Designing a Web Spam Classifier Based on Feature Fusion in the Layered Multi-Population Genetic Programming Framework

Amir Hosein Keyhanipour, Behzad Moshiri
School of Electrical and Computer Engineering, College of Engineering,
University of Tehran, Tehran, Iran

KEYWORD

Web Spam
Feature Fusion
Layered Multi-Population Genetic Programming

ABSTRACT

Nowadays, Web spam pages are a critical challenge for Web retrieval systems which have drastic influence on the performance of such systems. Although these systems try to combat the impact of spam pages on their final results list, spammers increasingly use more sophisticated techniques to increase the number of views for their intended pages in order to have more commercial success. This paper employs the recently proposed Layered Multi-population Genetic Programming model for Web spam detection task as well application of correlation coefficient analysis for feature space reduction. Based on our tentative results, the designed classifier, which is based on a combination of easy to compute features, has a very reasonable performance in comparison with similar methods.

1 Introduction

Due to the drastic growth of Web information, it has become an obligation to evaluate the presented information based on some metrics from both quantitative and qualitative aspects of view. One of the most important quality measurement criteria is spammicity of Web pages. As spam pages could have harmful influence on the functionality and performance of Web retrieval systems, it would be of the most importance to have powerful detection methods to filter such undesirable pages before being able to bias the functionality of Web retrieval systems, especially Web search engines. On the other hand, dynamic nature of Web data and newly developed spamming techniques has made it a necessity to design adaptive and intelligent spam detecting frameworks. In this regard, here we present a GP-based classifier which is able to detect different spamming patterns with considerable performance and efficiency. The achieved results indicate the noticeable performance of the proposed method against the current approaches.

2 Survey on Related Works

Web spamming is as old as commercial search engines. For instance, Lycos dealt with spam pages in 1995. Generally, one could define Web spamming as techniques used to bias the ranking mechanism of Web retrieval systems toward some specific Websites or pages. This could lead to the performance reduction in such systems and makes users unsatisfied. Therefore, most of the commercial search engines try to combat Web spams. Detection of Web spam pages could be thought as a vital step in the improvement of the performance of Web search engines. By doing this step properly, it would be possible to filter such undesirable pages from being included in the processing cycle of search engines including crawl, indexing and retrieval. By this way, the retrieval systems will pay less computational costs and would be able to achieve better performance.

Commonly, there are four known types of spamming techniques which are link-based, content-based, cloaking and click-based methods. Content-based spamming refers to inserting most frequent search terms in the content of Web pages with the aim to provide higher ranking for those pages in most information seeking tasks. The second type of spamming techniques are those which are used to improve the link-based score of Web pages by providing artificially created set of hyperlinks to a specific Website or Webpage. Cloaking as the next type of the spamming techniques is to serve different versions of contents about a specific Web page for Web crawlers and human users. Click spam refers to the method in which specific queries are submitted to Web search engines in order to retrieve some target pages. Then, some scripts are used to continuously click on those pages to simulate the interests of users to those pages.

Commercial search engines need to combat spamming due to their harmful influence on the performance of such systems. Moreover, as the diversity of spamming techniques is vastly increased, some academic sessions are also formed in recent years for more academic contributions. From such circles, one may point out the AIRWeb workshops and specially their Web Spam Challenge [WebSpamChallenge, 2008].

In general, spam detections methods could be categorized in three groups: link-based, content-based and combinative methods which are described here in brief.

The first category contains content-based methods which undertake content properties of Web pages in order to provide a classifier. For this category of algorithms, information such as Content, Title, URL, URL length and etc. are used. For example, in reference [FETTERLY, D. et al. 2004], some simple frequent-based criteria have been used for spam detection. Ntoulas et al. introduced new features based on checksum and word weighting methods [NTOULAS, M. et al. 2006]. In [PISKORSKI, J. et al. 2008], some linguistic features were used for spam detection. Biro et al. [JACINT, I.B. et al. 2008] and Marteniz-Rome et al. [MARTINEZ-ROMO, J. & ARAUJO, L. 2009] proposed a statistical language model based on the content of Web pages to identify spam ones.

The second category includes link-based methods. From those, one may mention the Truncated PageRank algorithm [BECCHETTI, L. et al. 2006a], in which the importance of those neighbors which are topologically close to a target page, are decreased in order to overcome the link farms. Becchetti et al. [BECCHETTI, L. et al. 2006b] used automatic classifiers to detect link-based Spam; Gy¨ongyi et al. [GYONGYI, Z. et al. 2004] separated useful Web pages from spam ones with TrustRank; Zhou et al. [ZHOU, D. et al. 2007] with transductive link spam detection.

The third series of Web spam detection techniques are combinative methods. In [ABERNETHY, J. et al. 2008] a SVM classifier is designed by fusing content and link data. Castillo et al. [CASTILLO, C. et al. 2007] combined content and topology information in a cost-sensitive tree. Bencz´ur et al. [BENCZUR, A. et al. 2006] proposed an approach to detect nepotistic links using language models. In this method, a link is down-weighted if the language models from its source and target page have a great disagreement.

Nowadays, the spam detection concept is flowed up as a hot research topic in many research communities across the world [Najork, M. 2009].

3 Proposed Approach

3.1 Steps of the Proposed Framework

The aim of the method which is discussed in this research is to provide an adaptive dynamic classifier to detect spam Web pages with high accuracy and low computational costs. In order to meet such goal, the algorithm will use a number of documents' features to be able to overcome the dynamic nature of spamming mechanisms. In this regard, a number of steps have to be followed:

1. Selection of suitable subset of features for Web documents: this set should contain informative features which could be computed with low

computational cost. In other words, these features need to be good representatives for Web documents. Meanwhile, the number of such features should be limited in the way not to impose heavy processing load on the detection system. In this regard, based on correlation coefficient analysis [GUYON, I. et al. 2006], a subset containing 82 features is selected which their statistics are demonstrated in Table 1. As it could be seen, we have selected about 26.88% of all features presented in WEBSPAM-UK2007 for our experiments.

To compute the correlation coefficient between any two features, we used the below formula:

$$r_{A,B} = \frac{\sum (A-\overline{A})(B-\overline{B})}{(n-1)\sigma_A\sigma_B} = \frac{\sum (AB) - n\overline{A}\overline{B}}{(n-1)\sigma_A\sigma_B} \quad (1)$$

, in which A and B are two features which their mean values are represented by $\overline{A}$ and $\overline{B}$, respectively.

$$F = \left\{ F_i \middle| |S_i| \geq \sum_{j=1}^{M} |S_j| \middle/ M \wedge T_i > T \right\} \quad (2)$$

$$S_i = \left\{ F_j \middle| Abs\left(CorrelationCoefficient\left(F_i, F_j\right)\right) \geq 0.4 \right\},$$

$$T_i = \sum_{\substack{j=1 \\ F_j \in S_i}}^{M} CorrelationCoefficient\left(F_i, F_j\right) \Big/ |S_i|$$

$$T = \sum_{i=1}^{M} \sum_{\substack{j=1 \\ F_j \in S_i}}^{M} CorrelationCoefficient\left(F_i, F_j\right) \Big/ M$$

The above formulae show the usage of correlation coefficient analysis for feature set reduction. Let M be the number of distinct features, S_i contains those features like F_j which the abstract value of their correlation coefficient with F_i is greater than 0.4. After that, T_i is average value of correlation coefficient values for S_i, and T is the average of all T_i's. By this configuration, those features that their average of correlation coefficients are greater than the average for total and also the number of such correlation coefficients are more than the total average, will fall in the set F and will be eliminated. This method is described formally in the above mentioned formulae. Totally, 223 features were removed in this process. Features employed in this investigation are listed in Appendix I.

2. Providing an initial population of classifiers based on different combination of the selected features: this population will be evolved in a Layered Multi-Population Genetic Programming structure during a number of generations under genetic operators (cross over and mutation) and finally will hand over appropriate classifiers.
3. Computation of evaluation metrics and comparison of the performance of the proposed method with available algorithms.

Feature Category	Selected	All	% Selected
Obvious	2	2	100
Link-based	32	41	78
Content-based	48	96	50
Total	82	305	26.88

Table 1: Features selected for the proposed approach from WEBSPAM-UK2007 dataset

As it could be observed, the usage percentage of Link-based features is more than those of the Content-based ones. This shows the information richness of link-based features.

3.2 Details of the Designed GP-Based Classifier for Spam Detection

The genetic programming as an evolutionary program solving approach is a powerful means to solve a variety of problems. With the use of genetic operators such as cross over and mutation, GP mechanism evolves a population of potential solutions to find out the best ones based on specified evaluation criteria named as fitness function in a number of generations. The fitness function is a user defined criteria used to quantify the goodness of an individual for a specific propose.

As we will introduce the dataset used here for our experimentations, it contains a set of training data like T which are pairs of hostnames and spammicity values plus a features vector corresponding to different specifications of a host. Generally, considering a collection of hosts: $W = \{w_1, w_2, \ldots, w_{|W|}\}$, a set of features: $F = \{f_1, f_2, \ldots, f_{|F|}\}$, and spammicity values: $Y = \{Spam, Non\text{-}Spam\}$, one may define the training dataset as:

$$T = \{\langle (f_1(w_i), \ldots, f_{|F|}(w_i)), y_i \rangle\} \quad (3)$$

, where $y_i \in Y$; and $(f_1(w_i), \ldots, f_{|F|}(w_{|W|}))$ is a $|F|$-dimension vector of features in which $f_k(q_i, d_j)$ shows the value of feature f_k for document d_j.

The values of features are normalized to fall in $[0,1]$ with the use of Min-Max Normalization equation:

$$f_k(w_i) = \frac{f_k(w_i) - \min\{f_k(w_l)\}}{\max\{f_k(w_l)\} - \min\{f_k(w_l)\}} \quad (4)$$

In our method, each individual of a population is a potential spam detection function which is based on a combination of features. This individual assigns a spammicity value to each site. Any individual classifier *I*, consists of three components: a set of variables (features): S_v; a set of constant values: S_c; and a collection of arithmetic operators: S_{op}. By this means, an individual I could be could be represented as: $I = (S_v, S_c, S_{op})$, where:

$$\begin{aligned} S_v &= \{f_i \mid f_i \in F\}, \\ S_c &= \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}, \\ S_{op} &= \{+, -, \times, /, Sin(\ ), Cos(\ ), Exp(\ ), Ln(\ )\}. \end{aligned} \quad (5)$$

In practice, each individual is modeled as a binary tree where its depth is usually predetermined. Figure 1 shows the binary tree schema for $I : ((f_1 + f_2) + (0.3 \times f_3))$.

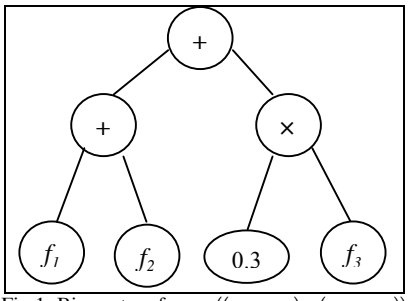

Fig 1: Binary tree for $I : ((f_1 + f_2) + (0.3 \times f_3))$

The overall process is that by having a number of initial populations, they are evolved in parallel by the Layered Multi-Population Genetic Programming framework using genetic operators and after passing a specific number of iterations, the best individuals will be selected as final spam classifiers based on the determined fitness function. In our experiments, we used precision as the fitness function. Figure 2 demonstrates the proposed framework.

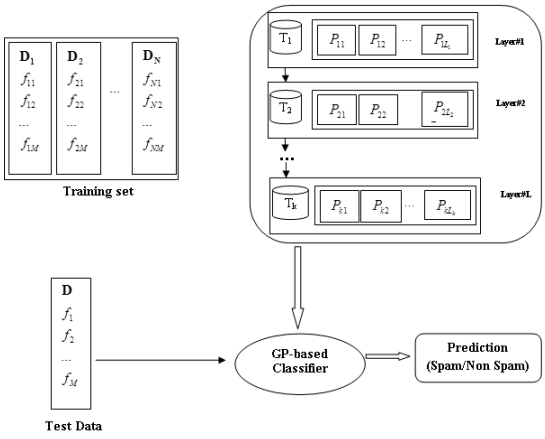

Fig 2: The Proposed Web Spam Detection Framework

The mutation operator is used to change any part of binary-tree of an individual by a pre-specified probability P_m. On the other hand, the crossover operator will combine different parts of two randomly selected individuals with probability P_c. In order to select individuals for crossover, the tournament selection method is used. Briefly, the tournament selection is a method of selecting an individual from a population of individuals in a genetic algorithm. Tournament selection involves running several

"tournaments" among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover. Generally, tournament selection shows a better performance on parallel architectures than usual selection method and allows the selection pressure to be easily adjusted. The tournament size also needs to be preset before beginning of the GP algorithm.
The Layered Multi-Population Genetic Programming framework needs a set of tuning factors which need to be determined by trial and error. Their listing and best values for them are presented in Table 2.

Parameter	Value
Function Type	Linear Function, Shorter ones are preferred
# Layers	2
# Populations in each layer	3
Population size	600 individuals
Tree depth	6, 7, 8 and 9
Tournament size	5
Crossover rate	0.95
Mutation rate	0.05
Reproduction rate	0.003 (2 most fit individuals in each population)
Arithmetic operations weight	Equal selection probability for: +, -, /, *, Sin(), Cos(), Ln() and Exp()

Table 2: The tuning parameters of the proposed Layered Multi-Population Genetic Programming framework

4 Evaluation Framework

4.1 Benchmark Dataset

We use a publicly available Web Spam collection [JACINT, I.B. et al. 2008] based on crawls of the .UK Web domain done in May 2007. WEBSPAM-UK2007 includes 105.9 million pages and over 3.7 billion links for about 114,529 hosts. This reference collection is tagged by a group of volunteers labeling hosts as “normal”, “spam” or “borderline”. Corresponding to each document, a number of features are considered and their values are computed. From this volume of data, a subset containing about 6479 hosts were selected for Web Spam Challenge 2008 workshop. Table 3 shows the overall distribution of these hosts based on their type. For this subset, about 2/3 which contains 4,725 instances was determined by the workshop committee as training set and the rest 2,024 instances were considered as test set. Moreover, in our research based on the rule of Web Spam Challenge 2008 workshop, the undecided items were ignored in our computations. Therefore, only about 6% of hosts presented in this dataset are actually tagged as spam.

Host Type	No. of features
Non-Spam	5709
Spam	344
Undecided	426

Table 3: Type Distribution of Hosts presented at Web Spam Challenge 2008

In general, WEBSPAM-UK2007 includes about 305 different features per each document which are categorized in three different categories [UK-2007, 2008]:

- 2 obvious features, which are: *total number of pages for each host*, *and length of host name*;
- 96 content-based features, derived from the content of Web pages including "# of words in the first page", "Average length of title for pages in a host" and so on. To see the complete list of these features, please see [UK-2007, 2008].
- Link based features, extracted from link structure between Web pages. These features are of two major types:
  - Direct link-based features for the hosts, measured in both the home page and the page with the maximum PageRank in each host. This set contains in-degree, out-degree, PageRank, edge reciprocity, Assortativity coefficient, TrustRank, Truncated PageRank, estimation of supporters, and so on. See [UK-2007, 2008] for comprehensive description.
  - Transformed link-based features, which usually work better than direct

link-based features for classification purpose in practice. However, their computational cost is higher than direct link-based features. They include mostly ratios between features such as Indegree/PageRank or TrustRank/PageRank, and $\log(.)$ of several features. Their list is also available at [UK-2007, 2008].

4.2 Evaluation Metrics

Based on the convention of Web Spam Challenge 2008 workshop, we considered AUC[1] as our main evaluation metric to compare the performance of the proposed approach with respect to similar ones; in the meantime, other classification metrics such as Cross-Entropy, Accuracy, Sensitivity, Specificity, F1-measure, MSE[2], MAP[3], Precision and Recall are also computed.

It has been shown that AUC measure is statistically consistent and more discriminating than accuracy to evaluate the performance of binary classifiers. In fact, an ROC[4] diagram is a plot of true positive rate vs. false positive rate as the prediction threshold sweeps through all the possible values. It is the same as plotting sensitivity vs. $1-\text{specificity}$ while sweeping the threshold. AUC is the area under this curve. AUC of 1 is perfect prediction (all positive cases sorted above all negative cases). AUC of 0.5 is random prediction in which, there is no relationship between the predicted values and truth. AUC below 0.5 indicates there is a relationship between predicted values and truth, but the model is backwards. It is possible to have another definition of the AUC. Imagine sorting the data by predicted values. Suppose this sort is not perfect, i.e., some positive cases sort below some negative cases. AUC effectively measures how many times you would have to swap cases with their neighbors to repair the sort. In fact, AUC would be the normalization of this value:

$$AUC = 1.0 - \frac{\#\ of\ swaps\ \ to\ repair\ \ sort}{(\#\ of\ \ positives)\times(\#\ of\ \ negatives)} \quad (6)$$

Another important metric is the cross-entropy which indicates the distance between the truth values and the predicted ones as below:

$$CrossEntropy = SUM(Targ\times log(Pred)+(1-Targ)\times log(1-Pred)) \quad (7)$$

Unlike squared error, cross-entropy considered the predicted values as probabilities on the interval $[0,1]$ which indicate the probability that the case is class 1.

5 Experimental Results

Our experiments are done with two phases; the first phase uses the selected subset of about 26.88% of all features which was described previously and the second phase which is used for comparison purpose and utilizes the whole set of features. For the selected subset, we achieved the AUC value of 0.7983 which is 4th place; and for the whole features, we got AUC of 0.8145 which is the 3th position. The reported AUC values for top-ranked participant teams in Web Spam Challenge 2008 are presented in Table 4 [DENOYER, L. 2008].

Rank	Team	AUC Value
1	Geng et al.	0.848
2	Tang et al.	0.824
3	Abernethy and Chapelle	0.809
4	Siklosi and Benczur	0.796
5	Bauman et al.	0.783
6	Skvortsov	0.731
7	Siklosi	0.726

Table 4: Reported AUC values for top-ranked participants of Web Spam Challenge 2008

Figures 3 and 4 illustrate the ROC curve for our two experimentations as well as those of the algorithms proposed by top-ranked participants of Web Spam Challenge 2008, respectively.

[1] Area Under the ROC Curve
[2] Mean Square Error
[3] Mean Average Precision
[4] Receiver-Operating Characteristic

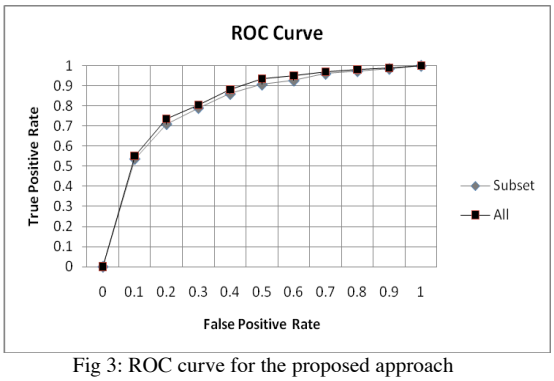

Fig 3: ROC curve for the proposed approach

It could be observed from Figure 3 that the usage of all features will show a slight improvement but the application of the reduced set has completely comparable performance.

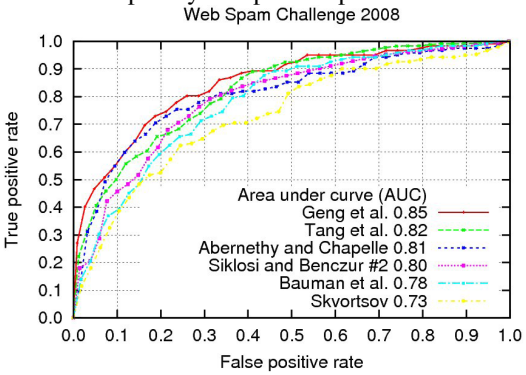

Fig 4: Reported ROC curves for participants in Web Spam Challenge 2008 [DENOYER, L. 2008]

We have also computed other familiar classification criteria such as accuracy, sensitivity, specificity, F1 measure and so on for the reduced set of features. These computations are presented in Table 5. It is mentionable that our approach has the accuracy more than 0.92 to detect spam pages.

Criterion	Value	Comments
Accuracy	0.924102998	pred_thresh 0.500000
Positive Predictive Value	0.288271688	pred_thresh 0.500000
Negative Predictive Value	0.953916764	pred_thresh 0.500000
Sensitivity	0.259208129	pred_thresh 0.500000
Specificity	0.966106234	pred_thresh 0.500000
Precision	0.288271688	pred_thresh 0.500000
Recall	0.259208129	pred_thresh 0.500000
F1 score	0.271452387	pred_thresh 0.500000
Lift (at threshold)	4.847611929	pred_thresh 0.500000
Precision/Recall Break Even Point	0.349001316	
Mean Average Precision	0.307743284	
ROC area	0.79453	
ROC area up to 50 negative examples	0.184389988	
Rank of last (poorest ranked) positive case	2052.761507	
The top ranked case positive	0	
Is there a positive in the top 10 ranked cases	1	
Slac Q-score [VOGEL, D.S. et al. 2004]	0.837775161	
Root Mean Squared Error	0.248347729	

Table 5: Classification measurements for the proposed algorithm applied on the selected subset of features

The results achieved, confirm that although the proposed classifier uses a number of simple link-based and content features to identify spam Web documents, its performance is comparable with those of similar proposed algorithms which employ all features set. This could mainly be thought as a result of the Layered Multi-Population Genetic Programming framework which provides a parallel and extensive search in the space of possible solutions to find the near-global optimum results.

6 Discussion and Further Works

The rise in popularity of Web search engines has caused a raise in the amount of Web spam,

aimed at manipulating the rank function in search engines. Web spams can origin serious problems for web retrieval systems, because they degrade the ranking quality, and increases the index size. Web spam has received much interest recently. Every day, spammers are making progress on new techniques used to mislead search engines. Having such a dynamic nature, spam detection needs adaptive and efficient algorithms to find newly emerged spamming patterns.

In this paper, we applied the newly introduced Layered Multi-Population Genetic Programming model to the problem of Web spam classification. This model provides a more comprehensive search in the solution space by less computational effort. We also used correlation coefficient analysis in order to reduce the input space for more efficiency and effectiveness. In this way, we used less than 27% of features set of WEBSPAM-UK2007 presented for the Web Spam Challenge 2008. Using this method, we could achieve acceptable results which are comparable with the performance of other presented methods which employ all the feature set.

In future works, we would like to analyze the feature space with other feature selection methods such as C4.5 [KOTSIANTIS, S.B. 2007] or principle component analysis approach. The use of other classification techniques such as neural networks could also be considered as future research directions.

7 Acknowledgment

The authors would like to acknowledge the financial support of University of Tehran for this research under grant number 8101004/1/02. We also give special thanks to Ms. Maryam Piroozmand, M.Sc. student in Artificial Intelligence at Department of Computer Engineering and Information Technology, Amir-Kabir University, Tehran, Iran, for her support and help.

8 References

[WebSpamChallenge, 2008]	Official Website of the Web Spam Challenge 2008, 2008, http://Webspam.lip6.fr/wiki/pmwiki.php?n=Main.PhaseIII, Accessed 17 August 2013
[FETTERLY, D. *et al*. 2004]	FETTERLY, D., MANASSE, M., & NAJORK, M. Spam, damn spam, and statistics: using statistical analysis to locate spam Web pages. In: *7th international workshop on the Web and Databases*, pp. 1-6, 2004
[NTOULAS, M. *et al*. 2006]	NTOULAS, A., NAJORK, M., MANASSE, M., & FETTERLY, D. Detecting spam Web pages through content analysis. In: *15th international conference on World Wide Web*, pp. 83-92, 2006
[PISKORSKI, J. *et al*. 2008]	PISKORSKI, J., SYDOW, M., & WEISS D. Exploring linguistic features for Web spam detection: a preliminary study, In: *4th international workshop on Adversarial information retrieval on the Web*, pp. 25-28, 2008
[JACINT, I.B. *et al*. 2008]	JACINT, I.B., ANDRAS, S., & BENCZUR, A. Latent Dirichlet Allocation in Web Spam Filtering. In: *4th international workshop on Adversarial information retrieval on the Web*, pp. 29-32, 2008
[MARTINEZ-ROMO, J. & ARAUJO, L. 2009]	MARTINEZ-ROMO, J., & ARAUJO, L. Web Spam Identification through Language Model Analysis. In: *5th international workshop on Adversarial Information Retrieval on the Web*, pp. 21-28, 2009
[BECCHETTI, L. et al. 2006a]	BECCHETTI, L., CASTILLO, C., DONATO, D., LEONARDI, S., & BAEZA-YATES, R. Using rank propagation and probabilistic counting for link-based spam detection. In: *Workshop on Web Mining and Web Usage Analysis*, 2006
[BECCHETTI, L. *et al*. 2006b]	BECCHETTI, L., CASTILLO, C., DONATO, D., LEONARDI, S., & BAEZA-YATES, R. Link-based characterization and detection of Web spam. In: *second international workshop on Adversarial information retrieval on the Web,* 2006
[GYONGYI, Z. *et al*. 2004]	GYONGYI, Z., GARCIA-MOLINA, H., & PEDERSEN, J. Combating Web spam with TrustRank, In: *30th international conference on Very large data bases, VLDB Endowment,* pp. 576-587, 2004
[ZHOU, D. *et al*. 2007]	ZHOU, D., BURGES, C., & TAO, T. Transductive link spam detection. In: *3rd international workshop on Adversarial information retrieval on the Web*, pp. 21-28, 2007
[ABERNETHY, J. *et al*. 2008]	ABERNETHY, J., CHAPELLE, O., & CASTILLO, C. Webspam identification through content and hyperlinks. In: *4th international workshop on Adversarial information retrieval on the Web*, pp. 41-44, 2008
[CASTILLO, C. *et al*. 2007]	CASTILLO, C., DONATO, D., GIONIS, A., MURDOCK, V., & Silvestri, F. Know your neighbors: Web spam detection using the Web topology. In: *30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 423-430*, 2007
[BENCZUR, A. *et al*. 2006]	BENCZUR, A., BIRO, I., CSALOGANY, K., & UHER, M. Detecting nepotistic links by language model disagreement. In: *15th international conference on World Wide Web*, pp. 939-940, 2006
[Najork, M. 2009]	NAJORK, M. Web Spam Detection, In: *Encyclopedia of Database Systems*, ed. by Liu, L., & Ozsu, M.T., pp. 3520-3523, 2009
[GUYON, I. *et al*. 2006]	GUYON, I., GUNN, S., NIKRAVESH, M., & ZADEH, L.A. Feature Extraction: Foundations and Applications, *Series Studies in Fuzziness and*

Soft Computing, First ed., Springer, 2006

[UK-2007, 2008] UK-2007 Dataset Website, 2008, http://www.yr-bcn.es/Webspam/datasets/uk2007/features/, Accessed 17 August 2013

[DENOYER, L. 2008] L. DENOYER, Web Spam Challenge Results, 2008, http://airWeb.cse.lehigh.edu/2008/Web_spam_challenge/results.pdf, Accessed 17 August 2013

[VOGEL, D.S. *et al*. 2004] VOGEL, D.S., GOTTSCHALK, E., & WANG, M.C. Anti-matter detection: Particle Physics Model for KDD Cup 2004. *ACM SIGKDD Explorations Newsletter*, 6(2): 109-112, 2004

[KOTSIANTIS, S.B. 2007] KOTSIANTIS, S.B. Supervised Machine Learning: A Review of Classification Techniques, *Informatica*, 31: 249-268, 2007

9 Appendix I

List of feature used in this paper as a subset of WEBSPAM-UK2007 Dataset

No.	Feature ID	Feature Name	Category	Comments
1	1	number_of_pages	Obvious	number of pages in the host
2	2	length_of_hostname	Obvious	number of characters in the host name
3	3	HST_1	Content	Number of words in the page (home page = hp)
4	4	HST_2	Content	Number of words in the title (hp)
5	5	HST_3	Content	Average word length (hp)
6	6	HST_4	Content	Fraction of anchor text (hp)
7	7	HST_5	Content	Fraction of visible text (hp)
8	8	HST_6	Content	Compression rate of the hp
9	9	HST_7	Content	Top 100 corpus precision (hp)
10	13	HST_11	Content	Top 100 corpus recall (hp)
11	17	HST_15	Content	Top 100 queries precision (hp)
12	21	HST_19	Content	Top 100 queries recall (hp)
13	25	HST_23	Content	Entropy (hp)
14	26	HST_24	Content	Independent LH (hp)
15	27	HMG_25	Content	Number of words in the page (page with max PageRank in the host = mp)
16	28	HMG_26	Content	Number of words in the title (mp)
17	29	HMG_27	Content	Average word length (mp)
18	30	HMG_28	Content	Fraction of anchor text (mp)
19	31	HMG_29	Content	Fraction of visible text (mp)
20	32	HMG_30	Content	Compression rate (mp)
21	33	HMG_31	Content	Top 100 corpus precision (mp)
22	37	HMG_35	Content	Top 100 corpus recall (mp)
23	41	HMG_39	Content	Top 100 queries precision (mp)
24	45	HMG_43	Content	Top 100 queries recall (mp)
25	49	HMG_47	Content	Entropy (mp)
26	50	HMG_48	Content	Independent LH (mp)

27	51	AVG_49	Content	Number of words in the page (average value for all pages in the host)
28	52	AVG_50	Content	Number of words in the title (average value for all pages in the host)
29	53	AVG_51	Content	Average word length (average value for all pages in the host)
30	54	AVG_52	Content	Fraction of anchor text (average value for all pages in the host)
31	55	AVG_53	Content	Fraction of visible text (average value for all pages in the host)
32	56	AVG_54	Content	Compression rate (average value for all pages in the host)
33	57	AVG_55	Content	Top 100 corpus precision (average value for all pages in the host)
34	61	AVG_59	Content	Top 100 corpus recall (average value for all pages in the host)
35	65	AVG_63	Content	Top 100 queries precision (average value for all pages in the host)
36	69	AVG_67	Content	Top 100 queries recall (average value for all pages in the host)
37	73	AVG_71	Content	Entropy (average value for all pages in the host)
38	74	AVG_72	Content	Independent LH (average value for all pages in the host)
39	75	STD_73	Content	Number of words in the page (Standard deviation for all pages in the host)
40	76	STD_74	Content	Number of words in the title (Standard deviation for all pages in the host)
41	77	STD_75	Content	Average word length (Standard deviation for all pages in the host)
42	78	STD_76	Content	Fraction of anchor text (Standard deviation for all pages in the host)
43	79	STD_77	Content	Fraction of visible text (Standard deviation for all pages in the host)
44	80	STD_78	Content	Compression rate in the home page (Standard deviation for all pages in the host)
45	81	STD_79	Content	Top 100 corpus precision (Standard deviation for all pages in the host)
46	85	STD_83	Content	Top 100 corpus recall (Standard deviation for all pages in the host)
47	89	STD_87	Content	Top 100 queries precision (Standard deviation for all pages in the host)
48	93	STD_91	Content	Top 100 queries recall (Standard deviation for all pages in the host)
49	97	STD_95	Content	Entropy (Standard deviation for all pages in the host)
50	98	STD_96	Content	Independent LH (Standard deviation for all pages in the host)

51	102	assortativity_hp	Link	Assortativity coefficient of the home page (degree / average degree of neighbors). Degree in this case is undirected (in_degree+out_degree)
52	103	assortativity_mp	Link	Assortativity coefficient of the page with the maximum PageRank
53	104	avgin_of_out_hp	Link	Average in-degree of out-neighbors of home page (hp)
54	105	avgin_of_out_mp	Link	Average in-degree of out-neighbors of page with maximum PageRank (hp)
55	106	avgout_of_in_hp	Link	Average out-degree of in-neighbors of hp
56	107	avgout_of_in_mp	Link	Average out-degree of in-neighbors of mp
57	108	indegree_hp	Link	Indegree of hp
58	109	indegree_mp	Link	Indegree of mp
59	110	neighbors_2_hp	Link	Neighbors at distance 2 of hp
60	111	neighbors_2_mp	Link	Neighbors at distance 2 of mp
61	116	outdegree_hp	Link	Out-degree of hp
62	117	outdegree_mp	Link	Out-degree of mp
63	118	pagerank_hp	Link	PageRank of hp (calculated in the doc graph with no self-loops, using a damping factor of 0.85, with 50 iterations)
64	119	pagerank_mp	Link	PageRank of mp
65	120	prsigma_hp	Link	Standard deviation of the PageRank of in-neighbors of hp
66	121	prsigma_mp	Link	Standard deviation of the PageRank of in-neighbors of mp
67	122	reciprocity_hp	Link	Fraction of out-links that are also in-links of hp. For instance, if the hp has 5 out-links, and 3 of those pages links back to the home page, the assortativity coefficient is 3/5. A page with no out-links has assortativity coefficient of 0.
68	123	reciprocity_mp	Link	Fraction of out-links that are also in-links of mp
69	124	siteneighbors_1_hp	Link	Number of different hosts pointing to hp, obtained by approximate algorithm (could have been done exactly, but used the approximate algorithm)
70	125	siteneighbors_1_mp	Link	Number of different hosts pointing to mp
71	126	siteneighbors_2_hp	Link	Number of different hosts (approx.) supporting at distance 2 the hp
72	127	siteneighbors_2_mp	Link	Number of different hosts (approx.) supporting at distance 2 the mp
73	132	truncatedpagerank_1_hp	Link	TruncatedPageRank using truncation distance 1, hp
74	133	truncatedpagerank_1_mp	Link	TruncatedPageRank using truncation distance 1, mp
75	134	truncatedpagerank_2_hp	Link	TruncatedPageRank using truncation distance 2, hp

76	135	truncatedpagerank_2_mp	Link	TruncatedPageRank using truncation distance 2, mp
77	136	truncatedpagerank_3_hp	Link	TruncatedPageRank using truncation distance 3, hp
78	137	truncatedpagerank_3_mp	Link	TruncatedPageRank using truncation distance 3, mp
79	138	truncatedpagerank_4_hp	Link	TruncatedPageRank using truncation distance 4, hp
80	139	truncatedpagerank_4_mp	Link	TruncatedPageRank using truncation distance 4, mp
81	140	trustrank_hp	Link	TrustRank of hp (obtained using 3,800 hosts from ODP as trusted set) -- the list of URL identifiers used is at http://www.yr-bcn.es/Webspam/datasets/uk2007/features/uk-2007-05.odp_docid.csv.gz NOTE: this feature can be improved by using more ODP hosts in the seed set.
82	141	trustrank_mp	Link	TrustRank of mp