Discovering Latent Graphs with Positive and Negative Links to Eliminate Spam in Adversarial Information Retrieval

Ryan Anthony Rossi†

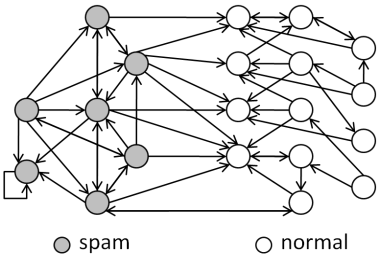
Jet Propulsion Laboratory, California Institute of Technology,
ryan.a.rossi@jpl.nasa.gov

Abstract. This paper proposes a new direction in Adversarial Information Retrieval through automatically ranking links. We use techniques based on Latent Semantic Analysis to define a novel algorithm to eliminate spam sites. Our model automatically creates, suppresses, and reinforces links. Using an appropriately weighted graph spam links are assigned substantially lower weights while links to normal sites are created and reinforced. The empirical validity of these techniques to eliminate and drastically decrease the impact of spam is shown by both our Local Sapling Heuristic and a machine learning algorithm used to classify sites with features derived from our Latent Graph.

Keywords: Adversarial Information Retrieval, Ranking Links, Latent Graph, Link Prediction, Latent Features, Singular Value Decomposition.

1 Introduction

Preventing and eliminating spam is a top challenge for search engines. Web spam is described as any insertion of content or links on a web site that manipulates a search engines results[1]. Adversarial information retrieval on the Web is the study and design of algorithms used to detect spam web sites. In recent years, a substantial research effort has been directed towards discovering a method that efficiently eliminates spam sites.



TrustRank[1] is based on PageRank and uses a set of trusted sites evaluated by humans to propagate the trust to other locally reachable sites. SpamRank[2] measures the amount of undeserved PageRank by analyzing the backlinks of a site. There are other algorithms that try to identify link farms[3], link spam alliances[4] and spam sites using web topology[5]. TrustRank is the most widely known proposed method but suffers from biases where the human selected set of trustworthy sites may favor

† This work is supported by grant ATM-0521002 from the National Science Foundation.

certain communities over others. TrustRank also does not detect spam sites but assigns them a lower PageRank value.

Our paper provides a new direction in Adversarial Information Retrieval by *ranking links* instead of the traditional approaches of ranking sites or using textual features for classification. We use Latent Semantic Analysis and other related techniques to eliminate and lessen the impact of spam links using the web graph. Our model creates, reinforces, and suppresses links. Spam links are given lower weights while normal links are reinforced based on the structure of spam sites and communities. The creation of links reinforces the normal sites increasing accessibility. Surfing [6] from any site (spam or normal) by following the maximum weighted link will bring the user to normal sites. We define a link analysis algorithm called Local Sapling Heuristic to use as a basis to rank spam sites and validate our techniques. Finally using an automated classification algorithm with weighted link-based features derived from our Latent Graph we classify 90.54% of the sites correctly.

2 Local Sapling Heuristic

The PageRank algorithm is a direct application of the Ergodic Theorem and Kirchhoff's Matrix Tree Theorem [6].

We define a new algorithm that uses the local structure around a site to provide a ranking of sites as this proves to be more useful than a global ranking of sites in the light of spam. The justification being that the properties of a site are often correlated with neighboring sites. Furthermore, the local sapling heuristic provides a good indication of authoritativeness[7] while accessing only the local k-neighborhood around a site and is therefore more efficient to compute for large graphs. The Local Sapling Heuristic is defined as follows:

$$r = \sum_{n=1}^{k} \frac{\alpha M^{n}}{n} \tag{1}$$

where M is the adjacency matrix, α is a unit vector and k is the depth of the Local Sapling Heuristic. We typically chose k to be five.

Local Sapling Heuristic has several advantages over HITS[7], PageRank[8] and SALSA[9] such as stability, robust to tightly knit communities, rank sinks, dangling links, and it is very efficient. We take advantage of the heuristic not only to rank sites but also to derive features from our Latent Graph.

3 Latent Semantic Analysis of the Web

In the task of detecting spam sites we have both content features as well as the associated web graph at our disposal. One could apply related techniques described in [10] to extract textual relationships and eventually spam signatures that could be combined with this work to build a more robust spam detection system. In this work

we use only the web graph to extract latent relationships between the sites that inherently allow us to eliminate spam sites and spam communities.

The dataset we used is from the Web Spam Challenge [11] and is considered a benchmark for web spam detection. We start with a web graph. Let G = (V,E) denote a directed graph, where V is the set of web sites and every link (x, y) ∈ E corresponds to a link from site x to site y. There are 9072 sites in our graph where 1934 are spam and 7138 are normal sites. The web graph G is represented as an adjacency matrix M, where $M_{i,j}$ = 1 if there is a link from site i to site j, and $M_{i,j}$ = 0 if there is no link between site i and j.

Let $M \in \Re^{nxm}$, we decompose M into three matrices using Singular Value Decomposition:

$$M = U\,S\,V^T \tag{2}$$

where $U \in \Re^{nxm}$, $S \in \Re^{mxm}$ and $V^T \in \Re^{mxm}$. The matrix S contains the singular values located in the $[i, i]_{1,..,n}$ cells in decreasing order of magnitude and all other cells contain zero. The eigenvectors of MM^T make up the columns of U and the eigenvectors of M^TM make up the columns of V. The matrices U and V are orthogonal, unitary and span vector spaces of dimension n and m, respectively. The inverses of U and V are their transposes.

$$\underbrace{\begin{bmatrix} | & | & & | \\ d_1^h & d_2^h & \cdots & d_k^h \\ | & | & & | \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & s_k \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} — & d_1^a & — \\ — & d_2^a & — \\ & \vdots & \\ — & d_k^a & — \end{bmatrix}}_{V^T}$$

The columns of U are the *principal directions of the hubs* and the rows of V^T are the *principal directions of the authorities*. The principal directions are ordered according to the singular values and therefore according to the importance of their contribution to M. The singular value decomposition is used by setting some singular values to zero, which implies that we approximate the matrix M by a matrix:

$$M_k = U_k\,S_k\,V_k^T \tag{3}$$

A fundamental theorem by Eckart and Young[13] states that M_k is the closest rank-k least squares approximation of M. The error approximating M by M_k is given by

$$\left\|M - M_k\right\|_F = \min_{rank(B)\le k} \left\|M - B\right\|_F = \sqrt{\sigma_{k+1}^2 + .. + \sigma_{rM}^2} \tag{4}$$

The theorem can be used in two ways. To reduce noise by setting insignificant singular values to zero or by setting the majority of the singular values to zero and keeping only the few influential singular values in a manner similar to principal component analysis. In Latent Semantic Analysis we extract information about the relationships between sites as they change when we set all, but the most significant, singular values to zero. The singular values in S provide contribution scores for the principal directions in U and V^T.

We use the terminology “principal direction” for the following reason. In zoomed clusters [15] it was shown that (assuming unit vectors) the principal eigenvector is an ‘iterated centroid’ where outliers are given a decreasing weight.

$$C_{\infty} = \lim_{n \to \infty} \left(M^T M \right)^n e \tag{5}$$

The iterative centroid is the reason Kleinberg’s HITS algorithm favors the most tightly knit communities.

In this work we are only concerned with the web graph as we exploit the nature of LSA to discover latent links. Latent relationships between sites are discovered based on the structure of the normal and spam communities. We will see that over-reducing dimensionality results in a bipartite structure while reducing too little may result in the formation of cliques. Therefore, the values of S represent a measure of tightness or strongly connectedness within a given graph.

4 Ranking of Links

If at the theoretical level some algorithms assume that links are given some weights or probabilities, in practice they are given a uniform probability distribution [1,2,8,9]. We compute the Singular Value Decomposition of the adjacency matrix M from the given graph G and set all but k singular values to zero. We then compute M_k a low rank approximation of M, to which corresponds a new graph G_k. Links in G_k are automatically created, reinforced, and suppressed based on the structure of the graph. It is important to note that we are not following a Markov model now because M_k is not a stochastic matrix, it even has negative numbers. It does not correspond to a Kirchhoff model either as it does not fit a conservation system.

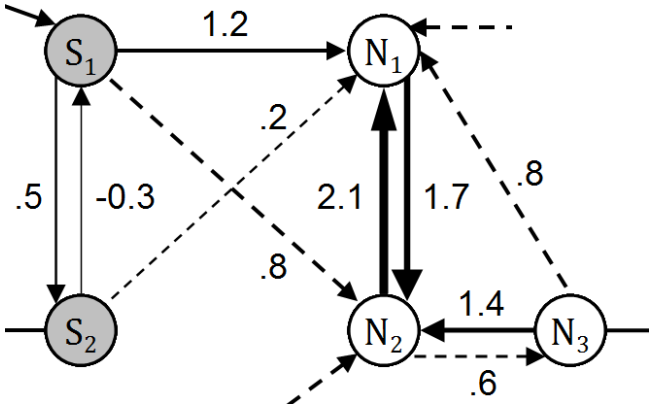


Fig. 1. Automatic ranking of links where latent links are discovered (links are created [dashed arrows], reinforced and suppressed) between the spam sites $\{S_1, S_2\}$ and normal sites $\{N_1, N_2, N_3\}$ based on the fundamental structure of the two communities.

4.1 Maximum Weighted Latent Links

In this section we study the maximum weighted outlinks from the sites in the graphs $\{G_1, G_3, G_5, G_{10}, G_{50}$, and $G_{100}\}$. Informally it means we consider the sites that are the most likely to be accessible in the graph if we follow the ranking of the links. The results provide significant evidence that our Latent Graph strongly favors normal sites while essentially eliminating spam sites and communities through the suppression of their links.

Table 1. We select the latent link that has the maximum weight for all sites in the graphs $\{G_1, G_3, G_5, G_{10}, G_{50}, G_{100}\}$ and count the occurence of the four cases where normal → normal, normal → spam, spam → spam and spam → normal.

Graph	**N** → N	**S** → N	**N** → S	**S** → S
G_1	99.95%	99.80%	0.05%	0.20%
G_3	100%	99.95%	0%	0.05%
G_5	99.87%	99.53%	0.13%	0.47%
G_{10}	97.54%	74.51%	2.46%	25.49%
G_{50}	98.53%	75.18%	1.47%	24.82%
G_{100}	98.75%	73.68%	1.25%	26.32%

For every site we select the outlink with the maximum weight in the Latent graphs. We find that a significant amount of the spam sites have the maximum weighted outlink to themselves creating a sink. A majority of the spam sites maximum weighted outlink are to normal sites. This infers that normal communities are becoming more accessible (or stronger) as spam sites create and reinforce links to normal sites while links between spam sites are suppressed.

As for normal sites we find very few of the maximum weighted outlinks to spam sites. The majority of the normal sites that are directed to spam sites originally have no outlinks or are only pointing to spam sites. If a site has only outlinks to spam sites it should be classified as spam. The human experts may have made a mistake classifying the site. Nevertheless for the majority of normal sites the maximum weighted outlink are to other normal sites. The normal sites are reinforced and form a distinct community. One can see the maximum weighted links in the graphs point to high quality sites as indicated by the Local Sapling Heuristic.

4.2 Spam Ranking: Local Sapling Heuristic

Using our Local Sapling Heuristic paired with the Latent Graph G_5 (where spam links are suppressed) we rank the sites and show the position of the first ten spam sites in the ranking. The first ten spam sites to appear in the ranking of sites are at positions

{2093, 2189, 2623, 3398, 3537, 3615, 3779, 3900, 3901, 3911}. Spam sites are penalized with a substantially lower ranking. Furthermore one can see that a significant amount of spam sites are assigned a negative score.

4.3 Surf Sessions of Spam and Normal Sites

In a surf session we select spam and normal sites at random and walk the links with the maximum weight. We find three cases. In the first table we show a typical surf session starting with a normal site. It is shown that if we follow the maximum weighted link we go to other normal sites. The sites with relatively large link weights indicate very good quality sites or what we call 'metasites' and 'targets' [6] {as an example, 7756 (metasite) → 7710 (target) with weight 1.866}. In the second table it is shown that if we start from a spam site and surf we quickly converge to normal sites. The last case is seen very infrequently. The last table shows if we start from a spam site and follow the maximum weighted link we infrequently go to other spam sites. Furthermore, the link weights in the last case are extremely close to zero indicating the links have been suppressed therefore eliminating these spam sites from the graph. Interestingly surfing could be used to automatically extract a 'high quality' seed set of normal sites for use in TrustRank or other algorithms.

Table 2. A typical surf session starting from a normal site.

Link Weight	$Site_1$ →	$Site_2$
0.327	4000	4712
0.829	4712	7710
1.712	**7710**	**605**
0.259	605	7756
1.866	**7756**	**4985**
0.133	4985	7273
1.535	**7273**	**7756**

Table 3. A typical surf session starting from a spam site.

Link Weight	$Site_1$ →	$Site_2$
0.646	9 **(S)**	1924
1.699	1924	2698
1.721	2698	2698

Table 4. A surf session surf session we have seen less frequently starting from a spam site.

Link Weight	$Site_1$ →	$Site_2$
0.007	1594 **(S)**	411 **(S)**
0.009	411 **(S)**	411 **(S)**

5 Empirical Validation

We use a simple machine learning algorithm called MaxSim to classify sites. The algorithm is conceptually simpler and more efficient alternative to Support Vector Machines for an arbitrary number of classes. It has many attractive theoretical properties regarding underfitting, overfitting, power of generalization, computational complexity and robustness. MaxSim has proven to perform essentially as well as or better than SVM for these types of problems.

We are interested in classifying a site X by comparing its similarity to a set of previously classified training sites. The site X will be assigned to the class (normal, spam) whose sites are most similar to X. Given a set of I class-labeled training sites $\{X_i, \xi(X_i)\}$, i = 1..I, where $\xi(X_i)$ is the class of X_i, and for an unclassified site X, we define the class similarity of X with respect to a class C as

$$S_C(X) = \sum_{X_k \in C} \alpha_k \, s\langle X_k, X \rangle \tag{6}$$

where s is the similarity function and $\alpha_k \geq 0$ reflects the relative importance given to each X_k with respect to the classification. We can therefore predict the class of X using the following decision function:

$$\xi(X) = \arg c \ \{\max(S_C(X))\} \tag{7}$$

5.1 Latent Link-based Features

It is natural to define a measure of flow entering and leaving a site called inflow and outflow, respectively. Furthermore since our weighted links are either positive or negative this definition is extended to measure the positive and negative $inflow^{+\backslash -}$ and $outflow^{+\backslash -}$ of sites. Let (i, j) be a link from S_i to S_j where $\omega^+(i, j)$ is a positive weight and $\omega^-(i, j)$ is a negative weighted link. The $inflow^+$ and $outflow^+$ of a site S are defined respectively as

$$inflow^+(S_j) = \sum_{S_i \in E} \omega^+(i, j) \quad and \quad outflow^+(S_i) = \sum_{S_j \in E} \omega^+(j, i) \tag{8}$$

Similarly $inflow^-$ and $outflow^-$ of a site S are defined respectively as

$$inflow^-(S_j) = \sum_{S_i \in E} \omega^-(i, j) \quad and \quad outflow^-(S_i) = \sum_{S_j \in E} \omega^-(j, i) \tag{9}$$

We derive seven novel features using the notion of flow as a basis {$inflow^+$, $inflow^-$, $outflow^+$, $outflow^-$, *inflow*, *outflow*, and *flow*} where *inflow* is the sum of both the $inflow^+$ and $inflow^-$ of a site and conversely for *outflow*. Therefore we define *flow* as the sum of the *inflow* and *outflow* of a site. We also use the Local Sapling Heuristic as a feature for classification. This heuristic essentially computes the flow of a site from its local neighborhood of a given length.

Using MaxSim with the Radial Basis Function we classify 90.54% of the sites correctly where ($\sigma = 0.015$). The results validate our model for ranking links as well as detecting spam. Furthermore our Latent Graph can be used to derive more sophisticated features to achieve potentially better classification results.

6 Conclusion

We provide a new direction in Adversarial Information Retrieval by ranking links instead of the traditional approaches of ranking sites or using textual based features for classification. Our model automatically suppresses spam links therefore eliminating their influence from the graph while reinforcing and creating links to normal sites making them more accessible. We show the validity of these techniques by ranking spam sites, surfing and using a classification algorithm. We classify 90.54% of the sites correctly with features derived from our Latent Graph.

Acknowledgments. This research was also made with Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a.We thank Jean-Louis Lassez for the many the many insightful discussions.

References

1. Gyongyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with TrustRank. VLDB Conference, 576-587 (2004)
2. Benczur, A., Csalogany, K., Sarlos, T., Uher, M.: SpamRank – Fully Automatic Link Spam Detection. AIRWeb (2005)
3. Wu, B., Davison, B.: Identifying Link Farm Spam Pages. WWW, 820-829 (2005)
4. Gyongyi, Z., Garcia-Molina, H.: Link Spam Alliances. VLDB, 517-528 (2005)
5. Castillo, C., Donato, D. Gionis, A., Murdock, V., Silvestri, F.: Know your neighbors: Web spam detection using the web topology. Yahoo! Research (2006)
6. Lassez, J-L., Rossi, R., Jeev, K.: Ranking Links on the Web: Search and Surf Engines. Lecture Notes of Artificial Intelligence, IEA/AIE, 199-208 (2008)
7. Kleinberg, J.: Authoritative sources in a hyperlinked environment. In Proceedings of the 9th ACM-SIAM, SODA (1998)
8. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web. Stanford Digital Library Technologies Project (1998)
9. Lempel, R., Moran, S.: The stochastic approach for link-structure analysis (SALSA) and the TKC effect. In 9th International WWW Conference, (2000)
10. Lassez, J-L., Rossi, R., Sheel, S., Mukkamala, S.: Signature Based Intrusion Detection System using Latent Semantic Analysis, IJCNN, 1068-1074 (2008)
11. Web Spam Challenge: http://webspam.lip6.fr
12. Berry, M.W., Browne, M.: Understanding Search Engines: Mathematical Modeling and Text Retrieval. SIAM (2005)
13. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika. 1, 211-218 (1936)
14. Deerwester, S., Dumais, S., Landauer, T.K., Furnas, G., Harshman, R.: Indexing by latent semantic analysis. J. Amer. Soc. Info. Sci. 41, 391-407 (1990)

15. Lassez, J-L., Karadeniz, T., Mukkamala, S.: Zoomed Clusters. ICONIP, 824-830 (2006)