Making Eigenvector-Based Reputation Systems Robust to Collusion

Hui Zhang[1], Ashish Goel[2], Ramesh Govindan[1], Kahn Mason[2], and Benjamin Van Roy[2]

[1] University of Southern California, Los Angeles CA 90007, USA
{huizhang, ramesh}@usc.edu
[2] Stanford University, Stanford CA 94305, USA
{ashishg, kmason, bvr}@stanford.edu

Abstract. Eigenvector based methods in general, and Google's PageRank algorithm for rating web pages in particular, have become an important component of information retrieval on the Web. In this paper, we study the efficacy of, and countermeasures for, *collusions* designed to improve user rating in such systems.

We define a metric, called the amplification factor, which captures the amount of PageRank-inflation obtained by a group due to collusions. We prove that the amplification factor can be at most $1/\epsilon$, where ϵ is the reset probability of the PageRank random walk. We show that colluding nodes (e.g., web-pages) can achieve this amplification and increase their rank significantly in realistic settings; further, several natural schemes to address this problem are demonstrably inadequate.

We propose a relatively simple modification to PageRank which renders the algorithm insensitive to such collusion attempts. Our scheme is based on the observation that nodes which cheat do so by "stalling" the random walk in a small portion of the web graph and, hence, their PageRank must be especially sensitive to the reset probability ϵ. We perform exhaustive simulations on the Web graph to demonstrate that our scheme successfully prevents colluding nodes from improving their rank, yielding an algorithm that is robust to gaming.

1 Introduction

Reputation systems are becoming an increasingly important component of information retrieval on the Web. Such systems are now ubiquitous in electronic commerce, and enable users to judge the reputation and trustworthiness of online merchants or auctioneers. In the near future, they may help counteract the free-rider phenomenon in peer-to-peer networks by rating users of these networks and thereby inducing social pressure to offer their resources for file-sharing [6, 10, 12]. Also, they may soon provide context for political opinion in the Web logging (blogging) world, enabling readers to calibrate the reliability of news and opinion sources.

A simple, and common way to measure a user's reputation is to use a referential link structure, a graph where nodes represent entities (users, merchants,

S. Leonardi (Ed.): WAW 2004, LNCS 3243, pp. 92–104, 2004.

authors of blogs) and links represent endorsements of one user by another. A starting point for an algorithm to compute user reputations might then be the class of eigenvector- or stationary distribution- based reputation schemes exemplified by the PageRank algorithm[1].

Algorithms based on link structure are susceptible to collusions; we make the notion of collusion more precise later, but for now we loosely define it as a manipulation of the link structure by a group of users with the intent of improving the rating of one or more users in the group. The PageRank algorithm published in the literature has a simple "resetting" mechanism which alleviates the impact of collusions. The PageRank value assigned to a page can be modeled as the fraction of time spent at that page by a random walk over the link structure; to reduce the impact of collusions (in particular, rank "sinks"), the algorithm resets the random walk at each step with probability ϵ.

In this paper, we define a quantity called the *amplification factor* that characterizes the amount of PageRank-inflation obtained by a group of colluding users. We show that nodes may increase their PageRank values by at most an *amplification factor* $\frac{1}{\epsilon}$; intuitively, a colluding group can "stall" the random walk for that duration before it resets. While this may not seem like much (a typical value for ϵ is 0.15), it turns out that the distribution of PageRank values is such that even this amplification is sufficient to significantly boost the *rank* of a node based on its PageRank value. What's worse is that all users in a colluding group could and *usually* do benefit from the collusion, so there is significant incentive for users to collude. For example, we found that it was easy to modify the link structure of the Web by having a low (say 10,000-th) ranked user collude[2] with a user of even lower rank to catapult themselves into the top-400. Similar results exist for links in other rank levels.

Two natural candidate solutions to this problem present themselves – identifying groups of colluding nodes, and identifying individual colluders by using detailed return time statistics from the PageRank random walk. The former is computationally intractable since the underlying optimization problems are NP-Hard. The latter does not solve the problem since we can identify scenarios where the return time statistics for the colluding nodes are nearly indistinguishable from those for an "honest" node.

How then, can PageRank based reputation systems protect themselves from such collusions? Observe that the ratings of colluding nodes are far more sensitive to ϵ than those of non-colluding nodes. This is because the PageRank values of colluding nodes are amplified by "stalling" the random walk; as explained before, the amount of time a group can stall the random walk is roughly $1/\epsilon$. This suggests a simple modification to the PageRank algorithm (called the *adaptive-*

[1] Although not viewed as such, PageRank may be thought of as a way of rating the "reputation" of web sites.

[2] Collusion implies intent, and our schemes are not able to determine intent, of course. Some of the collusion structures are simple enough that they can occur quite by accident.

resetting scheme) that allows different nodes to have different values of the reset probability. We have not been able to formally prove the correctness of our scheme (and that's not surprising given the hardness result), but we show, using extensive simulations on a real-world link structure, that our scheme significantly reduces the benefit that users obtain from collusion in the Web. Furthermore, while there is substantial intuition behind our detection scheme, we do not have as good an understanding of the optimum policy for modifying the individual reset probabilities. We defer an exploration of this to future work.

While we focus on PageRank in our exposition, we believe that our scheme is also applicable to other eigenvector-based reputation systems (e.g. [10, 12]). We should point out that the actual page ranking algorithms used by modern search engines (*e.g.*, Google) have evolved significantly and incorporates other domain specific techniques to detect collusions that are not (and will not be, for some time to come) in the public domain. But we believe that it is still important to study "open-source style" ranking mechanisms where the algorithm for rank computation is known to all the users of the system. Along with web-search, such an algorithm would also be useful for emerging public infrastructures (peer-to-peer systems and the blogosphere) whose reputation systems design are likely to be based on work in the public domain.

The remainder of this paper is organized as follows. We discuss related work in Section 2. In Section 3 we study the impact of collusions on the PageRank algorithm, in the context of the Web. Section 4 shows the hardness of making PageRank robust to collusions. In Section 5 we describe the *adaptive-resetting* scheme, and demonstrate its efficiency through exhaustive simulations on the Web graph. Section 6 presents our conclusions.

2 Related Work

Reputation systems have been studied heavily in non-collusive settings such as eBay [5, 14] – such systems are not the subject of study in this paper.

In the literature, there are at least two well-known eigenvector-based link analysis algorithms: HITS [11] and PageRank [16]. HITS was originally proposed to refine search outputs from Web search engines and discover the most influential web pages defined by the principal eigenvector of its link matrix. As discovering the principal eigenvector is the goal, original HITS doesn't assign a total ordering on the input pages, and collusion is less of a problem for it than for PageRank. On the contrary, PageRank was proposed to rank order input pages and handling clique-like subgraphs is a fundamental design issue.

Despite their difference, both algorithms have been applied into the design of reputation systems for distributed systems [10, 12]. These designs have mainly focused on the decentralization part, while their collusion-proofness still relies on the algorithm itself.

In the context of topic distillation on the Web, many extensions to PageRank and HITS algorithms [2, 4, 8, 9, 13] have been proposed for improving search-query results. Two general techniques - content analysis, and bias ranking with a seed link set - are used to handle problematic (spam) and irrelevant web links. While working well in their problem space, these approaches do not give answers to the algorithmic identification of collusions in a general link structure.

Ng *et al.* [15] studied the stability of HITS and PageRank algorithm with the following question in mind: when a small portion of the given graph is removed (*e.g.*, due to incomplete crawling), how severely do the ranks of the remaining pages change, especially for those top ranked nodes? They show that HITS is sensitive to small perturbations, while PageRank is much more stable. They proposed to incorporate the PageRank's "reset-to-uniform-distribution" into HITS to enhance its stability.

Finally, for context, we briefly describe the original PageRank algorithm with its random walk model. Given a directed graph, a random walk W starts its journey on any node with the same probability. At the current node x, with probability $(1-\epsilon)$ W jumps to one of the nodes that have links from x (the choice of neighbor is uniform), and with probability ϵ, W decides to restart (*reset*) its journey and again choose any node in the graph with the same probability. Asymptotically, the stationary probability that W is on node x is called the PageRank value of x, and all nodes are ordered based on the PageRank values.

In the rest of the paper, we use the term *weight* to denote the PageRank (PR) value, and *rank* to denote the ordering. We use the convention that the node with the largest PR weight is ranked first.

3 Impact of Collusions on PageRank

In this section, we first show how a *group* of nodes could modify the referential link structure used by the PageRank algorithm in order to boost their PageRank weights by up to $1/\epsilon$. We then demonstrate that it is possible to induce simple collusions in real link structures (such as that in the Web) in a manner that raises the *ranking* of colluding nodes[3] significantly.

3.1 Amplifying PageRank Weights

In what follows, we will consider the PageRank algorithm as applied to a directed graph $G = (V, E)$. $N = |V|$ is the number of the nodes in G. A node in G corresponds, for example, to a Web page in the Web graph, or a blog in the blog graph; an edge in G corresponds to a reference from one web page to another, or from one blog to another. Let $d(i)$ be the out-degree of node i, and $W_v(i)$ be the *weight* that the PageRank algorithm computes for node i. We define on each edge $e_{ij} \in E$ the weight $W_e(e_{ij}) = \frac{W_v(i)\times(1-\epsilon)}{d(i)}$.

Let $V' \subset V$ be a set of nodes in the graph, and let G' be the subgraph induced by V'. E' is defined to be the set of all edges e_{ij} such that at least one of i and j is in V'. We classify the edges in E' into three groups:

[3] We use "pages", "nodes" and "users" interchangeably in the rest of the paper.

In Links: An edge e_{ij} is an in link for G' if $i \notin V'$ and $j \in V'$. E'_{in} denotes the set of in links of G'.

Internal Links: An edge e_{ij} is an internal link for G' if $i \in V'$ and $j \in V'$. $E'_{internal}$ denotes the set of internal links of G'.

Out Links: An edge e_{ij} is an out link for G' if $i \in V'$ and $j \notin V'$. E'_{out} denotes the set of out links of G'.

One can then define two types of weights on G':

- $W_{in}(G') = \sum_{e:e \in E'_{in}} W_e(e) + \frac{N'}{N}$, $N = |V|$, $N' = |V'|$.
- $W_G(G') = \sum_{v:v \in V'} W_v(v)$.

Intuitively, $W_{in}(G')$ is, in some sense, the "actual" weight that should be assigned to G', when G' is regarded in its entirety (*i.e.* as one unit). On the other hand, $W_G(G')$ is the total "reputation" of the group that would be assigned by PageRank. Note that nodes within G' can boost this reputation by manipulating the link structure of the internal links or the out links.

Then, we can define a metric we call the *amplification factor* of a graph G as $Amp(G) = \frac{W_G(G)}{W_{in}(G)}$. Given this definition, we prove (see Appendix A in the companion technical report [17] for the proof) the following theorem:

Theorem 1. *In the original PageRank system,* $\forall G' \subseteq G, Amp(G') < \frac{1}{\epsilon}$.

3.2 PageRank Experiments on the Real-World Graph

It might not be surprising to find out that the weight inflation in PageRank groups could be as high as $\frac{1}{\epsilon}$, since it's already known from [15] that eigenvector-based reputation systems are not stable under link structure perturbation. However, it's not clear what is the practical import of amplifying PageRank weights. Specifically, is it easy for a group of colluding nodes to achieve the upper bound of the amplification factor, $\frac{1}{\epsilon}$? Can nodes improve their *ranking* significantly?

To answer these questions, we obtained a large Web subgraph from the Stanford WebBase [18]. It contains upwards of 80 million URLs, and is called $\mathcal{W}$ in the rest of the paper. We then modified one or more subgraphs in $\mathcal{W}$ to simulate collusions, and measured the resulting PageRank weights for each node. We tried a few different modifications, and report the results for one such experiment.

Our first experiment on $\mathcal{W}$ is called *Collusion200*. This models a small number of web pages *simultaneously* colluding. Each collusion consists of a pair of nodes with *adjacent* ranks. Such a choice is more meaningful than one between a low ranked node and a high ranked node, since the latter could have little incentive to collude. Each pair of nodes removes their original out links and adds one new out link to each other. In the experiment reported here, we induce 100 such collusions at nodes originally ranked around 1000th, 2000th, . . ., 100000th.

There is a subtlety in picking these nodes. We are given a real-world graph in which there might already be colluding groups (intentional or otherwise). For this reason, we carefully choose our nodes such that they are unlikely to be already colluding (the precise methodology for doing this will become clear in Section 5.2 when we describe how we can detect colluding groups in graphs).

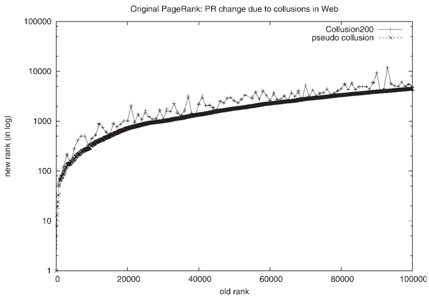

Fig. 1. $\mathcal{W}$: New PR rank after *Collusion200*

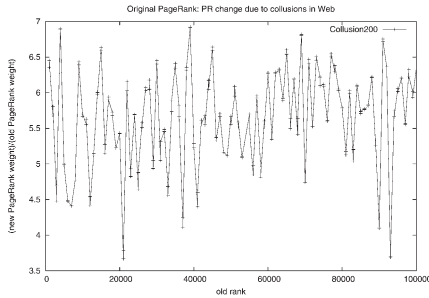

Fig. 2. $\mathcal{W}$: New PR weight (normalized by old PR weight) after *Collusion200*

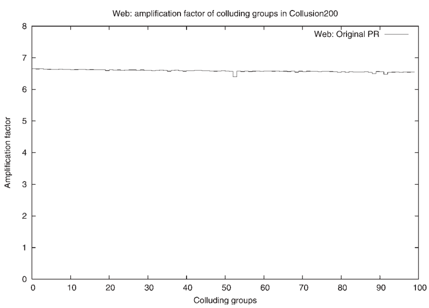

Fig. 3. Amplification factors of the 100 colluding groups in Collusion200

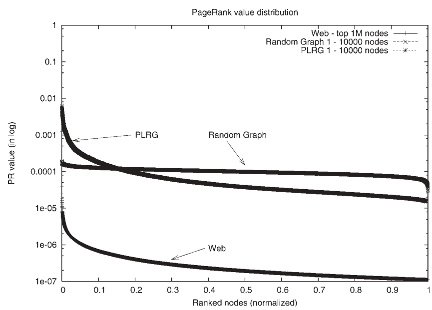

Fig. 4. PR distribution in 3 topologies

We calculate the PageRank weights and ranks for all nodes before (called old rank and weight) and after (called new rank and weight) *Collusion200* on $\mathcal{W}$ with $\epsilon = 0.15$ (a default value assumed in [16]). Figures 1 & 2 show the rank and weight change for those colluding nodes. In addition, we also plot in Figure 1 the rank that each colluding node could have achieved if its weight were amplified by $\frac{1}{\epsilon}$ while all other nodes remained unchanged in weight, which we call *pseudo collusion*.

As we can see, all colluding nodes increased their PR weight by at least 3.5 times, while the majority have a weight amplification over 5.5. More importantly, collusion boosts their ranks to be more than 10 times higher and close to the best achievable. For example, a colluding node originally ranked at 10002th had a new rank at 451th, while the 100005th node boosted its rank to 5033th by colluding with the 100009th node, which also boosted its rank to 5038th.

Thus, even concurrent, simple (2-node) collusions of nodes with comparable original ranks can result in significant rank inflation for the colluding nodes. For another view of this phenomenon in Figure 3 we plot the amplification factors achieved by the colluding groups in $\mathcal{W}$. It clearly shows that almost all colluding groups attain the upper bound.

But what underlies the significant *rank inflation* in our results? Figure 4 shows the PageRank weight distribution of $\mathcal{W}$ (only top 1 million nodes for interest). It also includes, for calibration, the PageRank weight distribution on power low random graph (PLRG) [1] and the classical random graph [3] topologies. First, observe that the random graph has a flat curve, which implies that in such a topology, almost any nodes could take one of the top few positions by amplifying its weight by $\frac{1}{\epsilon}$. Secondly, $\mathcal{W}$ and $PLRG$ share the same distribution characteristic, *i.e.*, the top nodes have large weights, but the distribution flattens quickly after that. This implies that in these topologies, low ranked nodes can inflate their ranks by collusion significantly (though perhaps not to the top 10).

While we have discussed only one experiment with a simple collusion scheme, there are many other schemes through which nodes can successfully achieve large rank inflation (Section 5.2 presents a few such schemes). We believe, however, that our finding is both general (*i.e.*, not constrained to the particular types of collusions investigated here) and has significant practical import since $\mathcal{W}$ represents a non-trivial portion of the Web. Having established that collusions can be a real problem, we now examine approaches to making the PageRank algorithm robust to collusions.

4 On the Hardness of Making PageRank Robust to Collusions

We will now explore two natural approaches to detecting colluding nodes, and demonstrate that neither of them can be effective.

The first approach is to use finer statistics of the PageRank random walk. Let the random variable X_v denote the number of time steps from one visit of node v to the next. It is easy to see that the PageRank value of v is exactly $1/\mathbf{E}[X_v]$ where $\mathbf{E}[X_v]$ denotes the expectation of X_v. For the simplest collusion, where two nodes A and B delete all their out-links and start pointing only to each other, the random walk will consist of a long alternating sequence of A's and B's, followed by a long sojourn in the remaining graph, followed again by a long alternating sequence of A's and B's, and so on [4] . Clearly, X_A is going to be 2 most of the time, and very large (with high probability) occasionally. Thus, the ratio of the variance and the expectation of X_A will be disproportionately large. It is now tempting to suggest using this ratio as an indicator of collusion.

Unfortunately, there exist simple examples (such as large cycles) where this approach fails to detect colluding nodes. We will present a more involved example where not just the means and the variances, but the *entire distributions* of X_H and X_C are nearly identical; here H is an "honest" node and C is cheating to improve its PageRank. The initial graph is a simple star topology. Node 0 points to each of the nodes $1 \ldots N$ and each of these nodes points back to node 0 in turn. Now, node N starts to cheat; it starts colluding with a new node $N+1$

[4] Incidentally, it is easy to show that this collusion mode can achieve the theoretical upper bound of $1/\epsilon$ on the amplification factor.

so that N and $N+1$ now only point to each other. The new distributions X_0 and X_N can be explicitly computed, but the calculation is tedious. Rather than reproduce the calculation, we provide simulation results for a specific case, where $N = 7$ and $\epsilon = 0.12$. Figure 5 shows the revisit distribution for nodes 0 (the original hub) and 7 (the cheating node). The distributions are nearly identical. Hence, any approach that relies solely on the detailed statistics of X_v is unlikely to succeed.

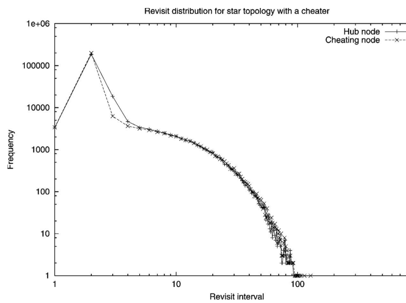

Fig. 5. Frequency of revisit intervals for the cheating node (node 7) and the honest node (node 0) for the star-topology. The simulation was done over 1,000,000 steps

Thus, a more complete look at the graph structure is needed, one that factors in the various paths the random walk can take. One natural approach to identifying colluders would be to directly find the subgraph with the maximum amplification (since colluders are those with high amplification). However, it is very unlikely that this problem is tractable. Consider the intimately related problem of finding a group S of size k which maximizes the difference of the weights, $W_G(S) - W_{in}(S)$, rather than the ratio. This problem is NP-Hard via reduction to the densest k-subgraph problem [7]. Details of the reduction are in the companion technical report [17]. There are no good approximation algorithms known for the densest k-subgraph problem (the best known is $O(N^{1/3})$). The reduction is approximation preserving. Hence, identifying colluding groups is unlikely to be computationally tractable even in approximate settings.

This suggests that our goals should be more modest – rather than identifying the entire colluding group, we focus on finding individual nodes that are cheating. This is the approach we take in the next section.

5 Heuristics for Making PageRank Robust to Collusions

Given our discussion of the hardness of making PageRank robust to collusions, we now turn our attention to heuristics for achieving this. Our heuristic is based on an observation explained using the following example. Consider a small (compared to the size of the original graph) group S of colluding nodes. These nodes can not influence links from $V - S$ into S. Hence, the only way these nodes can increase their stationary weight in the PageRank random walk is by stalling the random walk *i.e.* by not letting the random walk escape the group. But in the PageRank algorithm, the random walk resets at each node with probability ϵ. Hence, colluding nodes must suffer a significant drop in PageRank as ϵ increases.

This forms the basis for our heuristic for detecting colluding nodes. We expect the stationary weight of colluding nodes to be highly correlated [5] with $1/\epsilon$ and that of non-colluding nodes to be relatively insensitive to changes in ϵ . While our hypothesis can be analytically verified for some interesting special cases (details in the companion technical report [17]), we restrict ourselves to experimental evidence in this paper.

5.1 The Adaptive-Resetting Heuristic

The central idea behind our heuristic for a collusion-proof PageRank algorithm is that the value of the reset probability is *adapted*, for each node, to the degree of collusion that the node is perceived to be engaged in. This *adaptive-resetting* scheme consists of two phases:

1. Collusion detection
   (a) Given the topology, calculate the PR weight vector under different ϵ values.
   (b) Calculate the correlation coefficient between the curve of each nodes x's PR weight and the curve of $\frac{1}{\epsilon}$. Label it as $\textit{co-co}(x)$, which is our proxy for the collusion of x. $\textit{co-co}(x) = \textit{co-co}(x) < 0 \;?\; 0 : \quad \textit{co-co}(x)$.
2. ϵ Personalization
   (a) Now the node x's *out-link* personalized-$\epsilon = F(\epsilon_{default}, co - co(x))$.
   (b) The PageRank algorithm is repeated with these personalized-ϵ values.

The function $F(\epsilon_{default}, co - co(x))$ provides a knob for a system designer to appropriately punish colluding nodes. In our experiments we tested two functions:

Exp. function $F_{Exp} = \epsilon_{default}^{(1.0-\text{co-co}(x))}$.
Linear function $F_{Linear} = \epsilon_{default} + (0.5 - \epsilon_{default}) \times \text{co-co}(x)$.

The choice of function is subjective and application-dependent, and given space limitations, we mostly present results based on F_{Exp}.

5.2 Experiments

As in Section 3, we conducted experiments on the $\mathcal{W}$ graph. In all experiments with our adaptive-resetting scheme, we chose seven ϵ values in the *collusion*

[5] The correlation coefficient of a set of observations $(x_i, y_i) : i = 1, .., n$ is given by

$$\text{co-co}(x, y) = \frac{\sum_{i=1,\cdots,n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1,\cdots,n}(x_i - \overline{x})^2 \sum_{i=1,\cdots,n}(y_i - \overline{y})^2}}.$$

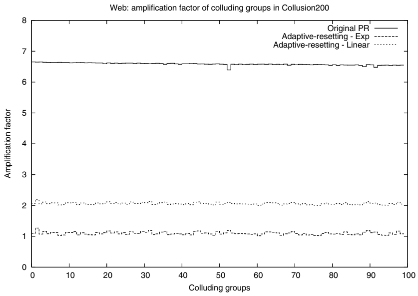

Fig. 6. $\mathcal{W}$: amplification factors of the 100 colluding groups in Collusion200

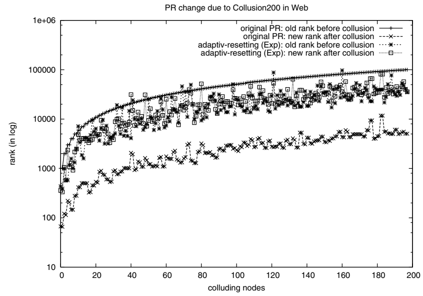

Fig. 7. $\mathcal{W}$: PR rank comparison between original PageRank and Adaptive-ϵ scheme in *Collusion200*

detection phase – $0.6, 0.45, 0.3, 0.15, 0.075, 0.05$, and 0.0375 – and used 0.15 as $\epsilon_{default}$. While there are eight PageRank calculations, the actual computational time for the adaptive-resetting scheme was only 2-3 times that of the original PageRank algorithm. This is because the computed PR weight vector for one ϵ value is a good initial state for the next ϵ value.

Basic Experiment: We first repeated the experiment *Collusion200* for adaptive-resetting scheme. As mentioned in Section 3.2, all the colluding nodes are chosen from the nodes unlikely to be already colluding, and this is judged by their *co-co* values in the original topology. Precisely, we select nodes with $co - co(x) \leq 0.1$. Choosing nodes with arbitrary *co-co* values doesn't invalidate the conclusions in this paper (as discusses in the companion technical report [17]), but our selection methodology simplifies the exposition of our scheme.

We compared the original PageRank algorithm, the adaptive-resetting schemes using F_{Exp} and F_{Linear}. As shown in Figure 6, the adaptive-resetting scheme F_{Exp} restricted the amplification factors of the colluding groups to be very close to one, and F_{Linear} also did quite well compared to the original PageRank.

In Figure 7 we compare the original PageRank and the adaptive-resetting scheme using F_{Exp} based on the old and new rank before and after *Collusion200* in $\mathcal{W}$. For the original PageRank algorithm the rank distribution clearly indicates how nodes benefit significantly from collusion. The curves for the adaptive-resetting scheme nearly overlap, illustrating the robustness of our heuristic. Furthermore, note that the curves of the PageRank algorithm before collusions and the adaptive-resetting before collusions are close to each other, which means the weight of non-colluding nodes is not affected noticeably when applying the adaptive-resetting scheme instead of the original PageRank scheme.

Other Collusion Topologies an Experiment with Miscellaneous Collusion Topologies: We tested adaptive-resetting scheme under other collusion topologies in an experiment called *Collusion22*. In *Collusion22* 22 low co-co (≤ 0.1) nodes are selected for 3 colluding groups:

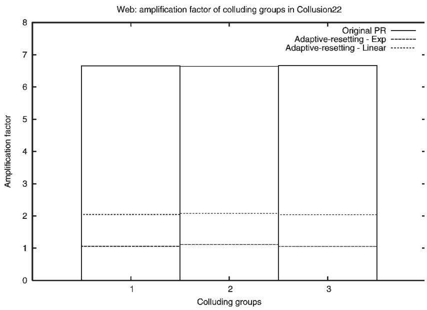

Fig. 8. $\mathcal{W}$: Amplification factors of the 3 colluding groups in Collusion22

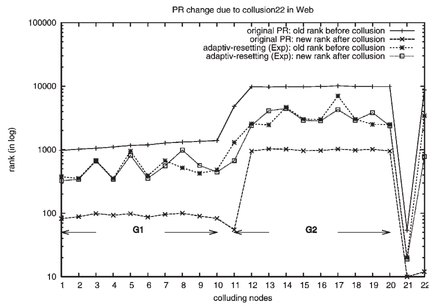

Fig. 9. $\mathcal{W}$: PR rank comparison between original PageRank and Adaptive-ϵ scheme in *Collusion22*

G1. $G1$ has 10 nodes, which remove their old out links and organize into a single-link ring. All nodes have their original ranks at around 1000th.

G2. $G2$ has 10 nodes, which remove their old out links and organize into a star topology by one hub pointing to the other 9 nodes and vice versa. The hub node has its original rank at around 5000th, while the other nodes are ranked at around 10000th originally.

G3. $G3$ has 2 nodes, which remove the old out links and organize into a circle. One is originally ranked at around 50th, and the other at around 9000th.

We ran *Collusion22* on $\mathcal{W}$ using both original PageRank and adaptive-resetting scheme. We first observed that the adaptive-resetting scheme *successfully detected all 22 colluding nodes* by reporting high *co-co* values (> 0.96).

In Figure 8, we compare the original PageRank algorithm, the adaptive-resetting schemes with function F_{Exp} and F_{Linear} based on the metric *amplification factor* under *Collusion22*. As in Figure 6 the two adaptive-resetting schemes successfully restricted the weight amplification for the colluding nodes.

In Figure 9 we compare original PageRank and adaptive-resetting scheme with function F_{Exp} based on the old and new rank before and after *Collusion22* in $\mathcal{W}$. The results for the graph $\mathcal{B}$ were similar and are omitted. As we can see, the nodes of $G1$ and $G2$ seem to have some rank improvement in adaptive-resetting before collusions compared to their ranks in the PageRank algorithm before collusion, while their weights have increased only marginally. This is due to the rank drop of many high rank nodes with high *co-co* values in *adaptive-resetting before collusions*. Lastly, it is interesting to observe that with the original PageRank algorithm, even the two nodes with significantly different ranks in $G3$ can benefit mutually from a simple collusion: the 8697th node rocketed to the 12th, and the 54th node also jumped to the 10th position.

More Experiment Results: We have done more experiments to validate the correctness of adaptive-resetting scheme. Due to the space limit, we do not present them here and refer the interested readers to the companion technical report [17] for details.

6 Conclusion

In this paper we studied the robustness of one eigenvector-based rating algorithm: PageRank. We point out the importance of collusion detection in PageRank based reputation systems for real-world graphs, its hardness, and then a heuristic solution. Our solution involves detecting colluding nodes based on the sensitivity of their PageRank value to the resetting probability ϵ and then penalizing them by assigning them a higher reset probability. We have done extensive simulations on the Web graph to demonstrate the efficacy of our heuristic.

Acknowledgement

We would like to thank the Stanford Webbase group for making a pre-processed copy of the Web link structure available to us.

References

1. W. Aiello, F. Chung, and L. Lu. *A Random Graph Model for Massive Graphs.* the 32nd Annual Symposium on Theory of Computing, 2000.
2. K. Bharat, M. R. Henzinger. *Improved Algorithms for Topic Distillation in a Hyperlinked Environment.* Proceedings of SIGIR98, 1998.
3. B. Bollobas. *Random Graphs.* Academic Press, Inc. Orlando, Florida, 1985.
4. S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, J. Kleinberg. *Automatic resource compilation by analyzing hyperlink structure and associated text.* Proceeedings of the Seventh International Conference on World Wide Web,, 1998.
5. C. Dellarocas. *Analyzing the economic efficiency of eBay-like online reputation reporting mechanisms.* In Proceedings of the 3rd ACM Conference on Electronic Commerce, pages 171–179, 2001.
6. D. Dutta, A. Goel, R. Govindan, H. Zhang. *The Design of A Distributed Rating Scheme for Peer-to-peer Systems.* First Workshop on Economics of Peer-to-peer Systems, UC Berkeley, June 2003.
7. U. Feige and M. Seltser, *On the densest k-subgraph problems.* Technical report no. CS97-16, Department of Applied Math and Comp. Sci., Weizmann Institute, 1997.
8. Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. *Combating Web Spam with TrustRank.* Technical Report, Stanford University, 2004.
9. T. Haveliwala. *Topic-sensitive PageRank.* In Proceedings of WWW2002, 2002.
10. S. Kamvar, M. Schlosser and H. Garcia-Molina. *EigenRep: Reputation Management in P2P Networks.* in Proc. WWW Conference, 2003.
11. J. M. Kleinberg. *Authoritative Sources in a Hyperlinked Environment.* Journal of the ACM, Vol.46, No.5, P604-632, 1999.
12. H.T. Kung, C.H. Wu. *Differentiated Admission for Peer-to-Peer Systems: Incentivizing Peers to Contribute Their Resources.* First Workshop on Economics of Peer-to-peer Systems, UC Berkeley, June 2003.

13. L. Li , Y. Shang , W. Zhang. *Improvement of HITS-based algorithms on web documents.* Proceedings of WWW2002, 2002.
14. N. Miller, P. Resnick, and R. Zeckhauser. *Eliciting honest feedback in electronic markets.* RWP02-39, Harvard Kennedy School of Government, 2002.
15. A. Y. Ng, A. X. Zheng, and M. I. Jordan. *Link analysis, eigenvectors, and stability.* International Joint Conference on Artificial Intelligence (IJCAI), 2001.
16. L. Page, S. Brin, R. Motwani, and T. Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.* Stanford Digital Library Technologies Project, 1998.
17. The companion technical report is available at *ftp://ftp.usc.edu/pub/csinfo/tech-reports/papers/04-817.pdf*
18. The Stanford WebBase Project. *http://www-diglib.stanford.edu/testbed/doc2/WebBase/*