

CS599: Structure and Dynamics of Networked Information (Spring 2005)
01/24/2005: The Efficacy of Collusions in Web Ranking and the Counter-measures
Scribes: Shishir Bharathi, Chansook Lim

1 The Collusion Problem

There are several search engine optimization (SEO) sites on the web today aimed at improving the visibility of their clients' web pages. For a search engine like Google, this is done by boosting the PageRank [2] of the client's pages. This is also a common webspamming technique. To understand how this works, we need to first understand some of the internal workings of a search engine.

Search Engines typically use two factors to decide the rank score:

- Relevance: A measure of textual similarity between the query terms and the page.
- Importance: The global popularity of the page which is independent of the query terms.

At Google, the rank depends on the product of relevance and importance. It is, however, difficult to say which of the two is a bigger factor in deciding the rank. An SEO can try to manipulate both of these factors in trying to boost the rank of the page. In this lecture, we're concerned only with efforts to try and increase the importance of the page by manipulating the link structure around the page.

Def. Collusion: A manipulation of the hyperlink structure by a group of users with the intention of improving the rating of one or more users¹ in the group.

2 Brief introduction to PageRank

PageRank (PR) is an eigen-vector based rating scheme to rank hypertext documents on the WWW. It is based on a random walk model where the walker jumps from a node in the graph to a random node with a probability ε (the resetting probability) and continues to walk to a random successor node with a probability $1 - \varepsilon$. The ranking algorithm is an iterative algorithm that calculates the importance of a page A based on the importance of its parent pages (pages that link into A). Originally, the PageRank of a page A was given by

$$PR(A) = (1 - d) + d(PR(T_1)/C(T_1) + \dots + PR(T_n)/C(T_n))$$

where $T_1, T_2 \dots T_n$ are pages pointing to A and $C(T_i)$ is the number of outgoing links from T_i . As time goes on, the expected percentage of steps for the walker to be each each node v converges to $PR(v)$. This can be interpreted as the percentage of time that a surfer spends at each page and thus as a measure of its popularity. The question, therefore, is whether PR is collusion-proof and whether a node can easily boost its rank by manipulating its out-going links with others.

3 Collusion detection

3.1 A metric on group collusion

Consider a node group G . For a subgroup G' , the *Amplification Factor*[1] is defined as

¹Research Question: Is link structure manipulation mutually beneficial to all members of the group? What are the benefit patterns?

$$Amp(G') = \frac{W_G(G')}{W_{in}(G')}$$

where,

$$W_G(G') = \sum_{i:i \in G'} PR(i)$$

$$W_{in}(G') = \sum_{(i,j):i \notin G',j \in G',\exists i \rightarrow j} \frac{PR(i)}{out(i)}(1 - \varepsilon) + \frac{|G'|}{|G|}(1 - W_G(G'))\varepsilon$$

The Amplification Factor is therefore, the ratio of the PR of the page after collusion to its “real” PR. The larger this value, the greater the possibility of collusion. It’s also possible that $Amp(G')$ might have a high value due to unintentional collusion – where the members are not really trying to boost each other’s rank. Ideally, for any G' , this value should be as close to 1 as possible. In the original PageRank system,

$$\forall G' \subset G, Amp(G') \leq \frac{2}{\varepsilon}$$

Specifically, when $\frac{|G'|}{|G|} \ll 1$,

$$Amp(G') \leq \frac{1}{\varepsilon}$$

Since ε was about 0.15, $Amp(G') \leq 7$.

4 Ranking’s sensitivity to PageRank weight

A group of nodes can modify the link structure to boost their PR weights. (PR Weight here, refers to the stationary value as calculated by the PageRank algorithm. The page with the highest weight is ranked first in Google’s rankings). The following experiment was conducted to find out how sensitive the rank is to PR weight. The topologies considered were:

1. \mathcal{W} : A web link topology, containing the link structure of 80M+ urls
2. \mathcal{B} : containing the blogroll structure of 72,000+ blogs

The experiment modeled a small number of web pages *simultaneously* colluding, by having 200 nodes participate in 100 colluding groups. Each group had 2 nodes of adjacent ranks, in the circle topology. The chosen node pairs originally had ranks 1000^{th} , 2000^{th} ... 100000^{th} , and had an original amplification factor very close to the upper bound of $\frac{1}{\varepsilon}$. Other outgoing links from each node were removed. ε was set to a low value of 0.15.

We see that after collusion, the amplification factor goes to about 7 for all nodes. The PR value increases dramatically. e.g. From about 1005^{th} to 67^{th} , 10001^{th} to 450^{th} etc. This can be explained from the PR weight distribution in the original graph. The PR weight distributions for 4 topologies (Web, Blog, Random Graph and Power Law Random Graph) can be seen in Figure 2.

5 Detecting Collusions

The problem of detecting colluding groups is based on these well known problems:

- The densest k-subgraph problem [3]

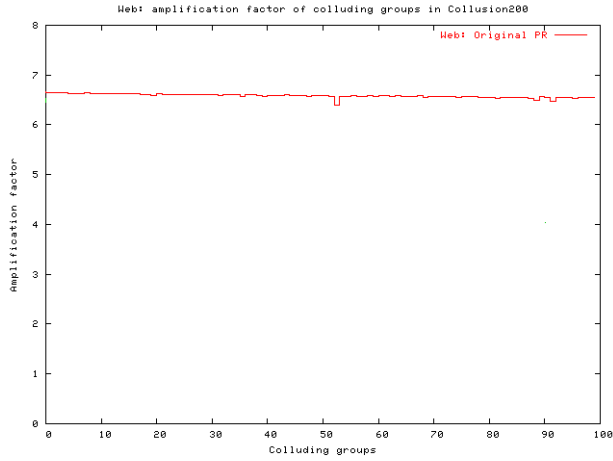


Figure 1: Original Amplification Factors

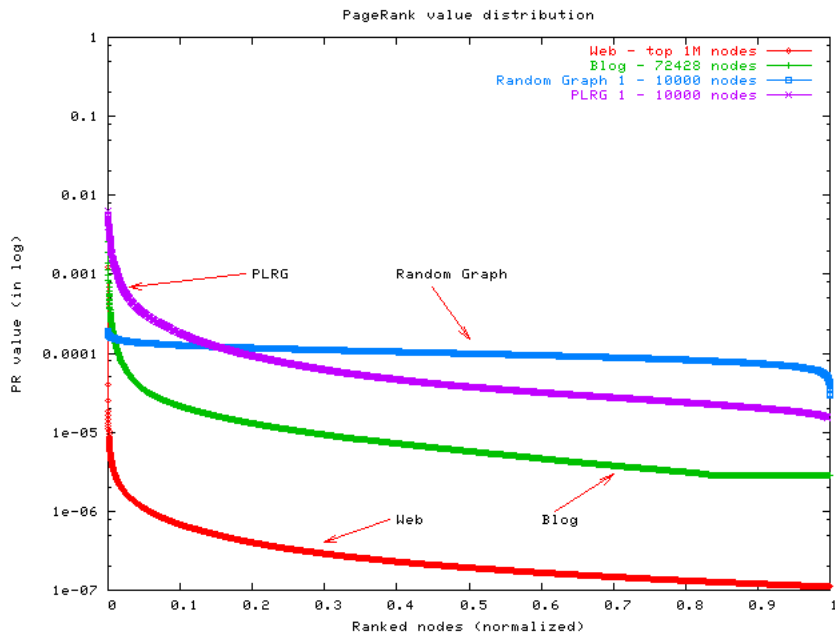


Figure 2: PR weight distributions for different topologies

- The classical CLIQUE problem
- The problem of finding large hidden cliques in random graphs [4]

It can be concluded that identifying colluding groups is unlikely to be computationally tractable.

Theorem 1 *Max $_{G' \subset G} \text{Amp}(G')$ is an NP-hard problem.*

An alternative is to use the finer statistics of the random walk algorithm to identify colluding groups. The revisit interval of a random walk algorithm on a colluding node is likely to have a large variance with its expectation. This is because the colluding nodes boost their PR weights by “stalling” the random walk (RW). For example, consider the circle topology from the experiment. Once the RW reaches a node in the circle, the probability of its getting out of the circle is low. As a consequence, it turns out that when the resetting probability ε increases, the colluding nodes should suffer a significant drop in their PR weight. This can be used in their detection. Consider the case of N nodes each of which links to all of the others. Suppose $K \ll N$ of these nodes start colluding and remove all outgoing links to the other $N - K$ nodes. For a colluding node,

$$PR(X) = \frac{1}{K + (N - K)\varepsilon} \approx \frac{1}{N\varepsilon}$$

For an honest node,

$$PR(X) = \frac{\varepsilon}{K + (N - K)\varepsilon} \approx \frac{1}{N}$$

6 Adaptive Resetting scheme

Using the above results, collusion is detected as follows:

- Given the topology, calculate the PR vector for different values of ε . $\{\varepsilon\} = \{0.0375, 0.05, 0.075, 0.15, 0.3, 0.45, 0.6\}$
- Calculate the Correlation Coefficient between the curve of each node X 's PR weight and the curve of $\frac{1}{\varepsilon}$. Call this *co-co*(X).

The *co-co* value is used in calculating a new resetting probability for each node:

- Calculate each node X 's out link personalized- $\varepsilon = F(\varepsilon_{default}, co-co(X))$, where F is one of:

$$\begin{aligned} - \textit{ExponentialFunction} \quad F_{exp} &= \varepsilon_{default}^{(1-co-co(X))} \\ - \textit{LinearFunction} \quad F_{Linear} &= \varepsilon_{default} + (0.5 - \varepsilon_{default}) * co-co(X) \end{aligned}$$

- The final PR weight vector is calculated using these personalized resetting probabilities.

The results of performing the collusion200 experiment after including the adaptive resetting scheme are shown in Figure 3. In the first graph, we see that the amplification factor is greatly reduced once adaptive resetting is employed. The exponential function defined above seems to perform better than the linear function and is consistently close to the ideal value of 1. The second graph has two sets of plots which show how this scheme affects the ranks of colluding nodes. We see that originally, colluding nodes could easily get better ranks. Once the adaptive resetting scheme is employed, the rankings do not change much when nodes start colluding. This shows that the PR weights calculated using the personalized resetting probabilities are very close to what the actual values should be.

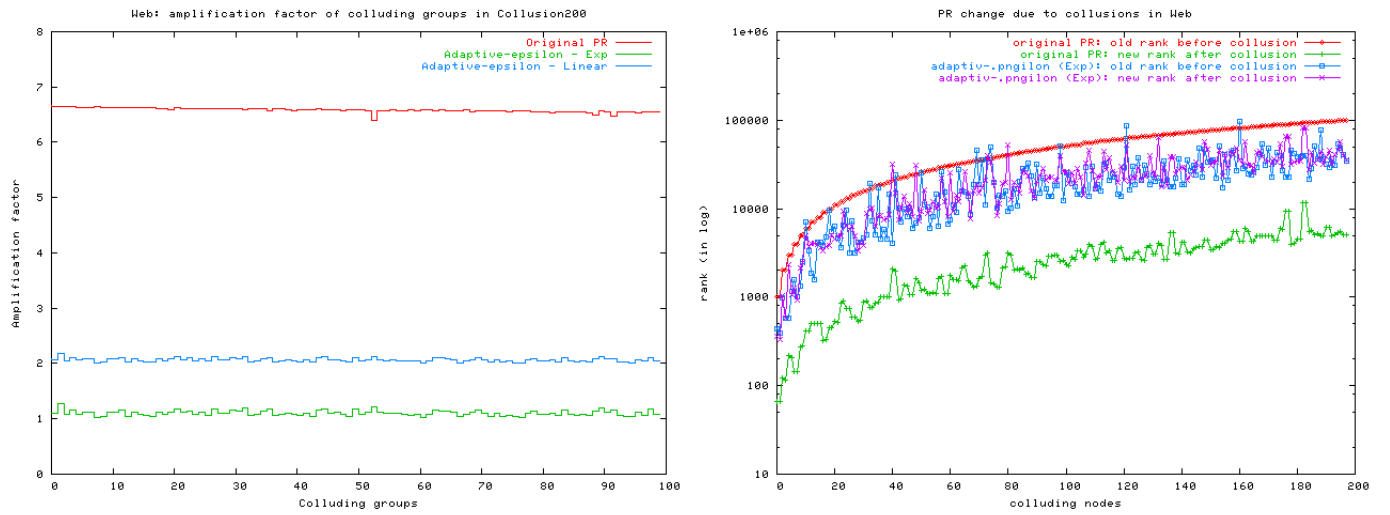


Figure 3: Results of Collusion200 after including adaptive resetting

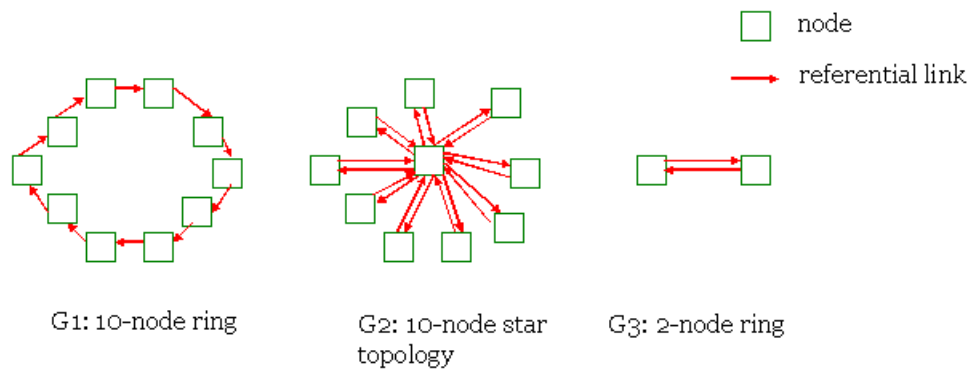


Figure 4: Topologies considered in Experiment collusion22

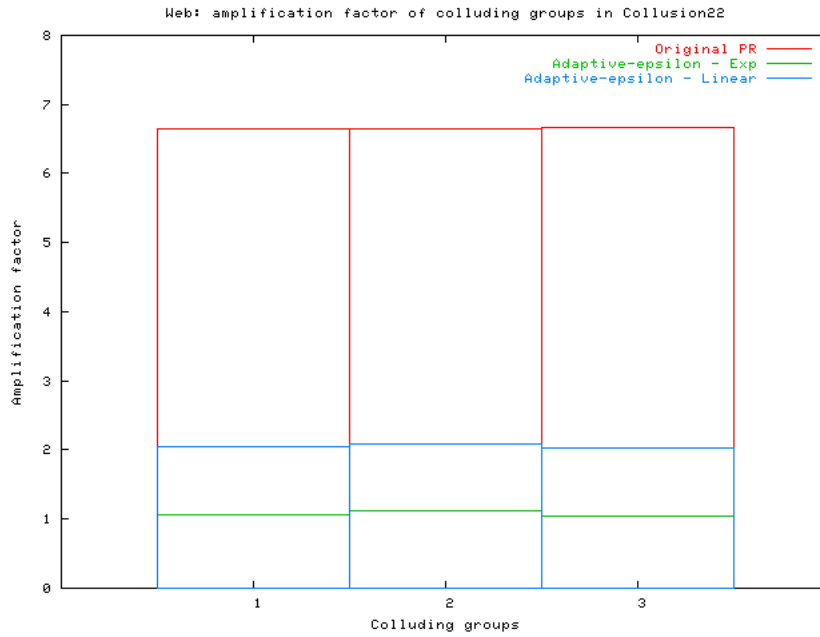


Figure 5: Amplification Factors for colluding groups

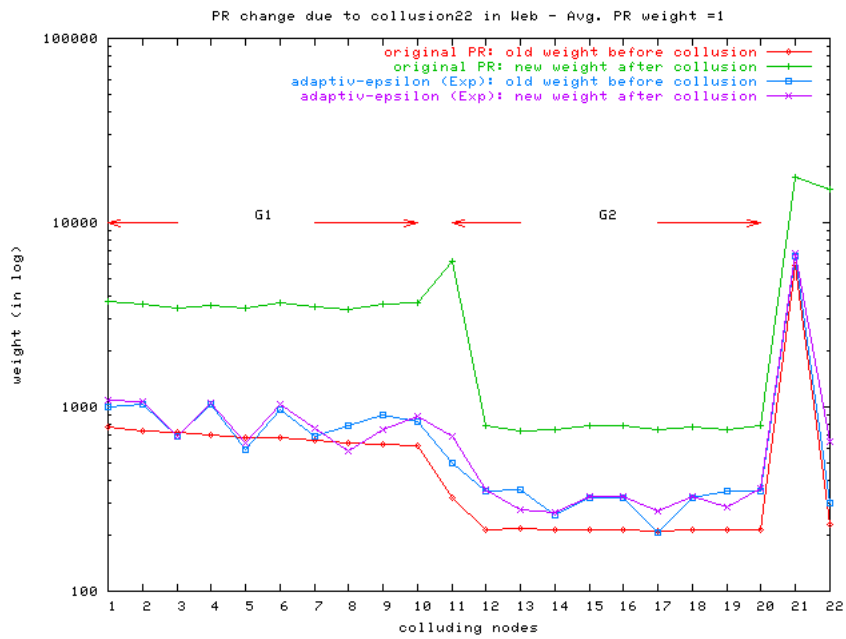


Figure 6: PR weights after adaptive resetting

6.1 Application to various types of colluding subgraphs

Experiment collusion22 was conducted to test how the adaptive resetting scheme performs when the colluding nodes arrange themselves into different topologies. The results are presented for the topologies shown in Figure 4.

Figure 5 shows the amplification factors for reach of the topologies (in W , the Stanford WebBase graph) before and after employing the adaptive resetting scheme. We see that with adaptive resetting, the amplification factor falls back to the desired value of 1 for all topologies.

Figure 6 shows a comparison between original PR weight and the value calculated using the adaptive-resetting scheme for both G1 and G2 topologies (in W). Once again, we see that earlier, the PR weights increased after collusion. After the adaptive resetting scheme is employed, we see that there is not much change in the PR weights before and after collusion. This shows that the scheme is effective in dealing with colluding nodes in varying topologies and the PR weights calculated is very close to the actual (ideal) weights.

Figure 7 shows the top 25 ranked urls from the W topology before and after applying the adaptive resetting scheme. It was found that <http://www.yahoo.com> and <http://messenger.yahoo.com> were in collusion. <http://messenger.yahoo.com> does not feature in the top 25 urls after applying the adaptive resetting scheme.

Rank	Old list	New list
1	http://www.yahoo.com/	http://www.tucows.com/
2	http://messenger.yahoo.com/	http://www.yahoo.com/
3	http://www.tucows.com/	http://www.domaindirect.com/
4	http://www.domaindirect.com/	http://news.tucows.com/
5	http://news.tucows.com/	http://ispcentral.tucows.com/
6	http://ispcentral.tucows.com/	http://www.microsoft.com/
7	http://www.microsoft.com/	http://www.acme.com/software/thttpd
8	http://www.microsoft.com/info/cpyright.htm	http://www.adobe.com/products/acrobat/readstep.html
9	http://www.adobe.com/products/acrobat/readstep.html	http://home.netscape.com/
10	http://home.netscape.com/	http://www.thecounter.com/
11	http://www.ibm.com/	http://www.gendex.com/ged2html
12	http://www.worldwidemart.com/scripts	http://www.adobe.com/
13	http://www.acme.com/software/thttpd	http://www.worldwidemart.com/scripts
14	http://search.internet.com/	http://upload.tucows.com/contactus.html
15	http://upload.tucows.com/contactus.html	http://www.w3.org/
16	http://www.thecounter.com/	http://www.listbot.com/
17	http://www.listbot.com/	http://www.tucows.com/privacy.html
18	http://www.w3.org/	http://www.worldwidemart.com/scripts/faq/wwwboard..
19	http://www.adobe.com/	http://www.microsoft.com/windows/ie/default.htm
20	http://www.tucows.com/search.html	http://www.usgs.gov/
21	http://www.tucows.com/privacy.html	http://www.bsdi.com/
22	http://www.gendex.com/ged2html	http://www.rsac.org/
23	http://cbl.leeds.ac.uk/nikos/personal.html	http://search.internet.com/
24	http://www.adobe.com/misc/privacy.html	http://www.nasa.gov/
25	http://www.adobe.com/homepage.html	http://cbl.leeds.ac.uk/nikos/personal.html

Table 1: The old and new top-25 list of W

Figure 7: Top 25 urls in W before and after adaptive resetting

7 Future Work

The correctness of the adaptive resetting algorithm is yet to be proved. Application of this technique to problems like “Google Bombing” is also to be tested. This is when users get links to their sites included into prominent web sites. This can be seen in the blog community where users send “empty” or irrelevant notes with links to their sites as comments to the entries of well known bloggers.

References

- [1] Hui Zhang 0002, Ashish Goel, Ramesh Govindan, Kahn Mason, and Benjamin Van Roy. Making eigenvector-based reputation systems robust to collusion. In *WAW*, pages 92–104, 2004.
- [2] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

- [3] Uriel Feige, David Peleg, and Guy Kortsarz. The dense -subgraph problem. *Algorithmica*, 29(3):410–421, 2001.
- [4] Ari Juels and Marcus Peinado. Hiding cliques for cryptographic security. In *SODA*, pages 678–684, 1998.