Internet Mathematics Vol. 5, No. 1–2: 71–90

Manipulation-Resistant Reputations Using Hitting Time

John Hopcroft and Daniel Sheldon

Abstract. Popular reputation systems for linked networks can be manipulated by spammers who strategically place links. In PageRank [Brin and Page 98], pages endorse others by placing links, and the global link structure is analyzed to determine the reputation of each page. Though this is meant to be a global measure, page v can boost its own PageRank considerably using a simple self-endorsement strategy: placing outlinks to form 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 does not depend on v's outlinks. We develop a reputation system based on hitting time and show that it 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.

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, not all participants are honorable—e.g., spammers will do their best to manipulate a search engine's rankings. A natural requirement for a reputation system is that individuals should not be able to improve their own reputation using simple self-endorsement strategies, such as participating in short cycles to boost PageRank. Since PageRank enjoys many nice properties, it is instructive to see where things go wrong.

1542-7951/08 $0.50 per page

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 at random. This ensures that π exists and is efficient to compute. Then a well-known fact about Markov chains [Aldous and Fill 09] 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 as a ranking system, one can easily manipulate return time by changing *only outlinks*. Intuitively, a node v should link only to nodes from which a random walk will return to v quickly (in expectation). 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 [Cheng and Friedman 06]. 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 a random node). This paper develops a reputation system based on hitting time that is provably resistant to manipulation. Our main contributions are these:

- In Theorem 3.1, we develop a precise relationship between expected return time and expected hitting time in a random walk with restart, and show that the expected hitting time of v is equal to $(1 - p)/\alpha p$, where p is the probability that v is reached before the first restart. We will adopt p as our measure of the reputation of v.

- We prove that the resulting reputation system resists manipulation, using a natural definition of influence. 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.

- We present an efficient algorithm to simultaneously compute hitting time for all nodes in a large graph. 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]

The rest of the paper is structured as follows. In Section 2 we discuss related work. In Section 3 we present Theorem 3.1, 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.

2. Related Work

Since PageRank [Brin and Page 98] was introduced, it has been adapted to a variety of applications, including personalized web search [Page et al. 98], web spam detection [Gyöngyi et al. 04], and trust systems in peer-to-peer networks [Kamvar et al. 03]. Each of these uses the same general formulation and our work applies to all of them.

Much work has focused on the PageRank system itself, studying computation methods, convergence properties, stability and sensitivity, and, of course, implementation techniques. See [Langville and Meyer 04] for a survey of this wide body of work. Computationally, the Monte Carlo methods in [Fogaras and Rácz 04] and [Avrachenkov et al. 05] are similar to our algorithms for hitting time. They use a probabilistic formulation of PageRank in terms of a *short* random walk that permits efficient sampling. In particular, we will use the same idea as [Fogaras and Rácz 04] to efficiently implement many random walks simultaneously in a massive graph, without requiring random access.

Recent works have addressed the manipulability of PageRank: how can a group of selfish nodes place outlinks to optimize their PageRank, and how can we detect such nodes [Gyöngyi and Garcia-Molina 05b, Gyöngyi and Garcia-Molina 05a, Gyöngyi et al. 06, Zhang et al. 04, Mason 05, Bianchini et al. 05, Cheng and Friedman 06]? In particular, [Gyöngyi and Garcia-Molina 05a, Bianchini et al. 05, Cheng and Friedman 06] all describe the manipulation strategy mentioned in the introduction.

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

For a more general treatment of reputation systems in the presence of strategic agents, see [Friedman et al. 09] for a nice overview with some specific results from the literature. Cheng and Friedman prove an impossibility result [Cheng and Friedman 05] that relates to our work—a wide class of reputation systems (including ours) cannot be resistant to a particular attack called the *sybil attack* [Douceur 02]. However, their definition of resistance is very strong, requiring that no node can improve its ranking using a sybil attack; our results can be viewed as positive results under a relaxation of this requirement by limiting the damage caused by a sybil attack. We will discuss sybils in Section 4.3.

Hitting time is a classical quantity of interest in Markov chains. See [Aldous and Fill 09, Chapter 2] 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. In a context similar to ours, hitting time was used as a measure of proximity between nodes to predict link formation in a social network [Liben-Nowell and Kleinberg 03]; also, the node similarity measure in [Jeh and Widom 02] can be formulated in terms of hitting time.

Finally, the relationship between hitting time and return time in a random walk with restart is related to regenerative stochastic processes. In fact, Theorem 3.1 can be derived as a special case of a general result about such processes. See [Heidelberger 95, equation (15)] and the references therein for details.

After the conference version of this paper [Hopcroft and Sheldon 07] was published, we discovered the paper [Avrachenkov and Litvak 06], which studies the effect of new links on PageRank. In particular, the authors note (in Proposition 2.1) one of the conclusions of Theorem 3.1: that the PageRank of page v can be written as a product of two terms, where only the first term depends on the outlinks of v. The second term in their formulation—which is independent of v's outlinks—is exactly our measure of the reputation of v.

3. Characterizing Hitting Time

This section paves the way toward a reputation system based on hitting time by stating and proving Theorem 3.1. Part (i) of the theorem relates expected hitting time to expected return time—the two are essentially the same *except* for nodes where the random walk is likely to return before jumping, the sign of a known manipulation strategy. Part (ii) proves that the expected 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 $\Pr[X_t = v \mid X_{t-1} = u] = 1/\operatorname{outdegree}(u)$ if $(u, v) \in E$, and zero otherwise. Here, we require $\operatorname{outdegree}(u) > 0$.[2] Now, suppose the walk is modified to restart with probability α at each step, meaning that the next node is chosen from the starting distribution (henceforth, *restart distribution*) instead of following a link. The new transition probabilities are

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

We call this the *α-random walk* on G, and we parameterize quantities of interest by the restart probability α. A typical setting is $\alpha = 0.15$, so a jump occurs every $1/0.15 \approx 7$ 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 3.1

Before stating Theorem 3.1, 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 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 we can model the α-random walk as follows: (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:

$$R_\alpha = \begin{cases} R & \text{if } R < J, \\ J + H'_\alpha & \text{otherwise,} \end{cases} \qquad H_\alpha = \begin{cases} H & \text{if } H < J, \\ J + H'_\alpha & \text{otherwise.} \end{cases} \tag{3.1}$$

[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.

Here H'_α is an independent copy of H_α. It is convenient to abstract from our specific setting and state Theorem 3.1 about general random variables of this form.

Theorem 3.1. *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* (3.1)*. 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 in which 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 3.2. We stress that the expected 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 our numeric measure of reputation.

[3]We mentioned that Theorem 3.1 can be derived from a result about regenerative stochastic processes [Heidelberger 95]. In fact, Theorem 3.1 captures most of the generality; to write recurrences as in (3.1), the process need not be Markovian; it is necessary only that the process following a restart be a replica of the original. The only nongeneral assumption made is that J is a geometric random variable; this simplifies the conclusions.

Lemma 3.2. *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 3.2 is proved in the appendix (Section 7).

Proof of Theorem 3.1. 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 3.2,

$$\begin{aligned} E[R_\alpha] &= E[\min(R,J)] + \Pr[R \geq J]\,E[H'_\alpha] \\ &= \frac{1}{\alpha}\Pr[R \geq J] + \Pr[R \geq J]\,E[H_\alpha] \\ &= \Pr[R \geq J]\left(\frac{1}{\alpha} + E[H_\alpha]\right). \end{aligned}$$

This proves part (i). The proof of (ii) uses part (i), taking advantage of the more general statement of the theorem. Note that H_α and R_α are defined similarly in (3.1), so substituting H for R in part (i), we get

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

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 node u cannot improve its 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 be able to damage v's reputation only 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 3.1, either quantity determines the other—they are roughly inversely proportional—and $\Pr[H < J]$ is convenient for reasoning about manipulation.

Definition 4.1. Let $\operatorname{rep}(v) = \Pr[H(v) < J]$ be the *reputation* of v.

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

Definition 4.2. Let $\mathrm{infl}(u, v) = \Pr[H(u) < H(v) < J]$ be the *influence of u on v.*

Definition 4.3. Let $\mathrm{infl}(u) = \sum_v \mathrm{infl}(u, v)$ be the *total influence of u.*

When the graph G is not clear from context, we write these quantities as $\Pr_G[\cdot]$, $\mathrm{rep}_G(\cdot)$, and $\mathrm{infl}_G(\cdot, \cdot)$ to be clear. To quantify what can change when u manipulates outlinks, let $\mathcal{N}_u(G)$ be the set of all graphs obtained from G by the addition or deletion of edges originating at u. 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 4.4. Fix a graph G and node u. We say that an event A is *u-invariant* if $\Pr_G[A] = \Pr_{G'}[A]$ for all $G' \in \mathcal{N}_u(G)$. If A is u-invariant, we also say that the quantity $\Pr[A]$ is u-invariant.

Lemma 4.5. *An event A is u-invariant if the occurrence or nonoccurrence of A is determined by time $H(u)$.*

Lemma 4.5 is proved in the appendix. With the definitions in place, we can quantify how much u can manipulate reputations.

Theorem 4.6. *For any graph $G = (V, E)$ and $u, v \in V$,*

(i) $\mathrm{infl}(u, u) = 0$,

(ii) $\mathrm{infl}(u, v) \geq 0$,

(iii) $\mathrm{infl}(u, v) \leq \mathrm{rep}(u)$,

(iv) $\mathrm{infl}(u) \leq \frac{1}{\alpha}\,\mathrm{rep}(u)$.

Let $G' \in \mathcal{N}_u(G)$. Then

(v) $\mathrm{rep}_{G'}(v) = \mathrm{rep}_G(v) + \mathrm{infl}_{G'}(u, v) - \mathrm{infl}_G(u, v)$.

Parts (i)–(iv) bound the influence of u in terms of its reputation. Part (v) states that when u modifies outlinks, the change in v's reputation is equal to the change in u's influence on v. Substituting parts (i)–(iii) into part (v) yields some simple but useful corollaries.

Corollary 4.7. *Let $G' \in \mathcal{N}_u(G)$. Then*

(i) $\text{rep}_{G'}(u) = \text{rep}_G(u)$,

(ii) $\text{rep}_{G'}(v) \geq \text{rep}_G(v) - \text{infl}_G(u, v)$,

(iii) $\text{rep}_{G'}(v) \leq \text{rep}_G(v) - \text{infl}_G(u, v) + \text{rep}_G(u)$.

No matter what actions u takes, it cannot alter its own reputation (part (i)). Nor can u damage the portion of v's reputation *not* due to u's influence (part (ii)). On the other hand, u may boost its influence on v, but its final influence cannot exceed its reputation (part (iii)).

Proof of Theorem 4.6. For the most part, the assertions of the theorem are simple consequences of the definitions. Parts (i) and (ii) are trivial:

$$\begin{aligned}\text{infl}(u, u) &= \Pr[H(u) < H(u) < J] = 0,\\ \text{infl}(u, v) &= \Pr[H(u) < H(v) < J] \geq 0.\end{aligned}$$

For part (iii), a walk that hits u then v before jumping contributes equally to u's reputation and u's influence on v:

$$\text{infl}(u, v) = \Pr[H(u) < H(v) < J] \leq \Pr[H(u) < J] = \text{rep}(u).$$

Part (iv) 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

$$E[L] = E\Big[\sum_v I\{H(u) < H(v) < J\}\Big] = \sum_v \Pr[H(u) < H(v) < J] = \text{infl}(u).$$

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

$$\begin{aligned}\text{infl}(u) = E[L] &\leq E[J] - E[\min(H(u), J)]\\ &= E[J](1 - \Pr[H(u) \geq J]) \qquad \text{(by Lemma 3.2)}\\ &= E[J]\Pr[H(u) < J]\\ &= \frac{1}{\alpha}\,\text{rep}(u).\end{aligned}$$

For part (v), we split walks that hit v before jumping into those that hit u first and those that don't:

$$\begin{aligned}\mathrm{rep}_G(v) &= \Pr_G[H(v) < J] \\ &= \Pr_G[H(u) < H(v),\, H(v) < J] + \Pr_G[H(u) \geq H(v),\, H(v) < J] \\ &= \mathrm{infl}_G(u,v) + \Pr_G[H(u) \geq H(v),\, H(v) < J].\end{aligned}$$

The event $[H(u) \geq H(v),\, H(v) < J]$ is determined by time $H(u)$, and hence it is u-invariant. By the above, $\Pr[H(u) \geq H(v),\, H(v) < J]$ is equal to $\mathrm{rep}_G(v) - \mathrm{infl}_G(u,v)$, and repeating the calculation for G' gives $\mathrm{rep}_{G'}(v) = \mathrm{infl}_{G'}(u,v) + \mathrm{rep}_G(v) - \mathrm{infl}_G(u,v)$. □

4.1. Manipulating the Rankings

The previous results quantify how much 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 following two corollaries follow easily. Suppose $\mathrm{rep}_G(u) < \mathrm{rep}_G(v)$ and u manipulates outlinks to produce $G' \in \mathcal{N}_u(G)$. We say that u *meets* v if $\mathrm{rep}_{G'}(u) = \mathrm{rep}_{G'}(v)$, and u *surpasses* v if $\mathrm{rep}_{G'}(u) > \mathrm{rep}_{G'}(v)$.

Corollary 4.8. *Node u cannot surpass a node that is at least twice as reputable.*

Corollary 4.9. *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 $(1+\gamma)$.*

Proof of Corollary 4.8. Suppose $\mathrm{rep}_G(v) \geq 2 \cdot \mathrm{rep}_G(u)$. Then

$$\begin{aligned}\mathrm{rep}_{G'}(v) &\geq \mathrm{rep}_G(v) - \mathrm{infl}_G(u,v) \geq \mathrm{rep}_G(v) - \mathrm{rep}_G(u) \\ &\geq 2 \cdot \mathrm{rep}_G(u) - \mathrm{rep}_G(u) \\ &= \mathrm{rep}_G(u) = \mathrm{rep}_{G'}(u).\end{aligned}$$

This completes the proof. □

Proof of Corollary 4.9. Let $A = \{v : \mathrm{rep}_G(v) \geq (1+\gamma)\,\mathrm{rep}_G(u),\ \mathrm{rep}_{G'}(v) \leq \mathrm{rep}_{G'}(u)\}$ be the set of all nodes with reputation at least $(1+\gamma)$ times the reputation of u that are met or surpassed by u. Then

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

so

$$\sum_{v \in A} (\mathrm{rep}_G(v) - \mathrm{rep}_{G'}(v)) \geq \gamma |A| \, \mathrm{rep}_G(u).$$

But by Corollary 4.7, $\mathrm{rep}_G(v) - \mathrm{rep}_{G'}(v) \leq \mathrm{infl}_G(u, v)$, so

$$\gamma |A| \, \mathrm{rep}_G(u) \leq \sum_{v \in A} (\mathrm{rep}_G(v) - \mathrm{rep}_{G'}(v)) \leq \sum_{v \in A} \mathrm{infl}_G(u, v) \leq \mathrm{infl}_G(u) \leq \frac{1}{\alpha} \mathrm{rep}_G(u),$$

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

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 $\mathrm{rep}(W) = \Pr[H(W) < J]$ to be the reputation of W, we define $\mathrm{infl}(U, W) = \Pr[H(U) < H(W) < J]$ to be the influence of U on W, and we define $\mathrm{infl}(U) = \sum_{v \in V} \mathrm{infl}(U, \{v\})$ to be the total influence of U. With these definitions, exact analogues of Theorem 4.6 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. On the web, a spammer might control a large number of sites, arranging them to boost the PageRank of a given target page; such a configuration is called a *spam farm* [Gyöngyi and Garcia-Molina 05b]. In general, a wide class of reputation systems is vulnerable to sybil attacks [Cheng and Friedman 05], 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 that 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 shall see that this is actually the *only* way that sybils can aid u, by gathering restart probability and funneling it toward u. So an application can limit the effect of sybils by limiting the restart probability granted to new nodes. In fact, applications of hitting time analogous to Personalized PageRank [Page et al. 98] and TrustRank [Gyöngyi et al. 04] are already immune, since they place all of the restart probability on a fixed set of known or trusted nodes.

Applications such as 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 to PageRank, where one sybil is enough to employ the 2-cycle self-endorsement strategy and increase PageRank by several times [Cheng and Friedman 06].

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 restart probability allotted to sybils, we require that $q'(v) = (1-\rho)q(v)$ for all $v \in V$.

Theorem 4.10. *Let $U = \{u\} \cup S$ be the nodes controlled by the attacker u, and let v be any other node in V. Then*

(i) $\operatorname{rep}_{G'}(u) \leq \operatorname{rep}_{G'}(U) = (1-\rho)\operatorname{rep}_G(u) + \rho,$

(ii) $\operatorname{rep}_{G'}(v) \geq (1-\rho)(\operatorname{rep}_G(v) - \operatorname{infl}_G(u,v)),$

(iii) $\operatorname{rep}_{G'}(v) \leq (1-\rho)(\operatorname{rep}_G(v) - \operatorname{infl}_G(u,v) + \operatorname{rep}_G(u)) + \rho.$

Compared with Corollary 4.7, 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 4.10. 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 4.6 for the step at which only links change. Let G^+ be the intermediate graph where we add the sybils but do not change links. Assume that the sybils have self-loops so the transition probabilities are well defined. We can compute $\operatorname{rep}_{G^+}(U)$ by conditioning on whether $X_0 \in V$ or $X_0 \in S$, recalling that $\Pr[X_0 \in S] = \rho$:

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

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

$$\mathrm{rep}_{G^+}(v) \;=\; (1-\rho)\,\mathrm{rep}_G(v) + \rho \cdot \Pr_{G^+}[H(v) < J \mid X_0 \in S] \;=\; (1-\rho)\,\mathrm{rep}_G(v).$$

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. Another similar calculation gives $\mathrm{infl}_{G^+}(U, v) = (1-\rho)\,\mathrm{infl}_G(u, v)$. Finally, we complete the sybil attack, obtaining G' from G^+ by making arbitrary changes to edges originating in U, and apply Corollary 4.7 (the version generalized to deal with sets) to G^+. Parts (i)–(iii) of this theorem are obtained by direct substitution into their counterparts from Corollary 4.7. $\square$

Theorem 4.10 can also be generalized to deal with sets.

5. Computing Hitting Time

To realize a reputation system based on hitting time, we require an algorithm to efficiently compute the reputation of all nodes. Theorem 3.1 suggests several possibilities. Recall that $\pi(v)$ is the PageRank of v. Then $E[R_\alpha(v)] = 1/\pi(v)$ can be computed efficiently for all nodes using a standard PageRank algorithm, and the quantity $\Pr[R(v) \geq J]$ can be estimated efficiently by Monte Carlo sampling. Combining these two quantities using Theorem 3.1 yields $E[H_\alpha(v)]$.

It is tempting to estimate the reputation $\Pr[H(v) < J]$ directly using Monte Carlo sampling. However, there is an important distinction between the quantities $\Pr[R(v) \geq J]$ and $\Pr[H(v) < J]$. We can get one sample of either 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 at least α, since the walk has probability α 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 that the walk jumps in the first two steps. For this reason we focus on $\Pr[R(v) \geq J]$.

5.1. A Monte Carlo Algorithm

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 $O\big(\frac{\log(1/\delta)}{\epsilon^2\alpha^2}|V|\big)$ in addition to the time of one PageRank calculation. The algorithm is as follows:

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

2. For each node v, run k random walks starting from v until the walk either returns to v or jumps. Let

$$y_v = \frac{1}{k} \cdot (\# \text{ of walks that jump before returning to } v).$$

3. Use y_v as an estimate for $\Pr[R(v) \geq J]$ in part (i) or (iii) of Theorem 3.1 to compute $E[H_\alpha(v)]$ or $\Pr[H(v) < J]$.

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 [Mitzenmacher and Upfal 05, 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 [Fogaras and Rácz 04] 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 that 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 whether 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_v if appropriate, and otherwise 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.

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

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 $\Pr[R \geq J,\ J > t] \leq \Pr[J > t] = (1-\alpha)^t$, which decays exponentially.

5.2. Finding Highly Reputable Nodes Quickly

We noted that estimating $\text{rep}(v) = \Pr[H(v) < J]$ directly by Monte Carlo sampling is troublesome in the case that this probability is very small. However, a benefit of this approach is that a single random walk gives a sample of $\text{rep}(v)$ for *all nodes*, and in some situations we may not care about nodes of low reputation. For example, suppose we want to find all nodes with reputation exceeding some fixed threshold c. A simple approach is to run many random walks and return all nodes for which the empirical estimate of $\text{rep}(v)$ exceeds c. We will show that the requisite number of walks depends very modestly on the size of the graph, and in some cases is independent of the size of the graph.

Specifically, we shall treat this as a classification problem, to label v as high reputation if $\text{rep}(v) \geq c$, 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.

Theorem 5.1. *Using $O(\log(1/\delta)/a\epsilon^2)$ Monte Carlo samples, we can label all nodes as high or low reputation in such a way 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 [Cheng and Friedman 06]. Also, the thresholds need not be set in advance, so Theorem 5.1 can be used to give on-the-fly confidence intervals for the discovery of reputable nodes.

Proof of Theorem 5.1. Let $\mu = \text{rep}(v)$, and suppose we perform k walks, letting

$$z_v = \frac{1}{k} \cdot (\text{\# of walks that hit } v \text{ before jumping})$$

be the empirical estimate for μ. The symmetric Chernoff bounds (see, e.g., [Mitzenmacher and Upfal 05, p. 64]) give

$$\begin{aligned}\Pr\left[z_v \geq (1+\epsilon)\mu\right] &\leq \exp(-k\mu\epsilon^2/3),\\ \Pr\left[z_v \leq (1-\epsilon)\mu\right] &\leq \exp(-k\mu\epsilon^2/3).\end{aligned}$$

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

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

The probability that a high-reputation node is misclassified is

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

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 $1-\delta$. □

6. Conclusion

As online environments become ubiquitous, it is vital to understand the interplay between participants, who are potentially selfish, and the tools we use to understand and navigate the environment. This paper addresses this issue from the perspective of a search engine: how can one measure link-based reputations in a way that can't easily be manipulated?

We have explored the use of hitting time to measure reputations, overcoming a vulnerability in PageRank that allows selfish pages to boost their reputation by using short cycles to trap the random walk. Theorem 3.1 shows that expected hitting time and PageRank are really quite similar—for a given page, expected hitting time can be obtained from PageRank via a multiplicative penalty term that captures the extent to which the page participates in short cycles.

Furthermore, hitting time is provably robust to manipulation. We show that v cannot boost its reputation by manipulating outlinks, nor can it do too much damage to the reputation of other pages; hence, v cannot advance too far in the ranking. Finally, using hitting time, we can limit the effect of sybils simply by limiting the restart probability granted to them; this in contrast to PageRank, where significant gains can be made by using a single sybil to form a short cycle.

Hitting time is more costly to compute than PageRank. However, we present a Monte Carlo algorithm that is suitable for the computation of hitting time for all nodes in a very large graph. We also describe a sampling technique to find the most important nodes very quickly—in some cases, with running time independent of the size of the graph.

7. Appendix: Additional Proofs

Proof of Lemma 3.2. 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 $\Pr[J \le t] = 1 - (1-\alpha)^t$. We have

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

which completes the proof. □

Proof of Lemma 4.5. Let $G' \in \mathcal{N}_u(G)$. It is enough to show that

$$\mathrm{Pr}_G\left[A \cap [H(u) = t]\right] = \mathrm{Pr}_{G'}\left[A \cap [H(u) = t]\right]$$

for all $t \geq 0$. 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 $\mathrm{Pr}\left[w\right]$ be shorthand for the probability of the walk w:

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

Then for $w \in W_{u,t}$, the transition probabilities in the expression above are independent of u's outlinks, so $\mathrm{Pr}_G\left[w\right] = \mathrm{Pr}_{G'}\left[w\right]$. 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 nonoccurrence of A for each $w \in W_{u,t}$. Putting it all together gives

$$\begin{aligned} \mathrm{Pr}_G\left[A \cap [H(u) = t]\right] &= \mathrm{Pr}_G\left[H(u) = t\right] \mathrm{Pr}_G\left[A \mid H(u) = t\right] \\ &= \sum_{w \in W_{u,t}} \mathrm{Pr}_G\left[w\right] I_A(w) \\ &= \sum_{w \in W_{u,t}} \mathrm{Pr}_{G'}\left[w\right] I_A(w) \\ &= \mathrm{Pr}_{G'}\left[A \cap [H(u) = t]\right], \end{aligned}$$

and the result. □

Acknowledgments. This material is based on work supported by the National Science Foundation under Grant No. 0514429, and by the AFOSR under Award No. FA9550-07-1-0124. 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) or the AFOSR.

References

[Aldous and Fill 09] David Aldous and James Fill. "Reversible Markov Chains and Random Walks on Graphs." In preparation, 2009. (Draft available at http://www.stat.berkeley.edu/users/aldous/RWG/book.html).

[Avrachenkov and Litvak 06] K. Avrachenkov and N. Litvak. "The Effect of New Links on Google PageRank." *Stochastic Models* 22:2 (2006), 319–331.

[Avrachenkov et al. 05] 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.

[Bianchini et al. 05] Monica Bianchini, Marco Gori, and Franco Scarselli. "Inside PageRank." *ACM Trans. Inter. Tech.* 5:1 (2005), 92–128.

[Brin and Page 98] Sergey Brin and Lawrence Page. “The Anatomy of a Large-Scale Hypertextual Web Search Engine.” *Computer Networks and ISDN Systems* 30:1–7 (1998), 107–117.

[Cheng and Friedman 05] Alice Cheng and Eric Friedman. “Sybilproof Reputation Mechanisms.” In *Proceeding of the 2005 ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems*, pp. 128–132. New York: ACM Press, 2005.

[Cheng and Friedman 06] Alice Cheng and Eric Friedman. “Manipulability of PageRank under Sybil Strategies.” In *Proceedings of the First Workshop of Networked Systems (NetEcon06)*, pp. 75–82. Available at http://www.cs.duke.edu/nicl/netecon06/papers/proceedings.pdf, 2006.

[Douceur 02] John Douceur. “The Sybil Attack.” In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pp. 251–260. London: Springer-Verlag, 2002.

[Fogaras and Rácz 04] Dániel Fogaras and Balázs Rácz. “Towards Scaling Fully Personalized PageRank.” In *Algorithms and Models for the Web-Graph: Third International Workshop, WAW 2004, Rome, Italy, October 16, 2004, Proceedings*, Lecture Notes in Computer Science 3243, pp. 105–117. Berlin: Springer, 2004.

[Friedman et al. 09] Eric Friedman, Paul Resnick, and Rahul Sami. “Manipulation-Resistant Reputation Systems.” In *Algorithmic Game Theory*, edited by N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani. To appear, 2009.

[Gyöngyi and Garcia-Molina 05a] Zoltán Gyöngyi and Hector Garcia-Molina. “Link Spam Alliances.” In *Proceedings of the 31st International Conference on Very Large Databases*, pp. 517–528. New York: ACM Press, 2005.

[Gyöngyi and Garcia-Molina 05b] Zoltán Gyöngyi and Hector Garcia-Molina. “Web Spam Taxonomy.” Paper presented at the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2005), Chiba, Japan, May 10–14, 2005.

[Gyöngyi et al. 04] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. “Combating Web Spam with TrustRank.” In *Proceedings of the 30th International Conference on Very Large Databases*, pp. 576–587. Amsterdam: Morgan Kaufmann, 2004.

[Gyöngyi et al. 06] 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*, pp. 439–450. New York: ACM, 2006.

[Heidelberger 95] Philip Heidelberger. “Fast Simulation of Rare Events in Queueing and Reliability Models.” *ACM Trans. Model. Comput. Simul.* 5:1 (1995), 43–85.

[Hopcroft and Sheldon 07] John E. Hopcroft and Daniel Sheldon. “Manipulation-Resistant Reputations Using Hitting Time.” In *Algorithms and Models for the Web-Graph: 5th International Workshop, WAW 2007, San Diego, CA, USA, December 11–12, 2007, Proceedings*, Lecture Notes in Computer Science 4863, edited by Anthony Bonato and Fan R. K. Chung, pp. 68–81. New York: Springer, 2007.

[Jeh and Widom 02] 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*. New York: ACM Press, 2002.

[Kamvar et al. 03] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. "The Eigentrust Algorithm for Reputation Management in P2P Networks." In *Proceedings of the 12th International Conference on World Wide Web*, pp. 640–651. New York: ACM Press, 2003.

[Langville and Meyer 04] Amy N. Langville and Carl D. Meyer. "Deeper inside PageRank." *Internet Mathematics* 1:3 (2004), 335–380.

[Liben-Nowell and Kleinberg 03] David Liben-Nowell and Jon Kleinberg. "The Link Prediction Problem for Social Networks." In *Proceedings of the Twelth International Conference on Information and Knowledge Management*, pp. 556–559. New York: ACM Press, 2003.

[Mason 05] Kahn Mason. "Detecting Colluders in PageRank: Finding Slow Mixing States in a Markov Chain." PhD thesis, Stanford University, 2005.

[Mitzenmacher and Upfal 05] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. New York: Cambridge University Press, 2005.

[Page et al. 98] 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.

[Zhang et al. 04] Hui Zhang, Ashish Goel, Ramesh Govindian, Kahn Mason, and Benjamin Van Roy. "Making Eigenvector-Based Reputation Systems Robust to Collusion." In In *Algorithms and Models for the Web-Graph: Third International Workshop, WAW 2004, Rome, Italy, October 16, 2004, Proceedings*, Lecture Notes in Computer Science 3243, pp. 92–104. Berlin: Springer, 2004.

John Hopcroft, Department of Computer Science, Cornell University, Ithaca, NY 14853 (jeh@cs.cornell.edu)

Daniel Sheldon, Department of Computer Science, Cornell University, Ithaca, NY 14853 (dsheldon@cs.cornell.edu)

Received February 14, 2008; accepted May 6, 2008.