Robust PageRank and Locally Computable Spam Detection Features

Vahab Mirrokni [Microsoft Research]
–joint work with–
Reid Andersen [Microsoft Research]
Christian Borgs [Microsoft Research]
Jennifer Chayes [Microsoft Research]
John Hopcroft [Cornell University]
Kamal Jain [Microsoft Research]
Shang-Hua Teng [Boston University]
Outline

- PageRank and PageRank Contributions.
- Applications to Link Spam Detection.
- A Local Algorithm for PageRank Contributions.
- Link Spam Detection Features and Experimental Results.
PageRank measures the importance of nodes in a graph.

PageRank on the web graph:
- Rank pages for a query.
- Priority in web crawls.

PageRank:
- Link Structure.
- PageRank score depends recursively on the PageRank score of incoming neighbors.
Where does the PageRank come from?

PageRank score: the sum of the *PageRank contributions* from other nodes.

**Outgoing contributions:** Each node sends small contributions to the nodes it can reach either directly or indirectly.

**Incoming contributions:** The PageRank of a particular node is the sum of the contributions it receives.
We now define the PageRank vector $\mathbf{pr}(\alpha, s)$.

$s$ is an arbitrary *restarting distribution*

$\alpha$ is the *restarting probability*.

Consider the following random walk in the graph. At each step:

\[
\begin{cases} 
\text{move to a neighbor at random with probability } (1 - \alpha) \\
\text{restart to } s \text{ with probability } \alpha.
\end{cases}
\]

PageRank $\mathbf{pr}(\alpha, s)[v]$ is the stable distribution of the above random walk.
Global PageRank and Personalized PageRank

These are special cases of PageRank, with specific starting distributions.

<table>
<thead>
<tr>
<th>Personalized PageRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>In personalized PageRank for $u$, $s = e_u$ (vector with a one at $u$).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Global PageRank (the usual PageRank)</th>
</tr>
</thead>
<tbody>
<tr>
<td>In PageRank, $s = 1$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relationship between the two</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global PageRank vector = the sum of the personalized PageRank vectors.</td>
</tr>
</tbody>
</table>
Global PageRank and Personalized PageRank

These are special cases of PageRank, with specific starting distributions.

<table>
<thead>
<tr>
<th>Personalized PageRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>In personalized PageRank for $u$, $s = e_u$ (vector with a one at $u$).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Global PageRank (the usual PageRank)</th>
</tr>
</thead>
<tbody>
<tr>
<td>In PageRank, $s = 1$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relationship between the two</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global PageRank vector = the sum of the personalized PageRank vectors.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>The <em>contribution</em> from $u$ to $v$ = the personalized PageRank of $u$ for $v$.</td>
</tr>
</tbody>
</table>
Link Spam and PageRank Contribution

**Link Spam**: Web Spammers abuse the link structure and get high PageRank without introducing new content.
Link Spam: Web Spammers abuse the link structure and get high PageRank without introducing new content.

- The PageRank of a high PageRank non-spam node consists of small contributions from a large set of nodes.
- The PageRank of a high PageRank spam node consists of large contributions from a small set of nodes.
- This has been formally observed by SpamRank [Benczur et al. 05].
Plot of contributions

Contribution Vector of a spam node from UK host graph.

X-axis: Node Number (sorted by contribution)
Y-axis: Contribution
Contribution Vector of a non-spam node from UK host graph.

X-axis: Node Number (sorted by contribution)
Y-axis: Contribution
Identifying top contributors

Problem: Given a page, identify its top contributors.

- Identify the top \( k \) contributors.
- Identify all pages who contribute above a certain threshold (i.e., they have large personalized PageRank to this page).

Our Goal: Approximate the contribution vector to a node, using a local algorithm.

Local Algorithm:

- It examines only a small part of the entire graph.
- It produces a sparse approximate solution. It produces an approximate contribution vector that differs from the true vector of contributions by at most \( \epsilon \) at each node.
Approximate contributions for different $\epsilon$. 

$\epsilon = 0.001$ 

$\epsilon = 0.0005$
Applications for locally computing PageRank contributions

Locally Computable Link Spam Features.
Supervised and Unsupervised features can be computed on-the-fly for a few selected nodes.

Support of Known Spam Pages.
For a spam node, identify its top contributors.
Description of the contribution algorithm

Technique: The *asynchronous pushing method* [Jeh/Widom 03] [McSherry 05] [Berkhin 06]
Description of the contribution algorithm

Technique: The *asynchronous pushing method* [Jeh/Widom 03] [McSherry 05] [Berkhin 06]

The algorithm $\text{Approx Contributions}(v, \alpha, \epsilon)$

**Input:**

$\nu$, the target node

$\alpha$, the PageRank restarting probability

$\epsilon$, the desired error in each entry of the contribution vector

**Output:** An $\epsilon$-approximate contribution vector
Technique: The *asynchronous pushing method* [Jeh/Widom 03] [McSherry 05] [Berkhin 06]

The algorithm $\text{ApproxContributions}(v, \alpha, \epsilon)$

**Input:**
- $v$, the target node
- $\alpha$, the PageRank restarting probability
- $\epsilon$, the desired error in each entry of the contribution vector

**Output:** An $\epsilon$-approximate contribution vector

**pushback Operation:** Push some probability to each in-neighbor.
Description of the contribution algorithm

Technique: The *asynchronous pushing method* [Jeh/Widom 03] [McSherry 05] [Berkhin 06]

The algorithm $\text{ApproxContributions}(v, \alpha, \epsilon)$

**Input:**
$v$, the target node  
$\alpha$, the PageRank restarting probability  
$\epsilon$, the desired error in each entry of the contribution vector

**Output:** An $\epsilon$-approximate contribution vector

**Pushback Operation:** Push some probability to each in-neighbor.

**Running Time:** The number of pushback operations is linear in size of contribution set.
Computing contributions locally

We will maintain two vectors,

- an $\epsilon$-approximate contribution vector $c$, and
- a residual vector $r$.

Let's define the pushback function:

```
pushback(c, r, u)

\[
c'[u] += \alpha \cdot r[u]
\]
\[
r'[u] = 0
\]

for v such that $v \to u$:

\[
r'[v] += (1-\alpha) \cdot r[u] / \text{outdegree}[v]
\]

change $r$ to $r'$ and $c$ to $c'$

Main Loop

While there is a node $u$ where $r(u) > \epsilon$,
pick any such node and perform the push operation.
Computing contributions locally

We will maintain two vectors,

- an $\epsilon$-approximate contribution vector $c$, and
- a residual vector $r$.

\textbf{pushback}(c,r,u)

\begin{align*}
  c'[u] &= \alpha \times r[u] \\
  r'[u] &= 0 \\
  \text{for } v \text{ such that } v \rightarrow u: \quad r'[v] &= (1-\alpha) \times r[u] / \text{outdegree}[v] \\
  \text{change } r \text{ to } r' \text{ and } c \text{ to } c'
\end{align*}
Computing contributions locally

We will maintain two vectors,
- an $\epsilon$-approximate contribution vector $c$, and
- a residual vector $r$.

\begin{verbatim}
pushback(c, r, u)

    c'[u] += alpha * r[u]
    r'[u] = 0
    for v such that v -> u:
        r'[v] += (1-alpha) * r[u] / outdegree[v]

    change r to r' and c to c'
\end{verbatim}

Main Loop

While there is a node $u$ where $r(u) > \epsilon$, pick any such node and perform the push operation.
Example: identifying sets of top contributors locally

target page: www.usajobs.opm.gov/b.htm
desired error: \( \epsilon = .001 \)

Top contributors and their approximate contributions

- 0.206109 www.usajobs.opm.gov/b.htm
- 0.105105 www.rurdev.usda.gov/rbs/oa/jobs.htm
- 0.105105 www.fsa.usda.gov/pas/fsajobs.htm
- 0.0946422 staffing.opm.gov/Immigrationinspector/
- 0.0846548 www.usajobs.opm.gov/survey.htm
- 0.0845882 profiler.usajobs.opm.gov/
- 0.0825384 www.usajobs.opm.gov/a9nasa.htm
- 0.0825384 www.usajobs.opm.gov/a9noaa.htm
- 0.0825086 www.usajobs.opm.gov/wfjic/jobs/T04034.htm
- 0.0825086 www.usajobs.opm.gov/wfjic/jobs/IA2386.htm
- 0.0825086 www.usajobs.opm.gov/wfjic/jobs/IZ9687.htm
- 0.0825086 www.usajobs.opm.gov/wfjic/jobs/IZ9590.htm

- The algorithm examines 5877 vertices.
- It finds 1777 pages that contribute at least \( \epsilon \) to the target.
- It produces an approximate contribution vector where the error in each entry is at most \( \epsilon = 0.001 \).
X-axis: Node Number (sorted by contribution)
Y-axis: Contribution (lower bounds with some error)
log-log plot
X-axis: Error level $\epsilon$ in the contribution vector
Y-axis: number of $\epsilon$-supporters and number of nodes examined
**Definition**

$S_\delta(v)$: The $\delta$-contributing set of a node $v$ is the set of nodes whose contributions to $v$ are at least $\delta \text{pr}(v)$.
Definition

$S_{\delta}(v)$: The $\delta$-contributing set of a node $v$ is the set of nodes whose contributions to $v$ are at least $\delta \text{pr}(v)$.

Supervised features:

- Ratio of spam in contributing set: Ratio of spam and Non-spam nodes in the $\delta$-contributing set.
Definition

$S_\delta(v)$: The $\delta$-contributing set of a node $v$ is the set of nodes whose contributions to $v$ are at least $\delta \text{pr}(v)$.

Supervised features:

- Ratio of spam in contributing set: Ratio of spam and Non-spam nodes in the $\delta$-contributing set.

Unsupervised features:

- Size of the $\delta$-contributing set ($|S_\delta(v)|$).
- $l_1$ and $l_2$ norm of contribution vector of the $\delta$-contributing set.
Robust PageRank

Robust PageRank = sum of truncated contributions.
Generalize Truncated PageRank by Becchetti et al. (2006).
Robust PageRank = sum of truncated contributions.
Generalize Truncated PageRank by Becchetti et al. (2006).

\[
\text{Robustpr}^\delta_\alpha(v) = \sum_{u \in V(G)} \min(\text{ppr}(u, v), \delta)
\]

\[
= \sum_{u \in V(G)} \text{ppr}(u, v) - \sum_{u \in S_\delta(v)} (\text{ppr}(u, v) - \delta)
\]

\[
= \text{pr}_\alpha(v) - \sum_{u \in S_\delta(v)} \text{ppr}(u, v) - \delta|S_\delta(v)|.
\]
Robust PageRank = sum of truncated contributions. 
Generalize Truncated PageRank by Becchetti et al. (2006).

\[
\text{Robustpr}_\alpha^\delta(v) = \sum_{u \in V(G)} \min(\text{ppr}(u, v), \delta) \\
= \sum_{u \in V(G)} \text{ppr}(u, v) - \sum_{u \in S_\delta(v)} (\text{ppr}(u, v) - \delta) \\
= \text{pr}_\alpha(v) - \sum_{u \in S_\delta(v)} \text{ppr}(u, v) - \delta |S_\delta(v)|.
\]

Feature: Ratio between Robust PageRank and PageRank.
Performance of Link Spam Features

Labeled UK Host Graph
11401 nodes, average degree 65,
Examined 24% high PageRank nodes
δ = 10^{-4}, average size of δ-contributing set= 301

<table>
<thead>
<tr>
<th>Feature</th>
<th>FNeg1</th>
<th>FPos1</th>
<th>FNeg2</th>
<th>FPos2</th>
</tr>
</thead>
<tbody>
<tr>
<td>l₁ Norm</td>
<td>6%</td>
<td>5%</td>
<td>67%</td>
<td>2%</td>
</tr>
<tr>
<td>Robust PR</td>
<td>5%</td>
<td>5%</td>
<td>38%</td>
<td>2%</td>
</tr>
<tr>
<td>Indegree (Base)</td>
<td>45%</td>
<td>5%</td>
<td>78%</td>
<td>2%</td>
</tr>
<tr>
<td>PRIndegree (Base)</td>
<td>50%</td>
<td>5%</td>
<td>82%</td>
<td>2%</td>
</tr>
<tr>
<td>Spam in Contrib. (Sup)</td>
<td>4%</td>
<td>5%</td>
<td>15%</td>
<td>2%</td>
</tr>
<tr>
<td>Spam in Neighbors (Base)</td>
<td>8%</td>
<td>5%</td>
<td>33%</td>
<td>2%</td>
</tr>
</tbody>
</table>
Performance of Link Spam Features

Labeled UK Host Graph
11401 nodes, average degree 65,
Examined 24% high PageRank nodes
\(\delta = 10^{-4}\), average size of \(\delta\)-contributing set = 301

<table>
<thead>
<tr>
<th>Feature</th>
<th>FNeg1</th>
<th>FPos1</th>
<th>FNeg2</th>
<th>FPos2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>8%</td>
<td>5%</td>
<td>78%</td>
<td>2%</td>
</tr>
<tr>
<td>(l_1) Norm</td>
<td>6%</td>
<td>5%</td>
<td>67%</td>
<td>2%</td>
</tr>
<tr>
<td>Robust PR PR</td>
<td>5%</td>
<td>5%</td>
<td>38%</td>
<td>2%</td>
</tr>
</tbody>
</table>
Performance of Link Spam Features

Labeled UK Host Graph
11401 nodes, average degree 65,
Examined 24% high PageRank nodes
\( \delta = 10^{-4} \), average size of \( \delta \)-contributing set = 301

<table>
<thead>
<tr>
<th>Feature</th>
<th>FNeg1</th>
<th>FPos1</th>
<th>FNeg2</th>
<th>FPos2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>8%</td>
<td>5%</td>
<td>78%</td>
<td>2%</td>
</tr>
<tr>
<td>( l_1 ) Norm</td>
<td>6%</td>
<td>5%</td>
<td>67%</td>
<td>2%</td>
</tr>
<tr>
<td>Robust PR/PR</td>
<td>5%</td>
<td>5%</td>
<td>38%</td>
<td>2%</td>
</tr>
<tr>
<td>Indegree (Base)</td>
<td>45%</td>
<td>5%</td>
<td>78%</td>
<td>2%</td>
</tr>
<tr>
<td>PRIndegree (Base)</td>
<td>50%</td>
<td>5%</td>
<td>82%</td>
<td>2%</td>
</tr>
</tbody>
</table>
Performance of Link Spam Features

Labeled UK Host Graph
11401 nodes, average degree 65,
Examined 24% high PageRank nodes
\( \delta = 10^{-4} \), average size of \( \delta \)-contributing set = 301

<table>
<thead>
<tr>
<th>Feature</th>
<th>FNeg1</th>
<th>FPos1</th>
<th>FNeg2</th>
<th>FPos2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>8%</td>
<td>5%</td>
<td>78%</td>
<td>2%</td>
</tr>
<tr>
<td>( l_1 ) Norm</td>
<td>6%</td>
<td>5%</td>
<td>67%</td>
<td>2%</td>
</tr>
<tr>
<td>Robust PR</td>
<td>5%</td>
<td>5%</td>
<td>38%</td>
<td>2%</td>
</tr>
<tr>
<td>PR</td>
<td>5%</td>
<td>5%</td>
<td>38%</td>
<td>2%</td>
</tr>
<tr>
<td>Indegree (Base)</td>
<td>45%</td>
<td>5%</td>
<td>78%</td>
<td>2%</td>
</tr>
<tr>
<td>PRIndegree (Base)</td>
<td>50%</td>
<td>5%</td>
<td>82%</td>
<td>2%</td>
</tr>
<tr>
<td>Spam in Contrib. (Sup)</td>
<td>4%</td>
<td>5%</td>
<td>15%</td>
<td>2%</td>
</tr>
<tr>
<td>Spam in Neighbors (Base)</td>
<td>8%</td>
<td>5%</td>
<td>33%</td>
<td>2%</td>
</tr>
</tbody>
</table>
Other Related Work

- Topic-sensitive PageRank [Haveliwala 03],
- TrustRank [Gyongyi et al. 04],
- Anti-TrustRank [Raj et al. 99],
- SpamMass algorithm [Gyongyi et al. 06].

  The PageRank of a node can be estimated within a smaller subgraph containing its large contributors [Chen et al. 04].
Thank You