Extracting Link Spam using Biased Random Walks From Spam Seed Sets

Baoning Wu
Dept of Computer Science & Engineering
Lehigh University
Bethlehem, PA 18015 USA
baw4@cse.lehigh.edu

Kumar Chellapilla
Microsoft Live Labs
One Microsoft Way
Redmond, WA 98052 USA
kumarc@microsoft.com

ABSTRACT

Link spam deliberately manipulates hyperlinks between web pages in order to unduly boost the search engine ranking of one or more target pages. Link based ranking algorithms such as PageRank, HITS, and other derivatives are especially vulnerable to link spam. Link farms and link exchanges are two common instances of link spam that produce spam communities – i.e., clusters in the web graph. In this paper, we present a directed approach to extracting link spam communities when given one or more members of the community. In contrast to previous completely automated approaches to finding link spam, our method is specifically designed to be used interactively. Our approach starts with a small spam seed set provided by the user and simulates a random walk on the web graph. The random walk is biased to explore the local neighborhood around the seed set through the use of decay probabilities. Truncation is used to retain only the most frequently visited nodes. After termination, the nodes are sorted in decreasing order of their final probabilities and presented to the user. Experiments using manually labeled link spam data sets and random walks from a single seed domain show that the approach achieves over 95.12% precision in extracting large link farms and 80.46% precision in extracting link exchange centroids.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Performance

Keywords

Search engine, web spam, link spam, random walks, seed sets

1. INTRODUCTION

Web spam comprises web pages that have been manipulated in ways to achieve higher ranking in search engine results than they deserve. This manipulation can be broadly classified into two types: content manipulation and link structure manipulation. Content manipulation is completely under the control of the web page authors and as a consequence is easy. Early search engines relied mostly on the classic vector space model [3] of information retrieval and did not use link-based ranking. As a result, content spam appeared as early as 1996, soon after the advent of successful search engines [1, 2]. Some content manipulation techniques such as meta tag stuffing and keyword stuffing [4] were very effective, especially when combined with text hiding techniques.

The advent of link based ranking algorithms such as HITS [5] and PageRank [6,7] significantly reduced the effectiveness of content spam. Unlike altering web page content, acquiring incoming links from reputed sites with high rank was much more difficult. Link based ranking algorithms were also more successful at recognizing popular web sites. While content spam became less effective, link spam became more prevalent. Link spam deliberately manipulates hyperlinks between web pages in order to unduly boost the search engine ranking of one or more target pages. Since PageRank, HITS, and other derivatives value hyperlinks more than page content, they are especially vulnerable to link spam. The ever increasing popularity of shared authorship of web pages on the internet has reduced the barrier to generating link spam. Common examples of shared online authorship include blog pages, user reviews and comments pages, visitor and guest book pages, etc.

In this paper, we propose utilizing an automated random model to detect link farms or link exchanges when given some spam seed domains. The motivation for this work is that usually it is easy to recognize a few sites (using an automated or manual process) that are joining link farms, but to enumerate or list all members from the same link farm or link exchange communities is nontrivial. The goal of our work is to provide a tool for search engine experts to automatically expand their spam blacklist when they have founded a few spam sites.

2. Background and Related Work

Link farms and link exchanges are two common instances of link spam [8-11]. In general, link farms are made up of sites or pages from the same owner, while link exchanges can be the joint collaboration between different content providers.

2.1 Link Exchanges

Link exchange or *reciprocal link exchange* is a practice of exchanging links with other websites. Both participating web sites agree to link to each other. Several methods exist for arranging a link exchange between webmasters. One common way is to show interest in exchanging links explicitly on the web pages. Another hidden method is to email another website owner and ask for a link exchange. A webmaster can also request for a link exchange through any of several webmaster discussion boards based on a

AIRWeb'07, May 8, 2007, Banff, Alberta, Canada.

specific topic/category or make the invitation even open to anybody.

One trick that some webmasters play is that they announce that they will only accept link exchanges from sites that are topically related, and not entertain link exchange requests from topically unrelated sites. Such sites are sill participating in link exchanges and, in some cases, such web sites end up dominating the top search ranks for queries related to the topic.

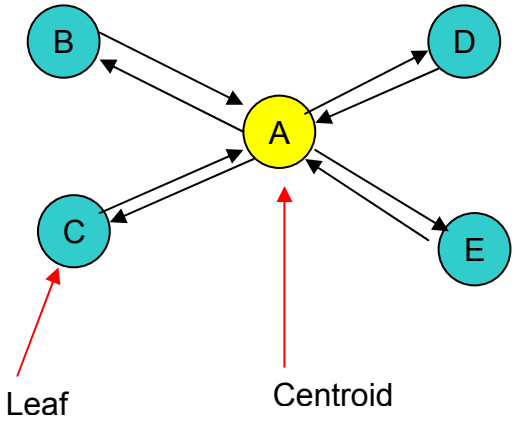

Figure 1. An example link exchange community, wherein node A is the centroid, and nodes {B, C, D, E} are the leaves.

Figure 1 shows an example star style graph for a link exchange. Node A is the initiator of the link exchange. Each of the other nodes, namely, B, C, D, and E have reciprocal links with node A. Node A is designated as the *centroid* of the link exchange community, while nodes B, C, D, and E are referred to as the *leaves* in the link exchange community.

When nodes participate in many link-exchange communities, a new, big link community will be generated, as depicted in Figure 2.

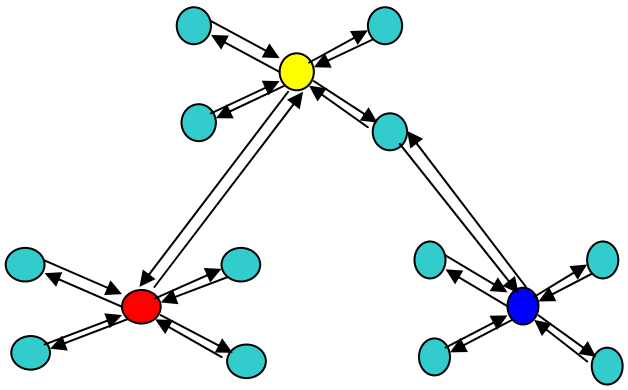

Figure 2. A community of link exchange communities.

2.2 Link Farms

While link exchange systems are designed to allow individual websites to selectively exchange links with other relevant websites, link farms comprise a group of web pages that all hyperlink to many/all other pages in the group. Owing to the size of these groups, most link farms are created through automated programs and services. Figure 3 presents an example of a small link farm. Nodes A, B, and C are densely connected to form a link farm.

Carefully devised link exchanges and link farms and alliances between multiple link farms and link exchanges can be reciprocally advantageous to all participants [9,12]. In this paper, pages that spammers wish to boost are called target pages. For any link farm and any target set of pages, wherein each target page is pointed to by at least one link farm page, the sum of PageRank scores over the target set's nodes is at least as large as a linear function of the number of pages in the link farm [7].

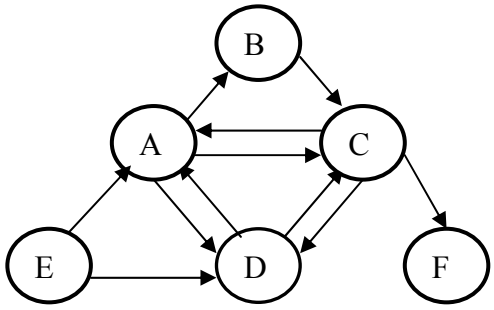

Figure 3. An example link farm. Nodes A, C, and D form the link farm.

2.3 Seed Set Expansion

The link structure of the Web automatically lends itself to seed set expansion. For example, the first step of the famous HITS algorithm uses a search engine to generate a small seed set of results which is then grown using a fixed-depth neighborhood expansion. HITS is then run on the enlarged seed set. Such seed set expansion is also broadly used in local link-based analysis, especially in community analysis [13,14], link based ranking algorithms [5,6], and classification [15]. Seed set expansion has also been used to some extent in identifying link spam. For example, Wu and Davison [27] identify a seed set of link farm pages based on the observation that the in- and out-neighborhood of link farm pages tend to overlap. Then the seed set of bad pages is iteratively extended to other pages which link to many bad pages. They neutralize the link spam by dropping links between link farm pages.

From a graph theoretic perspective, local link-analysis algorithms attempt to find cuts of small conductance [16,17] within a carefully expanded neighborhood of the seed vertex. As part of their work on graph partitioning and graph sparsification [16], Spielman and Teng present a method for finding cuts based on the mixing of a random walk starting from a single vertex. The mixing random walk is used as a subroutine to produce balanced separators and multiway partitions. Andersen and Lang [17] investigated the possibility of using the basic random walk model to detect communities on the web. Motivated by the above ideas, we propose using a random walk model to detect link spam. Compared to Andersen and Lang [17], our method requires only a few seeds to be used to extract link spam communities. Both link exchanges and link farms produce spam communities – i.e., clusters in the web graph with small conductance. Link spam generates a problem by creating whole communities of web pages. In this paper, we use random walks to extract link spam communities.

2.4 Detecting Link Spam

Link spamming is important not only because it is very effective but also because it is much harder to detect than content based spam both by humans and automated algorithms. Even when equipped with knowledge of some members of a link farm,

human judges still need to spend a nontrivial amount of time to extract their partners from the web. The difficulty stems from the fact that one needs to examine the link structure and assess whether back-linking patterns are natural or artificially generated to boost search engine rankings. False positives result often. For example, link spamming tends to be a common side-effect for affiliate pages as their business model is based on traffic redirection from other sites. Detailed analyses of how PageRank scores can be manipulated by link spam are presented in [9,10].

Traditional link spam detection algorithms [18-23] have adopted a fully automatic approach that does not require human input. PageRank based schemes such as SpamRank [18], TrustRank [19], Topical TrustRank [20], Anti-Trust Rank [21], HostRank [22], BadRank [23] etc. have been proposed. Statistical approaches based on machine learning that detect link spam by finding missing statistical features have also received much attention [24]. Decision trees trained on a broad set of features have been used to distinguish navigational and link-spam (dubbed as nepotistic) links from good links [25]. Linkage pattern based features [26] combined with web page features such as host names, IP addresses, in- and out-degree, page count, and rate of change [24] have also been shown to be useful in detecting web spam.

The strength of fully automated techniques lies in their scalability to the whole web. However, they can suffer from low precision at moderate to high recall. This is partly because of the inherent ambiguity in determining whether a community is spam or not. There can be disagreement even among different human judges. This is further exacerbated by the dynamic nature of web spam that results in a constant arms race between a search engine's efforts to reduce web spam and the spammers adapting their techniques to subvert the solution. Further, trust based ranking algorithms may need to solve the problem of generating the optimal trusted seed set and machine learning based algorithms need to pay the cost of generating many possibly expensive features.

For practical use, even the best web spam detection algorithms generate a non-trivial number of false positives and false negatives. False positives are much more damaging than false negatives. Commercial search engines employ manual effort to quickly identify these false positives and correct them appropriately. Tools that can make the manual post-processing more efficient and allow humans to scale their efforts are in great demand.

In this paper, we present a directed approach to extracting link spam communities when given one or more members of the community. In contrast to previous completely automated approaches to finding link spam, our approach is specifically designed to be used interactively. In many cases, our approach can be used as a post-processing step to resolve ambiguous spam communities. The proposed approach starts with a small spam seed set provided by the user (or an automated algorithm scrubbed by a human) and simulates a random walk on the web graph. The random walk is biased to explore the local neighborhood around the seed set through the use of decay probabilities. Truncation is used to retain only the most frequently visited nodes. After termination, the nodes are sorted in decreasing order of their final probabilities and presented to the user. With the proposed tool in hand, human judges need only make decisions at the spam community level. So, their involvement can be limited and human input can be scaled by several orders of magnitude.

3. METHOD

Given a seed set containing one link spam seed and the web graph, a biased random walk is applied to extract other members within the same community as the seed domain or page.

3.1 Random Walk Model

The basic random walk model is quite simple and is similar to those presented in [16] and [17]. Consider a graph $G = \{V, E\}$ with $n = |V|$ nodes. Let A denote the adjacency matrix of G, and let D be the diagonal matrix where $D_{ii} = d(v_i)$, the degree of the i-th vertex. Let S represent the seed set and $s = |S|$ represents the seed set size[1]. The random walk begins with an initial probability distribution p_0 given by

$$p_0(i) = \begin{cases} 1/|S| & \text{if } i \in S, \\ 0 & \text{otherwise} \end{cases}$$

Only the seed node(s) have non-zero probabilities. Then we iteratively update the probabilities as the random walk progresses, using

$$p^{t+1} = \frac{1}{2}\left(I + AD^{-1}\right)p^t$$

The above random walk model simulates the following random web surfer behavior. The user starts from one of the seed nodes and at each iteration

1. with 0.5 probability stays at the current node, and
2. with 0.5 probability jumps to one of the child nodes[2] with equal probability

Note that the model is also equivalent to the user starting with a seed node and at each iteration

1. with 0.5 probability stays at the current node, and
2. with 0.5 probability jumps to one of the non-zero probability nodes with probability proportional to their current value.

Intuitively, the nodes within the same community as the seed set will get higher probability values after a few iterations because these nodes are closer to the seed nodes and are also better connected to other nodes within the same community. Thus, a random surfer will jump to them with a greater chance. While the nodes that are not within the spam community will finally get poor probability values because a random walker will jump to them from fewer nodes. If iterated for a long time, for a connected graph the probabilities will asymptotically converge to the first Eigenvector of the transition probability matrix, given by

[1] All experiments in this paper use a seed set of size 1. However, the proposed approach is general and applicable to seed sets of any size.

[2] In directed web graphs, jumping to the child nodes corresponds to clicking on one of the out-links, while in undirected graphs this corresponds to jumping to either an out-link or an in-link.

$$T = \frac{1}{2}\left(I + AD^{-1}\right)$$

The formulation is similar to several PageRank style random walks over the entire web graph. However, there are several key differences. Firstly, there is no uniform jump vector. Secondly, we are interested in the transient behavior of the random walk and not the asymptotic converged probabilities. Further, several modifications are made when this random walk procedure is implemented in practice (see Section 3.2), that change the dynamics of the random walk.

In the transient phase, the node probabilities are good indicators of whether a node belongs to the same spam community as the seed set or not. Nodes with high probability are more likely to be part of the spam community than nodes with low probabilities. Nodes with zero probability are either not part of the spam community or they have not yet been discovered.

The random walk model can be modified by changing the composition of A in the formula in Section 3.1. By generalizing A from a simple adjacency matrix to a weighted matrix, one can envision incorporating extra information about the nodes and edges in the graph to guide the random walk. The random walk follows outgoing edges from a given node with probability proportional to the edge weight. Examples of useful information include, but are not limited to, node weights based on content spam classifier outputs, edge weights based on topic similarity between pairs of pages, node and edge weights based on user traffic, clicks, dwell-time, etc.

3.2 A Practical Random Walk Model

Naively applying the above random walk model as is to a web graph leads to several practical problems. For example, most web graphs have low diameter and small mean pair-wise distances. So, the number of nodes with non-zero probability grows very quickly. For example, for the web domain graph used in this paper, the diameter is about 3, and a BFS of 3 steps on average extracts almost 30% of the whole graph. This is undesirable both from a computational perspective and the quality of results perspective. If unchecked, the computation degenerates to a PageRank style computation[3] over the whole web graph and the resulting community will have little to do with the seed set. In this paper, following methods similar to those in [17], the random walk model was changed as follows.

3.2.1 Truncation

In order to improve the performance of the computation and also bias the random walk towards more promising nodes, a truncation step is added to the end of each iteration. The truncation procedure prunes some nodes (sets their probability to zero) from the tail of the sorted list of probabilities. Pruning can be done in two ways. One can pick a fixed threshold and remove all nodes with a probability value below the threshold. One can also choose to drop nodes in the bottom k-percentile of the probability distribution. We chose the latter approach, as it is more dynamic and adapts to communities of different sizes.

3.2.2 Renormalizing the Probabilities

In any web graph, leaf nodes (nodes with no children) will leak probability at each iteration. The truncation step also results in a probability leak from the nodes that were pruned. To compensate for this, at the end of each iteration, the probabilities are renormalized to sum to one.

3.2.3 White list of Good Domains

Random walks from spam seeds often lead to reputed domains that are well connected in the web. Good domains, such as yahoo.com or dmoz.org, often have a large fanout and point to lots of other domains on the web. This results in an explosive growth in the size of the candidate set every time the random walk encounters a reputed domain. The good domains eventually dominate the random walk resulting in community drift. In order to address this problem, we use a white list of known good domains. The random walk is modified to not follow any links to white listed domains. This assumption is reasonable because we expand from spam seed sets and reputed well-known domains are very unlikely to join these link farms or link exchange communities.

3.2.4 Decayed Random Walk

Since the members of a link farm or link exchange are expected to have short distances from the seed set, it makes sense that we give larger weights to the nearby nodes than nodes that are far away from the seed set. We propose using a decay to constrain the random walk from wandering too far away from the seed set. This is implemented through a probability adjustment step before the truncation step. The probability adjustment step decays each non-zero probability value by an exponential factor based on the distance of the node to the seed nodes, as follows:

$$p^t[i] = p^t[i] \times \gamma[i]$$

$$\gamma[i] = 2^{-\delta(i)}$$

where $\delta(i)$ is the distance of node i to the seed set. For weighted graphs, one can extend this distance to be the sum of the edge weights along the shortest path. It is also common to truncate the decay after a certain distance, i.e., set $\gamma(i) = 0$, whenever $\delta(i) > \delta_{max}$.

4. EXPERIMENTS

The proposed random walk model was tested on the domain graph from July 2006 obtained by processing crawl data from Live Search[4].

4.1 Datasets

4.1.1 Web Graph

The domain level graph, $G_D = \{V_D, E_D\}$, contained $|V_D|$ = 47.8 million domains (nodes) and about $|E_D|$ = 470 million directed edges. By making each edge bi-directional to form an undirected graph, we get the number of edges to be 820 million. Each node in the graph was a domain. All hosts for the same domain were collapsed into the same node. Edges between domain nodes

[3] Note that the simulated random walk does not have a jump vector. Thus the random walk is bound by the connected component containing the seed set. However, for most seed sets, on average they belong to the giant component which can be larger than 80% of the web graph.

[4] Live Search: http://www.live.com

represent the existence of at least one hyperlink between the two domains. The edge weights were the number of links between domains. The diameter of the domain graph was about 3 and the average fan out of the nodes was around 10. The largest connected component contained 13.2 million domains. The graph contained 34.1 million isolated domains, i.e., they had no edges. These were registered domains that either were not yet hosted, or did not link to pages outside their domain.

4.1.2 Spam Seeds

We used a two step process to select a set of seeds for our experiments. Following [27], we first generated a list of domains that had at least 30 common incoming and outgoing links in the directed domain graph. Second, we randomly sampled 75 link farm seeds and 50 link exchange seeds from this set. Manual checks were used to ensure that these seeds were from unique link farm or link exchange communities. Note that these 50 link exchange domains are (one of many) centroids in the link exchange community. While labeling the link farms, each of the link farms was also classified into small or big based on the number of nodes participating in the link farm. Among the labeled set of link farm seeds, 46 were big link farms, with at least 50 members, and 27 were small link farms, which contains less than 50 members. On average, the small link farms contained only 10-20 members. The remaining two seed domains were link farms made up of web blogs. Overall, for all experiments in this paper, the spam seed set size was 1, i.e., it contained only a single seed domain.

4.1.3 White List

A list of 25,667 good non-spam domains was used as a white list. Note that these made up less than 0.05% of the set of all domains in the domain graph.

4.1.4 Practical Random Walk Parameters

The random walk was run for 30 iterations starting from each of the 75 link farm seeds. We used the decayed random walk model and dropped the bottom 15 percentile during truncation. We choose the output from step 30. More intelligent ways of picking the best step to output the community are discussed and evaluated in [17]. However, in the interests of simplicity, we opted to choose a fixed iteration to output the extracted link farm. We did not tune either of these numbers. Real world systems should tune them for their intended use and improved performance in their problem domain.

4.1.5 Directed, Inverted, and Undirected Walks

The original domain graph contained directed edges. From the directed graph we computed both the inverted and undirected versions of the domain graph. The inverted graph was obtained by reversing the direction of each edge. The adjacency matrix for the inverted graph is obtained by simply transposing the original adjacency matrix. The inverted graph has been favored in previous link spam detection experiments [27]. The adjacency matrix for the undirected domain graph was obtained by taking the mean of the original adjacency matrix with its transpose.

4.1.6 Weighted Domain Graph

Usually, there is more than one link between domains. We can get a weighted domain graph by using the number of links as the weight.

5. RESULTS

We ran experiments using the directed, inverted, and undirected versions of the domain graph. Experiments were also conducted with weighted and unweighted versions of the graph. Only edge level weights (representing the number of edges between domains) were used in the experiments. Node level weights were not used in the experiments reported in this paper[5]. There were marginal differences between the performance, with all three versions producing roughly similar results. This is somewhat expected since strong link farms and link exchanges tend to have a symmetric structure. We often found that both the parents and the children of a seed node were members of the same spam community. The simulated random walks are very fast. Our test implementation took between 1 and 2 minutes for extracting both link farm and link exchange seed sets. We expect optimized implementations to be much faster. However, one consistent but relatively small difference was in the size of the spam communities extracted by the different random walk versions. In the domain graph, there were many more edges with small weights than large weights. As a result, all three weighted versions produced communities that were smaller than their unweighted counterparts.

Random walks on the undirected domain graph produced slightly larger communities than their directed and inverted versions. This is also expected since the undirected graph has roughly twice as many edges as the directed or inverted graphs. Further, for a fixed truncation percentile threshold, spam communities found using random walks on the undirected graph grew faster over successive iterations than on the directed or inverted graph. This is explained by the average fan-out for each node in the undirected graph being almost twice that in the directed and inverted versions.

The extracted spam communities were manually evaluated. Labeling a domain as part of a link farm or link exchange is a time consuming task. Unlike labeling individual web pages, to manually determine whether a given domain participates in a link exchange or not, one has to check several tens of pages in the domain. If one or more pages in the domain participate in a link exchange, the domain is marked as participating in a link exchange. One thing to note here is that we only mark the centroids of link exchange communities as link exchange nodes (see Section 5.2 for details). Link farms are even more time consuming to label, as one needs to find several colluding domains for each candidate link farm member domain being evaluated.

In view of the above, we report manual evaluation results on only the undirected and unweighted domain graph. Overall we manually evaluated about 4000 domains. We expect the other random walk variants to be of similar quality with maybe a slightly lower recall and slightly higher precision.

5.1 Link Farms

Each link farm community obtained using the random walk was evaluated as follows. The domains in the community were first ranked in decreasing order based on their final probabilities. They were then segmented into ten buckets of equal size. Bucket 1 contains the top 10 percentile nodes with highest probability values and bucket 10 contains the bottom 10 percentile nodes

[5] We plan to pursue using content based spam classifier outputs as node weights in future work.

with lowest probability values. Three domains were randomly chosen from each of the ten buckets and manually checked to determine whether the domain belonged to the same link farm as the seed set or not. Hence, for each seed set, we manually checked 30 domains from the extracted link farm community. In total, we checked about 2250 domains for link farm seeds.

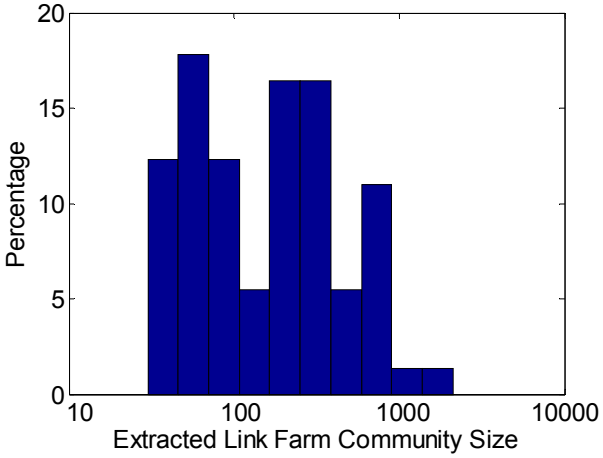

Figure 4. Histogram of the size of link farm communities extracted using the decayed random walk from a single link farm seed domain on the undirected and unweighted version of the domain graph. The mean of the histogram is 268.04 domains.

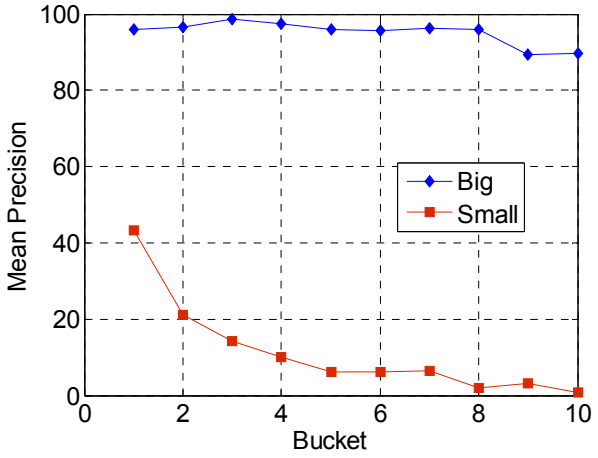

Figure 5. Mean precision curve for link farm communities extracted using random walks on the undirected domain graph from 73 seed sets. 46 seeds were from big link farms with over 50 nodes. 27 seeds were from small link farms with less than 50 nodes. On average the small link farms only had about 10-20 nodes. Bucket 1 contains nodes with probabilities in the top 10 percentile, while bucket 10 contains nodes with probabilities in the bottom 10 percentile.

Figure 4 presents a histogram of the size of the extracted link farm communities using the decayed random walk on the undirected and unweighted version of the domain graph. Note that each of the seed sets contained only a single domain. The number of domains in the extracted link farm community ranged from a few tens to several thousand with the mean being 268.04 domains.

Figure 5 presents the mean precision curves for 73 seeds comprising seeds from 46 big and 27 small link farms. Here precision refers to the percentage of sites marked by our random walk model that are true link farm nodes. The two blog seeds failed to generate reasonable results. One possible reason is that both of these blogs had many outgoing links, which caused a community drift making the real link farm invisible. The random walk model does remarkably well for large link farms with over 90% precision across almost all 10 buckets. The mean precision was 95.12% for large link farms over all 10 buckets. This suggests that starting from only one spam seed, we can identify several hundreds of its partners with more than 95% precision. For the two blog seeds, we observed a quick community drift and the resulting results were not as good. Blog seeds require a modified version of the algorithm that is robust to such community drift.

For small link farms, the trend is obvious in that nodes with higher probability are more likely to belong to the same link farm as the seed node. However, after the first couple of buckets, the false positives quickly overwhelm true positives. Manual inspection of the results in the first bucket showed that most of the nodes with high probability were from the same link farm as the seed. In general, random walk models are known to experience community drift with small link farm seeds [17]. One can potentially counter this by using a truncated decay (δ_{max}).

5.2 Link Exchanges

We followed a similar procedure for evaluating communities extracted from link exchange seeds. The nodes in the communities were sorted, bucketed, and sampled for evaluation. In total, 1500 link exchange domains were manually evaluated.

In comparison with the evaluation of link farms, the link exchange evaluation procedure was much more stringent. Only domains that were centroids/hubs of the link exchange were marked as belonging to the link exchange spam community. Leaf nodes of a link exchange community were considered to be false positives. There are two reasons for this choice. Primarily, only the centroid/hub domains of a link exchange community can be manually labeled reliably correctly. Our approach during labeling involved looking for one or more pages in the domain that contain an explicit invitation to cross link with the promise to link back. Secondly, the easiest way to neutralize a link exchange community is to neutralize/demote the hubs in the link exchange community. Thus identifying the centroids/hubs of a link exchange community is of most interest when trying to detect link exchanges.

Figure 6 presents a histogram of the size of the extracted link exchange communities using the decayed random walk on the undirected and unweighted version of the domain graph. Note that each of seed sets contained only a single link exchange centroid node. The number of domains in the extracted link exchange community ranged from a few tens to several thousand with the mean being 513.5 domains.

Figure 7 presents the mean precision curve for link exchange communities extracted from 50 seeds. The random walk model does quite well with precision values above 80% in the top half buckets and above 70% in the bottom half buckets. The mean precision was 80.46% over the 10 buckets. As we have mentioned before, we only mark centroids of link exchange communities.

Hence, higher precision would be obtained if leaf nodes of the link exchange communities were also included.

6. DISCUSSION

Stronger parametric flow methods do exist for finding low-conductance cuts within an expanded neighborhood of the seed set. However, the random walk-based method used in this paper offer a weaker spectral-style guarantee on conductance. At the same time, these guarantees are counterbalanced by a valuable locality property which ensures that we output a community consisting of nodes that are closely related to the seed set. Further improvements on the quality of the results can be obtained by cleaning up the walk-based cuts with a conservative use of flow that does not disturb this locality property very much [17].

In contrast to [17], wherein reliable extraction of communities required a large fraction (over 20%) of the target community be present in the provided seed set, the experiments in this paper clearly demonstrate that even a single seed is sufficient for extracting participating members from the link farm/exchange. An iterative process can be used to gradually grow the link spam seed set. Conducting a sequence of random walks with seed sets augmented with extracted results in the previous iteration might produce a larger number of results. However, preliminary experiments (not reported here) did not produce much improvement.

It is worth mentioning that the random walk models presented in this paper are still somewhat irreducible. This is definitely the case if δ_{max} is very large. As a consequence, under certain pathological cases, significant community drift is possible with the seed set completely missing from the extracted community. This occurs when the provided seed is not part of a tightly linked (spam) community, or is heavily dominated by another nearby (within δ_{max}) community. In these cases, the probability of losing the seed set grows exponentially with the number of iterations.

7. CONCLUSION

In this paper, we investigated the performance of random walk models for extracting link farms and link exchanges communities based on the known link farm seeds.

We presented a directed approach to extracting link spam communities when given one or more members of the community. In contrast to previous completely automated approaches to finding link spam, our method is specifically designed to be used interactively. Our approach starts with a small spam seed set provided by the user and simulates a random walk on the web graph. The random walk is biased to explore the local neighborhood around the seed set through the use of decay probabilities. Truncation is used to retain only the most frequently visited nodes. After termination, the nodes are sorted in decreasing order of their final probabilities and presented to the user. Experiments using manually labeled link spam data sets and random walks from a single seed domain showed that the approach achieves over 95.12% precision in extracting large link farms and 80.46% precision in extracting link exchange centroids.

Given the high precision for big link farms and link exchange, random walk models seem to be a promising direction for detecting spam communities based on the known spam domains. We plan to explore combinations of page/node level features with local random walks to improve detection of more advanced collusion strategies that employ both content and link spamming techniques. We also plan to explore the performance of these algorithms on a much larger scale using page-level web graphs.

8. ACKNOWLEDGMENTS

We would like to thank Chau Luu and Naoko Takanashi for help with labeling the extracted link farms and link exchanges. We would also like to thank the anonymous reviewers for providing valuable feedback.

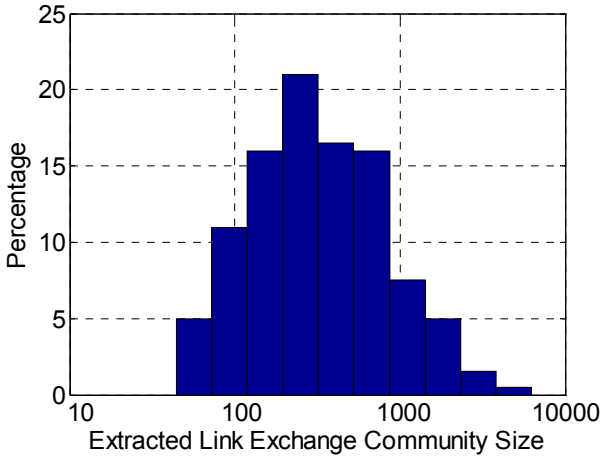

Figure 6. Histogram of the size of link exchange communities extracted using the decayed random walk from a single link exchange domain on the undirected and unweighted version of the domain graph. The histogram mean is 513.5 domains.

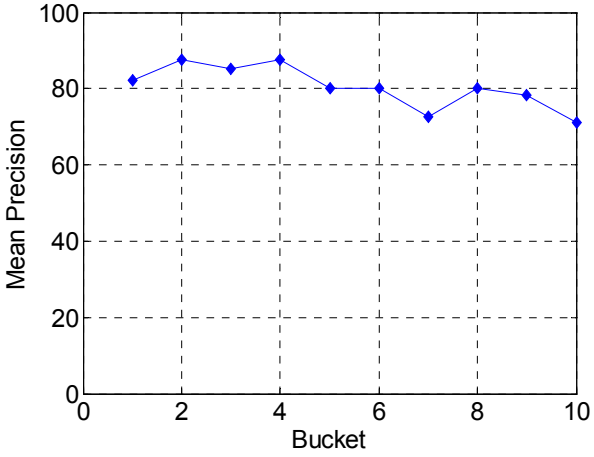

Figure 7. Mean precision curve for link exchange communities extracted using random walks on the undirected domain graph from 50 seed sets. Bucket 1 contains nodes with probabilities in the top 10 percentile, while bucket 10 contains nodes with probabilities in the bottom 10 percentile.

9. REFERENCES

[1] C. Chekuri, M. H. Goldwasser, P. Raghavan, and E. Upfal. "Web search using automatic classification." In Proceedings

of the 6th International World Wide Web Conference (WWW), San Jose, US, 1997.

[2] The Word Spy - Spamdexing. http://www.wordspy.com/words/spamdexing.asp.

[3] R. Baeza-Yates and B. Ribeiro-Neto, "Modern Information Retrieval." Addison Wesley, 1999.

[4] http://en.wikipedia.org/wiki/Spamdexing

[5] J. M. Kleinberg. "Authoritative sources in a hyperlinked environment." Journal of the ACM, 46(5):604-632, 1999.

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

[7] M. Bianchini, M. Gori, and F. Scarselli. "Inside PageRank." ACM Transactions on Internet Technology, 5(1), 2005.

[8] Z. Gyöngyi and H. Garcia-Molina. "Web spam taxonomy." In Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb), 2005.

[9] Z. Gyöngyi, H. Garcia-Molina. "Link Spam Alliances." In Proceedings of the 31st International Conference on Very Large Data Bases (VLDB), Trondheim, Norway, 2005.

[10] Y. Du, Y. Shi and X. Zhao. "Using Spam Farm to Boost PageRank." Online at http://www.eecs.umich.edu/~duye/

[11] H. Zhang, A. Goel, R. Govindan, K. Mason, and B. V. Roy. "Making eigenvector-based reputation systems robust to collusion." In Proceedings of the 3rd Workshop on Algorithms and Models for the Web-Graph (WAW), Rome, Italy, October 2004. Full version to appear in Internet Mathematics.

[12] R. Baeza-Yates, C. Castillo, and V. López. "PageRank increase under different collusion topologies." In Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb), 2005.

[13] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. "Trawling the Web for emerging cyber-communities." Computer Networks, 31(11{16):1481{1493, 1999.

[14] G. Flake, S. Lawrence, and C. Lee Giles. "Efficient identification of web communities." In Sixth ACM SIGKDD, pages 150{160, Boston, MA, August 20-23 2000.

[15] S. Chakrabarti, B. E. Dom, and P. Indyk, "Enhanced hypertext categorization using hyperlinks." In Proceedings of ACM SIGMOD-98, pages 307-318, Seattle, US, 1998. ACM Press, New York, US.

[16] D. A. Spielman and S. Teng. "Nearly-linear time algorithms for graph partitioning, graph sparsification and solving linear systems," In ACM STOC-04.

[17] R. Andersen and K. J. Lang. "Communities from seed sets." In Proceedings of the 15th International World Wide Web Conference (WWW), 2006.

[18] A. A. Benczur, K. Csalogany, T. Sarlos, and M. Uher. "SpamRank - Fully automatic link spam detection." In Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb), 2005.

[19] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. "Combating web spam with TrustRank." In Proceedings of the 30th International Conference on Very Large Data Bases (VLDB), Toronto, Canada, 2004.

[20] B. Wu, V. Goel, and B.D. Davison, "Topical TrustRank: Using topicality to combat web spam." In Proceedings of the 15th International World Wide Web Conference (WWW). Edinburgh, Scotland, 2006

[21] R. Raj, V. Krishnan. "Web Spam Detection with Anti-Trust Rank." Second International Workshop on Adversarial Information Retrieval on the Web (At the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval).

[22] N. Eiron, K. S. McCurley, and J. A. Tomlin. "Ranking the web frontier." In Proceedings of the 13th International World Wide Web Conference (WWW), pages 309–318, New York, NY, USA, 2004. ACM Press.

[23] BadRank as the opposite of PageRank. http://en.pr10.info/pagerank0-badrank/.

[24] D. Fetterly, M. Manasse, and M. Najork. "Spam, damn spam, and statistics – Using statistical analysis to locate spam web pages." In Proceedings of the 7th International Workshop on the Web and Databases (WebDB), Paris, France, 2004.

[25] B. D. Davison. "Recognizing nepotistic links on the web." In AAAI-2000 Workshop on Artificial Intelligence for Web Search, Austin, TX, pages 23–28, July 30 2000.

[26] E. Amitay, D. Carmel, A. Darlow, R. Lempel, and A. Soffer. "The Connectivity Sonar: Detecting site functionality by structural patterns." In Proceedings of the 14th ACM Conference on Hypertext and Hypermedia (HT), Nottingham, United Kingdom, August 26-30 2003.

[27] B. Wu and B. D. Davison. "Identifying link farm pages." In Proceedings of the 14th International World Wide Web Conference (WWW), 2005.