, only on the distributions induced on R and H. Then we can easily reason about PageRank and hitting time in certain graph structures.

Proposition 1. *Let S be the star graph where nodes $\{1, \ldots, k-1\}$ link only to k, and k links to at least one of $\{1, \ldots, k-1\}$. Then S minimizes both expected return time and expected hitting time of k over all graphs with k vertices and no self-loops.*

Proof. $H(k)$ is always 0 or 1, the minimum possible. Similarly, $R(k)$ is always 2, also the minimum possible since we prohibit self-loops. Then by (2), S simultaneously maximizes $\Pr[H(k) < J]$ and $\Pr[R(k) < J]$, so by Theorem 1, S minimizes $E[R_\alpha(k)]$ and $E[H_\alpha(k)]$. □

This reasoning holds even if S is a subgraph of G and we consider all possible arrangements of the links within S. We need the additional fact, proved in the previous section, that $H(S)$ is independent of the links inside S. Then arranging S as a star ensures that $H(k)$ is either $H(S)$ or $H(S) + 1$, always the minimum possible, and $R(k)$ is still 2. This gives a very simple proof of technical results from [5] and [16] describing the optimal way for a group of pages to boost the PageRank of a single target.

Another consequence of (2) is that pre-computing $\Pr[H = t]$ for small values of t gives a good estimate for $\Pr[H < J]$ for all values of α that are not too small, just by taking the low-order terms of the polynomial. The contribution of all terms of degree ℓ or more is at most $(1-\alpha)^\ell$, so taking the first $\ell - 1$ terms for $\ell \geq \log\epsilon / \log(1-\alpha_0)$ gives error at most ϵ for all $\alpha \geq \alpha_0$. An analogous result holds for $\Pr[R < J]$.

7 Conclusion

We have explored the use of hitting time as a reputation system. Hitting time is resistant to manipulation: an individual cannot alter her own reputation, and can alter another's reputation by only a limited amount. To quantify this amount, we developed a definition of influence, and showed that influence is limited by reputation. We presented algorithms to efficiently compute hitting time on large graphs, making it viable for real world applications. Our work also has consequences for spam detection and PageRank computation.

References

[1] David Aldous and James Fill. *Reversible Markov Chains and Random Walks on Graphs*. Monograph in Preparation, `http://www.stat.berkeley.edu/users/aldous/RWG/book.html`.

[2] Alon Altman and Moshe Tennenholtz. Incentive compatible ranking systems. In *Proceedings of the 2007 International Conference on Autonomous Agents and Multiagent Systems*, to appear.

[3] Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using PageRank vectors. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 475–486, Washington, DC, USA, 2006. IEEE Computer Society.

[4] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova. Monte carlo methods in PageRank computation: When one iteration is sufficient. Memorandum 1754, University of Twente, The Netherlands, 2005.

[5] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside PageRank. *ACM Trans. Inter. Tech.*, 5(1):92–128, 2005.

[6] Alan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas. Finding authorities and hubs from link structures on the world wide web. In *Proceedings of the Tenth International World Wide Web Conference*, pages 415–429, 2001.

[7] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[8] Alice Cheng and Eric Friedman. Sybilproof reputation mechanisms. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 128–132, New York, NY, USA, 2005. ACM Press.

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

[10] Steve Chien, Cynthia Dwork, Ravi Kumar, Daniel R. Simon, and D. Sivakumar. Link evolution: Analysis and algorithms. *Internet Mathematics*, 1(3):277–304, 2004.

[11] John Douceur. The sybil attack. In *Proceedings of the IPTPS02 Workshop*, 2002.

[12] Dániel Fogaras and Balázs Rácz. Towards fully personalizing PageRank. In *Proceedings of the 3rd Workshop on Algorithms and Models for the Web-Graph (WAW2004), in conjunction with FOCS 2004*, 2004.

[13] Eric Friedman, Paul Resnick, and Rahul Sami. Manipulation-resistant reputation systems. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, to appear.

[14] Krishna Gade and Amit Prakash. Using transient probability distributions of random walk to estimate spam resistant authority scores. Unpublished manuscript, 2007.

[15] Zoltán Gyöngyi, Pavel Berkhin, Hector Garcia-Molina, and Jan Pedersen. Link spam detection based on mass estimation. In *Proceedings of the 32nd International Conference on Very Large Databases*. ACM, 2006.

[16] Zoltán Gyöngyi and Hector Garcia-Molina. Link spam alliances. In *Proceedings of the 31st International Conference on Very Large Databases*, pages 517–528. ACM, 2005.

[17] Zoltán Gyöngyi and Hector Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[18] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Databases*, pages 576–587. Morgan Kaufmann, 2004.

[19] Philip Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Trans. Model. Comput. Simul.*, 5(1):43–85, 1995.

[20] Glen Jeh and Jennifer Widom. SimRank: A measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[21] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *WWW '03: Proceedings of the Twelfth International World Wide Web Conference*, pages 271–279, New York, NY, USA, 2003. ACM Press.

[22] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM Press.

[23] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[24] Amy N. Langville and Carl D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380, 2004.

[25] R. Lempel and S. Moran. Rank-stability and rank-similarity of link-based web ranking algorithms in authority-connected graphs. *Inf. Retr.*, 8(2):245–264, 2005.

[26] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM)*, 2003.

[27] Kahn Mason. *Detecting Colluders in PageRank - Finding Slow Mixing States in a Markov Chain*. PhD thesis, Stanford University, 2005.

[28] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA, 2005.

[29] Andrew Y. Ng, Alice X. Zheng, and Michael I. Jordan. Stable algorithms for link analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference*. ACM, 2001.

[30] 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.

[31] Hui Zhang, Ashish Goel, Ramesh Govindian, Kahn Mason, and Benjamin Van Roy. Making eigenvector-based reputation systems robust to collusion. In *Proceedings of the 3rd Workshop on Algorithms and Models for the Web-Graph (WAW2004), in conjunction with FOCS 2004*, 2004.

A Proof of Lemma 1

Proof of Lemma 1. Recall that J is the time of the first success in a sequence of independent trials that succeed with probability α, so $\Pr[J > t] = (1-\alpha)^t$, and $\Pr[J \le t] = 1-(1-\alpha)^t$. For fixed x, we have

$$\begin{aligned}
E[\min(X,J)] &= \sum_{t=0}^{\infty} \Pr[\min(X,J) > t] \\
&= \sum_{t=0}^{\infty}\sum_{x=0}^{\infty} \Pr[X = x]\Pr[\min(X,J) > t \mid X = x] \\
&= \sum_{x=0}^{\infty} \Pr[X = x]\sum_{t=0}^{\infty} \Pr[\min(x,J) > t] && \text{(using independence)} \\
&= \sum_{x=0}^{\infty} \Pr[X = x]\sum_{t=0}^{x-1} \Pr[J > t] \\
&= \sum_{x=0}^{\infty} \Pr[X = x]\sum_{t=0}^{x-1} (1-\alpha)^t \\
&= \sum_{x=0}^{\infty} \Pr[X = x]\,\frac{1-(1-\alpha)^x}{1-(1-\alpha)} \\
&= \sum_{x=0}^{\infty} \Pr[X = x]\,\frac{\Pr[J \le x]}{\alpha} \\
&= \frac{1}{\alpha}\Pr[X \ge J]
\end{aligned}$$

□

Note that if J is any random variable on the positive integers, then Lemma 1 uniquely characterizes J as a geometric random variable. To see this, observe that $\min(x,J)-\min(x-1,J)$ is an indicator variable for the event that $J \ge x$. Then for fixed x,

$$\begin{aligned}
\Pr[J \ge x] &= E[\min(x,J)] - E[\min(x-1,J)] \\
&= \frac{1}{\alpha}(\Pr[x \ge J] - \Pr[x-1 \ge J]) && \text{(by Lemma 1)} \\
&= \frac{1}{\alpha}\Pr[J = x].
\end{aligned}$$

Applying this for all $x \ge 0$ and using the boundary condition $\Pr[J \ge 1] = 1$, we can set up a recurrence that uniquely defines the distribution of J.

B Proof of Lemma 2

Proof. It is enough to show that $\Pr_G[A \wedge H(u) = t] = \Pr_{G'}[A \wedge H(u) = t]$ for all $t \geq 0$. Our conditions on A are vague about the case $H(u) = \infty$; to be definite, assume that $\Pr_G[A \wedge H(u) = \infty] = \Pr_{G'}[A \wedge H(u) = \infty] = 0$. For finite t, let $W_{u,t}$ be the set of all walks that first hit u at step t. Specifically, $W_{u,t} = \{w_0 \ldots w_t : w_t = u,\ w_i \neq u \text{ for } i < t\}$. For $w = w_0 \ldots w_t$, let $\Pr[w]$ be shorthand for the probability of the walk w:

$$\Pr[w] = \Pr[X_0 = w_0] \Pr[X_1 = w_1 \mid X_0 = w_0] \ldots \Pr[X_t = w_t \mid X_{t-1} = w_{t-1}].$$

Then for $w \in W_{u,t}$, the transition probabilities in the expression above are independent of u's outlinks, so $\Pr_G[w] = \Pr_{G'}[w]$. Finally, since A is determined by time $H(u)$, there is a function $I_A : W_{u,t} \to \{0, 1\}$ that indicates the occurrence or non-occurrence of A for each $w \in W_{u,t}$. Putting it all together,

$$\begin{aligned}
\Pr_G[A \wedge H(u) = t] &= \Pr_G[H(u) = t] \Pr_G[A \mid H(u) = t] \\
&= \sum_{w \in W_{u,t}} \Pr_G[w]\, I_A(w) \\
&= \sum_{w \in W_{u,t}} \Pr_{G'}[w]\, I_A(w) \\
&= \Pr_{G'}[A \wedge H(u) = t]
\end{aligned}$$

□

C Proof of Corollaries to Theorem 2

Proof of Corollary 1. Suppose $f(v) \geq 2f(u)$, then $f'(v) \geq f(v) - g(u, v) \geq f(v) - f(u) \geq 2f(u) - f(u) = f(u) = f'(u)$. □

Proof of Corollary 2. Let $A = \{v : f(v) \geq (1 + \gamma)f(u) \ \wedge \ f'(v) \leq f'(u)\}$ be the set of all such nodes. Then

$$\begin{aligned}
\sum_{v \in A} f(v) &\geq |A|(1 + \gamma)f(u), \\
\sum_{v \in A} f'(v) &\leq |A|f'(u) = |A|f(u),
\end{aligned}$$

so $\sum_{v \in A} f(v) - f'(v) \geq \gamma |A| f(u)$. But by part (iii) of Theorem 2, $f(v) - f'(v) \leq g(u, v)$, so

$$\gamma |A| f(u) \leq \sum_{v \in A} f(v) - f'(v) \leq \sum_{v \in A} g(u, v) \leq g(u) \leq \frac{1}{\alpha} f(u),$$

hence $|A| \leq \frac{1}{\alpha\gamma}$. □

D Proof of Theorem 4

Proof. Suppose we perform k walks, letting $z_v = \frac{1}{k} \cdot (\#\text{ of walks that hit } v \text{ before jumping})$ be the estimate for $f(v)$. The symmetric Chernoff bounds (see, e.g., [28] p. 64) give:

$$\begin{aligned}
\Pr[z_v \geq (1 + \epsilon)f(v)] &\leq \exp(-kf(v)\epsilon^2/3) \\
\Pr[z_v \leq (1 - \epsilon)f(v)] &\leq \exp(-kf(v)\epsilon^2/3)
\end{aligned}$$

Recall that $\epsilon = \frac{b-a}{b}$, so $a = (1-\epsilon)b$, and $b > (1+\epsilon)a$. The probability that a low-reputation node is misclassified is

$$\begin{aligned}
\Pr\left[z_v \geq b \mid f(v) \leq a\right] &\leq \Pr\left[z_v \geq b \mid f(v) = a\right] \\
&\leq \Pr\left[z_v \geq (1+\epsilon)f(v) \mid f(v) = a\right] \\
&\leq \exp(-ka\epsilon^2/3).
\end{aligned}$$

The probability that a high-reputation node is misclassified is

$$\begin{aligned}
\Pr\left[z_v \leq a \mid f(v) \geq b\right] &= \Pr\left[z_v \leq (1-\epsilon)b \mid f(v) \geq b\right] \\
&\leq \Pr\left[z_v \leq (1-\epsilon)f(v) \mid f(v) \geq b\right] \\
&\leq \exp(-kf(v)\epsilon^2/3) \\
&\leq \exp(-ka\epsilon^2/3).
\end{aligned}$$

Choosing $k \geq \frac{3\ln(1/\delta)}{a\epsilon^2}$ ensures that each node is misclassified with probability at most δ, so the expected number of misclassified nodes is at most $\delta|V|$. Furthermore, by the union bound, the probability that any node is misclassified is at most $|V|\exp(-ka\epsilon^2/3)$, so choosing $k \geq \frac{3\ln(|V|/\delta)}{a\epsilon^2}$ ensures that all nodes are classified correctly with probability at least $1-\delta$. $\square$