Lightweight Clustering Methods for Webspam Demotion

Thomas Largillier
Univ Paris-Sud, LRI;
CNRS, INRIA;
Orsay, F-91405
thomas.largillier@lri.fr

Sylvain Peyronnet
Univ Paris-Sud, LRI;
CNRS, INRIA;
Orsay, F-91405
syp@lri.fr

***Abstract*—To make sure they can quickly respond to a specific query, the main search engines have several mechanisms. One of them consists in ranking web pages according to their importance, regardless of the semantic of the web page. Indeed, relevance to a query is not enough to provide a high quality result, and popularity is used to arbitrate between equally relevant web pages.**

Webspam widely denotes any web page created with the only purpose of fooling ranking algorithms such as the PageRank. The aim of Webspam is to promote a target page by increasing its rank. It is an important issue for Web search engines to spot and discard Webspam to provide their users with a non biased list of results. Webspam techniques have to evolve constantly to remain efficient but most of the time they consist in creating a specific linking architecture around the target page to increase its rank.

In this paper we propose to study the effects of graph clustering on the well known ranking algorithm of Google (the PageRank) in presence of Webspam. Since the web graph is way to big to apply classic clustering techniques, we present three lightweight techniques to realise a clustering of the web graph. Experimental results show the interest of the approach, which is moreover confirmed by statistical evidence.

I. Introduction

Search engines are designed to provide users with results of the finest quality, that is web pages highly relevant to the user's query. To achieve this task, they of course sort web pages with respect to their relevance towards the requests. Unfortunately, this is not enough since malicious webmasters (called spammers from now on) have the ability to index and make web pages relevant to many (i.e. a really huge number of) requests. To avoid strong manipulation over the relevance of web pages, search engines use another metric to rank web pages. This metric is often based on some kind of web page popularity. Popularity together with relevance is used to sort results before presenting them to the users.

Nowadays it is really important for a website to have a good visibility to ensure its popularity and to attract some traffic. Being visible often means to be well ranked regarding requests on web search engines. Without surprise a site appearing on the first two result pages of Google will be more visited than a site on the 100^{th} page for the same request.

It is also important for a website to attract a sufficient amount of traffic since earnings on the Web is often proportional to the number of visitors. This leads to a lot of webmasters, who wants to earn money on the web, taking extra care of their ranking on web search engines.

There are not many actions a webmaster can do to increase its relevance towards requests without quickly falling into spamming. Plus improving its relevance has its limits, once you are relevant towards the requests you're interested in, you will appear into the results list. Then you need to improve your rank in the list.

The popularity is often based on a structural criterion. Webmasters can improve their popularity by making some publicity for their site. It means that they are trying to achieve for other sites to "vote" for theirs in order to improve its popularity. There is a lot of mechanisms that they can use to ensure a high rank to their pages. Many of these mechanisms depend on the targeted ranking algorithms.

As soon as a ranking algorithm is known people will ask themselves how to maximize their score. This is also true for the PageRank. This question has been resolved years ago either for a single page by Gyongyi *et al* in [7] or for a whole website by De Kerchove *et al* in [4].

Orthodox, but borderline, techniques designed to increase the pagerank of some pages on the web are regrouped under the name of *search engine optimization*. Not far from search engines optimizers (SEO) lie Web spammers. They are people whose only goal is to promote a page or a site. Many well-known techniques like link farms are well spread amongst web spammers. These techniques evolve quickly making exact automatic detection hard in practice.

The frontier between SEO and Webspam is thin but creating a whole network of dull pages just to increase a target page or site pagerank can clearly be seen as Webspam.

It is also of the utmost importance for web search engines to detect the cheaters in order to provide their users with a non biased ranking, thus making sure that really popular web sites have a real popularity and not an artificially increased one.

The main goal of this paper is to propose an approach based on graph clustering to demote the effects of web spamming and show its efficiency. We present statistical evidence of its viability at identifying Webspam in an automatic way. Our approach does not need any human assisted step.

This paper is organized as follows, in section II we present work related to ours regarding Webspam detection and demotion. In section III we introduce the lightweight clustering methods we chose to use in order to demote web spam. Section IV shows our experiments and their results. In section V we present statistical evidence of the viability of our approach and finally we conclude in section VI.

II. Related Work

Since the PageRank algorithm was introduced in [13], and became famous through its use within the search engine Google, web spammers tried to figure out how to increase their rank. The question of how to maximise the rank of a target page has been answered in [7] where they also proposed an analysis of the Webspam. If someone wants to maximize the score of an entire web site and not a single web page it has to use a more complicated linking structure between the site pages. The optimal structure is described in details in [4].

With the apparition of Webspam, many techniques were developed to detect or demote the effects of Webspam in order to ensure the user with a fair ranking. These measures can be separated in three categories, demotion, detection and prevention as stated in [10].

A first kind of measures to appear was the propagation of Trust or Distrust proposed by Gyongyi *et al.* in [9][11]. Those methods need a human preprocessing step that help to split a small subset of nodes between good (to be trusted) nodes and bad (not to be trusted) nodes. Then starting from the seeds, either the Trust is propagated or the Anti-Trust is propagated backward.

Wu *et al.* propose in their paper [14] an improvement of the TrustRank algorithm where topicality is considered to increase the results. Their results outperform those of TrustRank but the authors are using a human powered preprocessing step, making the method difficult to use in practice (even harder than the TrustRank).

Gyongyi *et al.* propose an other approach. They present in [8] a framework where the fraction of pagerank coming from spam pages is computed for each web page. This again requires a preprocessing human step where people label pages as spam or nonspam. The estimation of the fraction is calculated by evaluating the pagerank of each page when the source of pagerank is a subgraph composed only of good pages.

Ntoulas *et al.* present in [12] a classifier for webpages based on their content. They chose many criteria going from the number of words in the title to the conditional n-grams likelihood. Their classifier has a high precision and recall but the problem is that the target page of spammers (that is the page whose pagerank is boosted by unorthodox techniques) is often a relevant page. Moreover, using the classifier on every web page is fastidious and not really tractable.

All these methods fall into the scope of spam detection since they attempt to identify Webspam pages in order to reduce their influence. The following methods have a different goal: demoting the effects of web spam.

Andersen *et al.* propose in [1] an algorithm called Robust PageRank. It is designed to fight link spam engineering. They use the supporting sets of nodes (i.e. nodes contributing to the pagerank of a specific web page) regarding the pagerank computation. They locally compute approximate features in order to demote the effects of web spam.

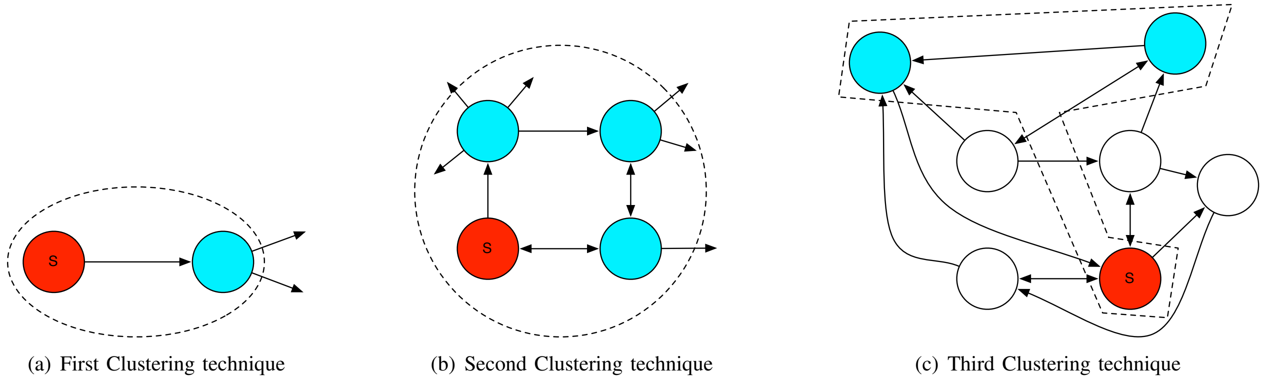
Chung *et al.* (see the paper [3]) have made a study of link farms during a period of two years to see how their distributions and compositions are evolving. They use the Strongly Connected Component (SCC) decomposition to find and study such link farms. Tarjan's algorithm complexity is $\mathcal{O}(n + m)$ where n is the number of nodes of the graph and m the number of edges. This is a good theoretical complexity but still too high to be used to fight Webspam since the algorithm should be iterated to find link farms (the graph is too large to be the input of an algorithm of this complexity). But the idea of clustering the graph to identify link farms is of the utmost interest. It is important to find a faster way to regroup nodes, this is the problem we address in this paper.

III. Clustering Methods

In this section we present the three graph clustering techniques we used in our experiments. First, it is important to notice that classical and efficient clustering methods for small graphs such as *the Markov Cluster Algorithm* (MCL, see [5]) and the *edge betweenness clustering* method (EBC, see [6]) are unsuitable for the web graph because of its size. Indeed MCL requires an explicit matrix representation of the graph which is totally infeasible in our case and EBC runs in time and space of $\mathcal{O}(nm)$ where n is the number of nodes in the graph and m the number of edges (these notations will be kept throughout the paper), which is in practice totally intractable. Moreover we are not interested in an exact clustering. Our interest is not the detection of Webspam but its demotion. If we can group enough Webspam with the target page, building big enough communities, we hope to stop a sufficient amount of incoming pagerank to nullify the Webspam's effects.

Google has indexed more than 1000 billion pages[1]. So every technique must have a low complexity, *i.e* linear at maximum. Indeed the PageRank has a linear complexity $\mathcal{O}(n + m)$. Since the PageRank must be calculated to offer a ranking to users, every method which purpose is to demote the effect of Webspam must add at most a constant amount of calculation to be effective. The ideal case would

[1] http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html



(a) First Clustering technique (b) Second Clustering technique (c) Third Clustering technique

Figure 1. Lightweight clustering techniques

be a method that could be embedded with the PageRank computation with only a constant overhead and no memory usage.

All three methods we present below are local algorithms computed for every node in the graph. The idea is always to have a very simple criterion to group nodes together, starting from a peculiar node. It is also important to use only local knowledge to compute our clusters.

The first technique can be embedded for free during the computation of PageRank. Figure 1 shows example of how nodes are regrouped for each method. The red node labelled S is the starting point of the algorithm and the blue nodes are the ones regrouped with it after the computation.

The first method we propose is very intuitive. We only want to group nodes with one outgoing link with the target of this link (see figure 1(a)). This means that we group people that give all their pagerank to one person with this person. A well-known technique to raise the pagerank of one page is to create a lot of dummy pages that will give all their pagerank to the target page. In the following this method will be referred to as Tech1.

The second technique we want to test regroup nodes that belongs to short loops in the graph. In figure 1(b) the length of the loop is 4. For every node in the graph we compute every path of length k and if the path ends on the starting node then everybody in the loop goes in the same cluster. We know that spammers don't like to waste pagerank, thus many links coming out the target page should return to the target page in a few steps. In the experiments we chose to use a length $k = 3$. In the rest of this paper this method will be called Tech2.

For the last method we simply launch r random walks of length l from every node in the graph. If the number of random walks that ends on a particular node is higher than a threshold t then this node and the starting node are regrouped in the same cluster. With this approach we hope to regroup Webspam with their target page even if some links may lead elsewhere to avoid automatic detection of well known structures. Following links from a Webspam page will lead to the target page with high probability. Later in the article this technique will be named Tech3. During our experiments we launched 200 random walks of length 15. The threshold was fixed at 40 meaning that more than 1/5 of the walks must end on the same node for it to be regrouped with the starting node.

More formally at the beginning of each algorithm every node belongs to its own cluster. When we regroup nodes we simply merge their clusters following the expression “Any friend of yours is a friend of mine”. This has no impact if the starting point of the algorithm has no neighbors in the cluster of the node it wants to regroup with but, on the other hand if it wants to regroup with someone who is very close to one of its successors the starting node probably wants to associate itself with that particular neighbor. Thus two nodes can end up in the same cluster even if the method did not explicitly regroup them.

IV. Experiments

We present in this section experiments that have been conducted on the dataset WEBSPAM-UK2007[2]. This dataset is a crawl of the .uk domain made in May 2007. It is composed of 105 896 555 nodes. These nodes belong to 114 529 hosts and 6 478 of these hosts have been tagged. Please pay attention to the fact that hosts are tagged, not pages (e.g. entire domains instead of peculiar pages). We use the Webgraph [2] version of the dataset by Boldi and Vigna since it allows to manipulate huge graphs without using a lot of memory.

The tagged hostnames are separated in 3 user-evaluated categories: *spam* (690 972 nodes), *nonspam* (5 314 671 nodes) and *undecided* (201 205 nodes). Using this information we can construct 3 sets of web pages corresponding to the 3 categories.

[2] Yahoo! Research: ”Web Spam Collections”. http://barcelona.research.yahoo.net/webspam/datasets/ Crawled by the Laboratory of Web Algorithmics, University of Milan, http://law.dsi.unimi.it/. URLs retrieved 05 2007.

	Webgraph	*spam*		*nonspam*		*undecided*	
		value	‰	value	‰	value	‰
PageRank	84 015 567.786	517 546.3795	6.16	4 230 292.491	50.35	167 809.751	2
Tech1	68 943 484.072	422 932.8076	6.13	3 449 440.644	50.03	141 221.9441	2.05
Tech2	48 431 264.361	294 940.5303	6.09	2 323 473.016	47.97	97 550.25296	2.01
Tech3	75 176 329.382	461 598.8212	6.14	3 809 062.589	50.67	150 273.0357	2

Table I
PAGERANKS OF EACH SET

	20%				30%			
	Total		Intersection		Total		Intersection	
	Nombre	Score	Nombre	Score	Nombre	Score	Nombre	Score
PageRank	56	104 580	46	89 726	186	155 798	147	128 351
Tech1	66	85 451.5	46	74 794.4	210	127 193	147	111 833
PageRank	56	104 580	36	87 877.8	186	155 798	142	136 765
Tech2	49	59 648.9	36	42 502.1	203	88 731	142	63 700.1
PageRank	56	104 580	52	99 496	186	155 798	161	143 803
Tech3	65	92 937.9	52	84 089.2	189	138 824	161	128 264

Table II
EFFECTS OF DIFFERENTS TESTS ON SPAM TAGGED PAGES

	20%				30%			
	Total		Intersection		Total		Intersection	
	Nombre	Score	Nombre	Score	Nombre	Score	Nombre	Score
PageRank	958	846 644	799	699 145	2 433	1 269 460	1 901	1 062 620
Tech1	1 049	690 274	799	596 293	2 776	1 035 060	1 901	892 642
PageRank	958	846 644	312	389 323	2 433	1 269 460	1 030	745 638
Tech2	538	465 120	312	331 698	1 605	697 270	1 030	547 989
PageRank	958	846 644	763	714 624	2 433	1 269 460	2 028	1 087 260
Tech3	830	762 651	763	728 457	2 166	1 142 930	2 028	1 102 770

Table III
EFFECTS OF DIFFERENTS TESTS ON NONSPAM TAGGED PAGES

	20%				30%			
	Total		Intersection		Total		Intersection	
	Nombre	Score	Nombre	Score	Nombre	Score	Nombre	Score
PageRank	49	34 156.5	42	29 705.4	109	50 569.9	96	45 923.6
Tech1	48	28 748.5	42	26 154.3	118	42 566.2	96	39 577.6
PageRank	49	34 156.5	16	13 851.7	109	50 569.9	49	31 952.9
Tech2	23	19 875.7	16	14 271.6	66	29 427.1	49	25 522.6
PageRank	49	34 156.5	38	27 011.7	109	50 569.9	85	41 865.5
Tech3	45	30 774.8	38	26 741.8	111	45 230.8	85	41 002

Table IV
EFFECTS OF DIFFERENTS TESTS ON UNDECIDED TAGGED PAGES

We first evaluate the pagerank of each node of our dataset. Then we sort each set *spam* , *nonspam* and *undecided* in decreasing pagerank value.

Then we apply each technique to the graph before computing a special version of the pagerank where i contributes to the pagerank of j iff $i \to j$ and i and j are in separate clusters. The contribution C_{ij} of i to j is the following: $C_{ij} = \frac{Pr(i)}{k_i}$ where $k_i = |\{j|i \to j, \mathrm{cl}(i) \neq \mathrm{cl}(j)\}|$. This is the same as running the PageRank on a graph where all intra clusters edges have been removed. Results can be found in table I. We do not use the normalized version of the PageRank where they all add up to 1 since we make the computation over a huge graph and don't want to be limited by the machine precision. All ‰ in table I don't add up to one since we consider only a fraction ($\sim 5.86\%$) of all pages (the tagged web pages). It can be seen in this table that every method make the pagerank of all three sets drop. This is easily understandable. Since many edges are removed from the graph, the pagerank can not spread as easily.

Tech1 reduces the whole pagerank of the graph of $\sim 18\%$, Tech2 reduces it by almost 43% and Tech3 by $\sim 10\%$.

We can see that proportions of each set is mostly respected using whatever technique. Tech2 is the only one to reduce some set's pagerank. Since it is the *nonspam* set (reduced by more than 2‰) this is not a good news. Tech3 slightly increases the *nonspam* pagerank while slightly decreasing the *spam* pagerank. Obviously this table does not provide enough information to take some conclusions.

Let's take a closer look on the results. We focus on nodes with a proportionately high pagerank (nodes well ranked). We will concentrate our analysis on the first 20% (resp. 30%) of each set, meaning nodes representing 20% (resp. 30%) of the set pagerank. Tables II, III and IV represent for each technique the number of nodes and score for the whole 20 (resp 30) top percent of each set and the number of nodes and score for the intersection with the 20 (resp 30) top percent of the pagerank of the set. We want to ensure that the demotion observed at the whole graph level is not uniformly distributed amongst all nodes.

Table II presents the results for the *spam* set. Regarding the top 20% of this set we can see that for Tech1 and Tech3 there are more nodes in this top 20 than for the PageRank meaning that each node is weaker. There are fewer nodes for the Tech2.

Looking at the intersection of each set we can observe that Tech1 demotes the intersection's pagerank by less than 17% which is less than the general demotion registered for this method. Tech2 demotes the pagerank by more than 51% which is better than the general reduction meaning that this spam is actually demoted. Tech3 performs a demotion of almost 15.5% on the intersection which is also better than the general demotion on the whole graph.

On the top 30% of the *spam* set, Tech2 improves its results up to a 53% demotion more than 10 points above the average demotion. Tech1 results worsen to almost 13% and Tech3 efficiency falls to the average demotion.

Now let us focus on the intersections for the 30 top percent. It is interesting for us to have big enough intersection in this case to be sure that strong demoted *spam* nodes are not replaced by stronger promoted *spam* nodes. The size of all intersections combined with the fraction of the set's pagerank they represent allow us to be sure that it is the case.

We can see in this table that only two methods (Tech2 and Tech3) succeeded in *spam* demotion. Tech1 has a trend at promoting important *spam* pages.

Table III shows the results for the *nonspam* set. It is important here to confirm the good results obtained by both Tech2 and Tech3.

We first study the results of Tech1. We observe that it has more pages in both top 20 and top 30 percent of the *nonspam* set. This shows that those pages are weaker and hence demoted. For the top 20% (resp. 30%), Tech1 demotes the intersection by 14.7% (resp. 16%) which represents a small promotion compared to the general demotion. The score on the top 30% is actually worse than the one for the top 30% of *spam* pages.

Tech2 has the smallest number of pages composing the top 20 and 30 percent of the *nonspam* set. This means that the pages are stronger after the application of this method than before. Tech2 demotes the intersection of the top 20% (resp. 30%) by 14.8% (resp. 26.5%) which is way less than the average demotion on the whole graph. This means that these *nonspam* pages are promoted compared to the rest of the graph.

Tech3 also has a smaller number of pages than the PageRank in its top 20 and 30 percent for the *nonspam* set ensuring that those pages have a higher pagerank on average. On this particular set, Tech3 realises negatives demotions *ie* promotions of respectively almost 2% for the top 20% and $\sim 1,43\%$ for the top 30%. These of course are better results than those observed on the whole graph since albeit the general graph lost some pagerank, those particular pages gained some.

Looking at the intersections we can see that Tech1 and Tech3 have large enough intersections but that Tech2 has a smaller one compared to its intersection on the *spam* set. This is of less harm here since we are less concerned about the promotion of *nonspam* nodes but we would like to keep the same sorting as the PageRank as much as possible. The size of this intersection can be explained by the fact that at the top level the fraction of pagerank represented by the *nonspam* set after Tech2 has been applied is smaller than the one of the PageRank, meaning that some important nodes have been demoted since the number of nodes is the same. We are allowed to think then that the ranking of Tech2 may preserve an important part of the PageRank ranking on the *nonspam* set.

Tech3 outperforms Tech2 on this table but it was the contrary on the *spam* table. It is of interest to see how they can be ranked and if the *undecided* set can be helpful to do that.

Table IV concerns the *undecided* set. This set is the smallest and the one that contains the least relevant information since pages contained in this set were not clearly identified as either *spam* or *nonspam* pages.

Our first method continues to produce the same effect previously seen on the first two sets. Meaning the demotion observed on the top 20% (resp. 30%) is $\sim 12\%$ (resp. 13.8%), being inferior to the average demotion. We can then conclude that this technique slightly increase the pagerank of already high ranked nodes and demotes poorly ranked nodes.

Tech2 promotes the top 20% intersection of the *undecided* set by more than 3% but if we consider the top 30% there actually is a demotion of 20% which is less than the average observed on the whole graph.

Tech3 practically does not touch to the pagerank of *undecided* nodes. The registered demotions for the top 20

	Demotion	Promotion	Total
nonspam	458	572	1030
spam	98	44	142
Total	556	616	1172

Table V
VALUES OBTAINED FOR TECH2

	Demotion	Promotion	Total
nonspam	488.63	541.37	1030
spam	67.37	74.63	142
Total	556	616	1172

Table VI
EXPECTED VALUES FOR TECH2

and 30 percent are respectively of 1% and 2%. These results are again way above the average results of this method.

The results obtained by Tech2 and Tech3 could be explained by the fact that these sites are borderline. it means that some may be *spam* while other are *nonspam* nodes. Thus, some nodes use the techniques tracked by our methods while others don't. This is clearly visible for Tech2 where we have a promotion on the top 20% but a demotion on the top 30%. Moreover we can see that the number of pages almost triple between the top 20 and 30 percent meaning that there is a gap in pagerank.

Analysing the effects of our three approaches on the *spam* , *nonspam* and *undecided* sets made us realise that Tech1 is not helpful but that Tech2 and Tech3 succeeded in demoting the effects of Webspam while promoting honest pages. Tech2 outperforms Tech3 concerning the demotion of Webspam and *vice versa* regarding *nonspam* promotion. In the next section we will use statistical tools to check whether these techniques are significantly efficient.

V. STATISTICAL TEST

In this section, we are looking for statistical evidence of the efficiency of our methods to ensure that they are more than just working heuristics. We saw that Tech2 and Tech3 have different effects on pages based on their set of origin.

	Demotion	Promotion	Total
nonspam	86	1942	2028
spam	7	154	161
Total	93	2096	2189

Table VII
VALUES OBTAINED FOR TECH3

	Demotion	Promotion	Total
nonspam	86.16	1941.84	2028
spam	6.84	154.16	161
Total	93	2096	2189

Table VIII
EXPECTED VALUES FOR TECH3

	Demotion	Promotion	Total
nonspam	1.92	1.73	3.65
spam	13.93	12.57	26.51
Total	15.85	14.31	**30.16**

Table IX
χ^2 VALUES FOR TECH2

	Demotion	Promotion	Total
nonspam	0	0	0
spam	0	0	0
Total	0	0	**0**

Table X
χ^2 VALUES FOR TECH3

We want to make sure that it is not just a huge coincidence but that it is in fact our methods that effectively help to separate web pages. We will use a χ^2 independence test to verify that fact.

Here we are only interested in pages with high pagerank before and after the computation of one of our method on the graph. We want to see how these pages are treated by Tech2 and Tech3. We make two categories, pages that are demoted (meaning their particular demotion is greater than the average one) and pages that are promoted (their particular demotion is either negative or less or equal to the average one). Since we are only interested in pages with high pageranks, our sample for each set will be the top 30%.

The hypothesis $\mathcal{H}_0$ we want to test is that both *spam* and *nonspam* pages share the same distribution.

The values V_{ij} for each set and each category can be found in table V for Tech2 and table VII for Tech3. All categories fill the minimum requirements for the χ^2 test. Table VI and VIII show the expected values calculated with the following formula:

$$E_{ij} = \frac{S_{i*} * S_{*j}}{S_{**}}$$

where S_{i*} is the sum of the i^{th} line S_{*j} the sum of the j^{th} column and S_{**} the sum over the lines and columns.

Finally the χ^2 value for both tests can be found in table IX and table X respectively. Since this χ^2 test is made over 2 categories and 2 sets of values, the critical value to exceed is 3.84 if we want to reject the hypothesis $\mathcal{H}_0$ with a probability of error of 5%. The χ^2 value is calculated according to the following formula:

$$\chi^2 = \sum_{i,j} \chi^2_{ij} \qquad \text{where} \qquad \chi^2_{ij} = \frac{(V_{ij} - E_{ij})^2}{E_{ij}}$$

The χ^2 value obtained for Tech2 is **30.16** meaning that we can reject the hypothesis $\mathcal{H}_0$ with at most 0.5% chances to be wrong. Thus it can be stated that *spam* and *nonspam* pages do not share the same distribution in this case *i.e* Tech2 effectively separates *spam* pages from *nonspam* ones.

The score for Tech3 leaves no doubt, **0** meaning that the two samples share the same distribution. As well as it seems to work in practice, there is no statistical evidence that tech3 may able to tell the difference between *spam* pages and *nonspam* ones.

We were only able to show statistical evidence for the good behaviour of one of our technique, leaving us with Tech3 just as an heuristic that seems to work.

VI. Conclusion

In this paper we have presented different clustering methods for the demotion of the effects of webspam on the PageRank algorithm. All three approximate methods are fast to compute and need only a small amount of memory. The last two techniques, based respectively on the identification of small circuits in the graph and random walks, are shown to have good results on webspam demotion. Moreover, for the method Tech2, we have statistical evidence that it can separate spam and nonspam nodes. The complexity of this method is $\mathcal{O}(n + m)$. Thus this fully automatic method could be effectively added to the already existing arsenal for the Webspam detection and demotion of a search engine. It is still of interest to investigate other methods to perform approximate clustering on the webgraph.

References

[1] Reid Andersen, Christian Borgs, Jennifer Chayes, John Hopcroft, Kamal Jain, Vahab Mirrokni, and Shanghua Teng. Robust pagerank and locally computable spam detection features. In *AIRWeb '08: Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, pages 69–76, New York, NY, USA, 2008. ACM.

[2] Paolo Boldi and Sebastiano Vigna. The webgraph framework I: Compression techniques. In *In Proc. of the Thirteenth International World Wide Web Conference*, pages 595–601. ACM Press, 2003.

[3] Young-joo Chung, Masashi Toyoda, and Masaru Kitsuregawa. A study of link farm distribution and evolution using a time series of web snapshots. In *AIRWeb '09: Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, pages 9–16, New York, NY, USA, 2009. ACM.

[4] C. de Kerchove, L. Ninove, and P. Van Dooren. Maximizing PageRank via outlinks. *Linear Algebra and its Applications*, 429(5-6):1254–1276, 2008.

[5] Stijn Dongen. A cluster algorithm for graphs. Technical Report 10, 2000.

[6] Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks. *PROC.NATL.ACAD.SCI.USA*, 99:7821, 2002.

[7] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. *Adversarial Information Retrieval on the Web*, 2005.

[8] Zoltan Gyongyi, Pavel Berkhin, Hector Garcia-Molina, and Jan Pedersen. Link spam detection based on mass estimation. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 439–450. VLDB Endowment, 2006.

[9] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, pages 576–587. VLDB Endowment, 2004.

[10] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, 11(6):36–45, 2007.

[11] Vijay Krishnan and Rashmi Raj. Web Spam Detection with Anti-Trust Rank. *AIRWeb 2006 Program*, page 37, 2006.

[12] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. Detecting spam web pages through content analysis. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 83–92, New York, NY, USA, 2006. ACM.

[13] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web, 1999.

[14] Baoning Wu, Vinay Goel, and Brian D. Davison. Topical trustrank: using topicality to combat web spam. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 63–72, New York, NY, USA, 2006. ACM.