Web spam detection through link-based features

Lorenzo Basile

Information Retrieval exam

January 15, 2021

UNIVERSITÀ
DEGLI STUDI DI TRIESTE

DATA SCIENCE &
SCIENTIFIC COMPUTING

Outline

1. The problem: Vulnerability of PageRank
2. Link-based features for spam detection
3. Robust PageRank

Web spam

Web spam is the fraudulent manipulation of web page content for the purpose of appearing high up in search results for selected keywords.
Spamming techniques can be broadly classified into two groups:

- Content spam: pages try to boost their search rank by inserting attractive keywords or hidden text
- Link spam: manipulation of the link structure of the sites, by means of link farms (densely connected sets of pages, created explicitly with the purpose of deceiving a link-based ranking algorithm)

Combating link spam

Common ranking algorithms such as PageRank are usually not sufficient to detect spam pages and rank them lower than trustworthy ones.

Combating link spam

Common ranking algorithms such as PageRank are usually not sufficient to detect spam pages and rank them lower than trustworthy ones.
Two possible solutions:

- Using simple Machine Learning techniques to detect spam websites to warn users of potential threats
- Building an alternative ranking algorithm that can be seen as a robust version of PageRank

This project is focused on link spam at host level.

Dataset

The dataset WEBSPAM-UK2006 contains 11402 hosts and it is based on a crawl of the .uk domain done in May, 2006.
8045 hosts were manually labeled either as normal, spam or undecided by domain experts.

Dataset

The dataset WEBSPAM-UK2006 contains 11402 hosts and it is based on a crawl of the .uk domain done in May, 2006.
8045 hosts were manually labeled either as normal, spam or undecided by domain experts.
For this project, undecided-labeled hosts are considered unlabeled, leaving only 7866 labeled samples.

	#	%
Normal	7093	62
Spam	773	7
Unlabeled	3536	31

Obtaining the PageRank transition matrix

Starting from a file containing the host-level web graph, the first step towards PageRank and further computations is extracting the transition matrix R, defined as:

$$R_{i,j} = \begin{cases} 0 & \text{if } i \not\rightarrow j, \\ \frac{1}{O[i]} & \text{if } i \rightarrow j \end{cases}$$

where $O[i]$ is the total number of hosts linked by host i.
To improve efficiency, R can be stored as a `scipy.sparse` matrix.

PageRank

PageRank (Brin and Page, 1998) is probably the most famous web page ranking algorithm and one of the keys to the success of Google.
It can be computed iteratively as:

$$rank_{k+1} = \frac{\alpha}{N}\mathbf{1} + (1-\alpha)R^T \cdot rank_k$$

until convergence ($|rank_k - rank_{k-1}|_1 < \epsilon$).

PageRank

Many studies proved that when performing a web search, users tend to click on one of the first links in the results page, meaning that it is particularly important to assign a high PageRank value to trustworthy pages.
Unfortunately, PageRank alone is not sufficient to discriminate spam and normal hosts.

PageRank

Many studies proved that when performing a web search, users tend to click on one of the first links in the results page, meaning that it is particularly important to assign a high PageRank value to trustworthy pages.
Unfortunately, PageRank alone is not sufficient to discriminate spam and normal hosts.
In fact, after performing PageRank, if we restrict our view to the top 25% highest ranked labeled hosts, 139 on 1966 (7.1%) are spam, a value comparable to the total proportion of spam pages on the entire labeled dataset (9.8%). An additional effort is needed.

Outline

1. The problem: Vulnerability of PageRank

2. Link-based features for spam detection

3. Robust PageRank

PageRank contributions

To move towards spam detection, some additional information is needed. We need to know how much each node contributes to the PageRank of other nodes.

We will store this information in a matrix PRM: $PRM[u, v]$ is the contribution of node u to the PageRank of node v and consequently $\sum_u PRM[u, v] = rank[v]$.
The contribution vector c_v of node v is defined as the v-th column of PRM.

PageRank contributions

PRM can be computed using an iterative approach based on the following equation[1]:

$$PRM = \alpha I + (1 - \alpha) PRM \cdot R$$

However, this method may be very slow and, in the case of a larger dataset, even infeasible for memory limitations. Hence, an alternative algorithm is needed to approximate *PRM*.

[1] Assuming PageRank normalization to N

PageRank contributions

To reduce computational cost, we can compute δ-approximations of contribution vectors for the nodes of interest (in this case, the ones in the labeled set).

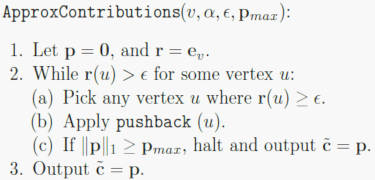
Definition

Given a node v and its contribution vector c_v, a δ-approximation of c_v is a non-negative vector c_v^* such that:

$$c_v[u] - \delta \cdot rank[v] \leqslant c_v^*[u] \leqslant c_v[u] \qquad \forall u$$

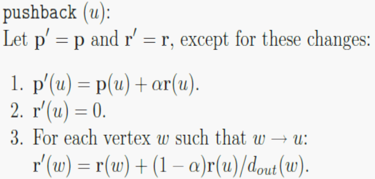
Approximating contributions (Andersen et al., 2007)

The algorithm `ApproxContributions` can be used to approximate contribution vectors.



```
ApproxContributions($v, \alpha, \epsilon, \mathbf{p}_{max}$):

1. Let $\mathbf{p} = \mathbf{0}$, and $\mathbf{r} = \mathbf{e}_v$.
2. While $\mathbf{r}(u) > \epsilon$ for some vertex $u$:
   (a) Pick any vertex $u$ where $\mathbf{r}(u) \geq \epsilon$.
   (b) Apply pushback ($u$).
   (c) If $\|\mathbf{p}\|_1 \geq \mathbf{p}_{max}$, halt and output $\tilde{\mathbf{c}} = \mathbf{p}$.
3. Output $\tilde{\mathbf{c}} = \mathbf{p}$.
```



```
pushback ($u$):
Let $\mathbf{p}' = \mathbf{p}$ and $\mathbf{r}' = \mathbf{r}$, except for these changes:

1. $\mathbf{p}'(u) = \mathbf{p}(u) + \alpha\mathbf{r}(u)$.
2. $\mathbf{r}'(u) = 0$.
3. For each vertex $w$ such that $w \to u$:
   $\mathbf{r}'(w) = \mathbf{r}(w) + (1-\alpha)\mathbf{r}(u)/d_{out}(w)$.
```

To compute a δ-approximation of a contribution vector the call would be `ApproxContributions(v,` α`,` δ `rank[v], rank[v])`. This algorithm is $O(\frac{1}{\alpha\delta})$. All the results in this presentation were obtained with $\alpha = 0.1$ and $\delta = 0.001$.

Features for web spam detection

Once an approximation c_v^* of the contribution vector of a node v is available, some features for spam detection can be computed:

- Indegree: number of incoming links to a host[2]
- Outdegree: number of outgoing links from a host[2]
- Size of δ-significant contributing set: $cs\_size[v] = |S_\delta[v]| = |\{u : c_v^*[u] > \delta rank[v]\}|$
- Contribution from vertices in the δ-significant contributing set: $cs\_contribution[v] = \frac{1}{rank[v]} \sum_{u \in S_\delta[v]} c_v^*[u]$
- l_2 norm of δ-significant contributing vector: $l_2\_norm[v] = \sqrt{\sum_{u \in S_\delta[v]} (\frac{c_v^*[u]}{rank[v]})^2}$

[2] These features only require web graph knowledge

Models and performance evaluation

The features can be used to train and test some simple Machine Learning models for binary spam/non-spam classification.
These models can be assessed using the following performance metrics:

- Accuracy
- Precision on spam class
- Recall on spam class

Models and performance evaluation

Three different binary classifiers were tested on these data: Logistic Regression, Decision Tree and Random Forest, all in their default implementation provided by the library `scikit-learn`.

Models and performance evaluation

Three different binary classifiers were tested on these data: Logistic Regression, Decision Tree and Random Forest, all in their default implementation provided by the library `scikit-learn`.
The only non-default setting for all the three models was `class_weight=balanced` , to try to mitigate the effects of unbalance in data (normal samples are almost 10 times more than spam samples).
All the models were assessed by means of a 5-fold cross-validation.

Models and performance evaluation

Results on full (labeled) dataset:

	Accuracy	Precision	Recall
Logistic Regression	0.71	0.25	0.96
Decision Tree	0.91	0.56	0.55
Random Forest	0.93	0.70	0.55

Models and performance evaluation

Results on top 25% highest ranked labeled dataset:

	Accuracy	Precision	Recall
Logistic Regression	0.94	0.55	0.99
Decision Tree	0.96	0.75	0.68
Random Forest	0.98	0.83	0.81

Outline

1. The problem: Vulnerability of PageRank
2. Link-based features for spam detection
3. Robust PageRank

A robust alternative to PageRank

Another possibility is designing an alternative version of PageRank, intrinsically robust to link spam.
Key properties:

- Robustness: spam pages should be pushed low in the rank
- Invariance: for non-spam pages order should be preserved

A robust alternative to PageRank

Another possibility is designing an alternative version of PageRank, intrinsically robust to link spam.

Key properties:

- Robustness: spam pages should be pushed low in the rank
- Invariance: for non-spam pages order should be preserved

Idea: spam sites tend to have a few very significant contributions.

We can penalize these contributions by defining:

Definition

$$robust\_rank[v] = \sum_{u} min(PRM[u, v], \delta \cdot rank[v])$$

A simple formula for Robust PageRank

Robust PageRank can be rewritten as:

$$robust\_rank[v] = \sum_{u \notin S_\delta[v]} PRM[u,v] + \sum_{u \in S_\delta[v]} \delta \cdot rank[v] =$$

$$= \sum_{u} PRM[u,v] - \sum_{u \in S_\delta[v]} PRM[u,v] + \sum_{u \in S_\delta[v]} \delta \cdot rank[v]$$

Recalling that $\sum_u PRM[u,v] = rank[v]$:

$$robust\_rank[v] = rank[v] - \sum_{u \in S_\delta[v]} PRM[u,v] + \delta \cdot rank[v] \cdot cs\_size[v]$$

A simple formula for Robust PageRank

Finally, $PRM[u,v]$ can be approximated by $c_v^*[u]$ and recalling the definition: $cs\_contribution[v] = \frac{1}{rank[v]} \sum_{u \in S_\delta[v]} c_v^*[u]$:

$$robust\_rank[v] = rank[v] - cs\_contribution[v] \cdot rank[v] + \delta \cdot rank[v] \cdot cs\_size[v]$$

Results and comparison with PageRank

- With Robust PageRank, only 3 (0.2%) of the highest scoring 1966 hosts (25%) are labeled as spam. 7.1% with Normal PageRank.

Results and comparison with PageRank

- With Robust PageRank, only 3 (0.2%) of the highest scoring 1966 hosts (25%) are labeled as spam. 7.1% with Normal PageRank.
- Positions of the 10 highest scoring spam hosts ranges between 1256 and 2198. Between 157 and 834 with Normal PageRank.

Results and comparison with PageRank

- With Robust PageRank, only 3 (0.2%) of the highest scoring 1966 hosts (25%) are labeled as spam. 7.1% with Normal PageRank.
- Positions of the 10 highest scoring spam hosts ranges between 1256 and 2198. Between 157 and 834 with Normal PageRank.
- The two top 25% sets are almost perfectly overlapping: 1717 elements in common out of 1966.

Conclusions and possible improvements

- PageRank is a fast and efficient ranking algorithm but it may be subject to link spam

Conclusions and possible improvements

- PageRank is a fast and efficient ranking algorithm but it may be subject to link spam
- Spam detection by means of link-based feature is effective but even the approximate algorithm for contributions may become slow on huge web graphs

Conclusions and possible improvements

- PageRank is a fast and efficient ranking algorithm but it may be subject to link spam
- Spam detection by means of link-based feature is effective but even the approximate algorithm for contributions may become slow on huge web graphs
- Robust PageRank looks like a valid and alternative to PageRank, once contribution vectors are available

Conclusions and possible improvements

- PageRank is a fast and efficient ranking algorithm but it may be subject to link spam
- Spam detection by means of link-based feature is effective but even the approximate algorithm for contributions may become slow on huge web graphs
- Robust PageRank looks like a valid and alternative to PageRank, once contribution vectors are available
- A faster algorithm for approximating contributions could make these approaches suitable on a larger scale

Thank you for your attention!