Beyond PageRank: Machine Learning for static Ranking

Authors: M. Richardson, A. Prakash, E. Brill (Microsoft Research)

Speakers: M. Innerebner, P. Miksys
1. Motivation
2. Problem Definition / Contribution
3. Related Work
4. fRank
5. Experiments
   1. Data Preparation, Measurement, Method
   2. Results
6. Conclusion / Future Work
7. Paper Valuation
Goals:

Outperform PageRank in:
- quality of static ranking
- computation time for rank calculation
- hampering high ranking manipulation of spamming or other malicious pages
Problem with PageRank:

- Links not the only factor to achieve good ranking results
  - topic drift: referenced links of different context/domain
  - intranet ranking: link based ranking not suitable
- PageRanking is expensive
  - requires big resources (large memory)
  - less performance improvement, even if a lot of investigation (still slow and expensive)
  - does not cover all pages in the web
- Model is easier attackable from spammers
  - only focusing on how to increase incomming links
Contribution

**Improvement:**

- Introduce an new ranking approach based on static ranking that:
  - uses features as input values
  - is qualitative better than PageRanking (based on results)
  - can be used for ranking within a non public network
  - is faster than PageRanking
  - can be used for crawl prioritization (filtering of low ranked pages, increase update on high ranked pages)
  - an incremental model (easily expandable in case of a spammer attack)
- Illustrate results in a detailed experimental study
Related Work

**PageRank:**

- Link from web page to another can be seen as an endorsement of that page
- PageRank formula:
  \[
  P(j) = \frac{\alpha}{N} + (1 - \alpha) \sum_{i \in B} \frac{P(i)}{|F_i|}
  \]
  - \(P(j)\) – PageRank score for node \(j\)
  - \(P(i)\) – PageRank score for node \(i\)
  - \(F_i\) – set of pages that page \(i\) links to
  - \(B_j\) – set of pages that link to page \(j\)
  - \(N\) – total number of pages
  - \(\alpha\) – probability to jump to random page

- Weaknesses:
  - easy manipulate for malicious users
  - not enough to get good ranking results
  - expensive in computation
Static Ranking

- General indicator that defines the overall quality of the page
- Ranking independent from the user query
**fRank (feature based ranking)**

- uses the same technique of RankNet, but:
  - input pairs are two feature vectors (instead of real value)
- features are extracted from the web page and assigned to one of the following categories:
  - page-level
  - domain-level
  - anchor text and inlinks
  - popularity
  - PageRank
Basic Idea:

- for each page we apply a regression function that maps its feature vector to a corresponding real value (rank)

\[ f : \mathbb{R}^d \rightarrow \mathbb{R} \]

- as result we have an ordered set of page \((S_p)\)

\[ \forall (i, j) \in Z, f(X_i) > f(X_j) \]

- based on an existing ordering of pages (a subset, judged by humans) the function is trained and optimized to generate a static rank, where its order corresponds to the order in the human judged set

\[ H_p = \{(i, j): H(i) \geq H(j)\} \quad H(i) \sim \text{maximum of human judgement of the relevance for page } i \]

\[ S_p = \{(i, j): S(i) \geq S(j)\} \quad S(i) \sim \text{static ranking assigned to page } i \]
fRank (RankNet)

- A Neural Network is used to train fRank
Input Features

Page (page level)
features which may be determined by looking at the URL alone, like
  • number of words in the body
  • frequency of most common terms

Domain (domain level)
features, that are computed as averages across all pages in the domain, i.e.:
  • average number of outlinks on any page
  • average PageRank
Input Features (2)

Anchor (anchor text and inlinks)

features based on information associated with links to the page in question, like:

• total amount of text in links pointing to the page
• the number of unique words in that text
Input Features (3)

Popularity

features based on the popularity of a page, like:
  • how many times a page was visited from users over a time period

Popularity is measured by:
  • MSN toolbar
  • proxy logs
  • records in a result of query, that are clicked on (measured internally in search engines)
Input Features (4)

PageRank

- computed on a large Web graph with:
  - 5 billions crawled pages
  - 370 billions of links
  - approximately same number of pages as used by Google, Yahoo, MSN
Data preparation

Huge test data set (28,000 queries of human judgement):

- queries were selected randomly
- documents to be judged were selected from authors (i.e. the most relevant for the authors, i.e. first 10 hits)
- more likely common queries than uncommon queries (tendency of qualitatively better results)
- judgement rating between 0 (poor) and 4 (excellent)

quality of input data is considerably better than PageLink's
Experiments

Data usage

Queries where randomly assigned to:
  • Training set (84%): for training fRank
  • Validation set (8%): for selecting the most accurate model
  • Testing set (8%): used to evaluate the best model

  • each set contains all of the ratings for a given query
  • no query appears more than in one set
Data transformation

How to make the human judged ranking query independent?

• for a URL that appears in more than one query take the maximum of judgement
• therefore common pages get a higher weight than uncommon pages

➢ right index ordering of data (roughly corresponds to the index of the result of a Web crawler)
Experiments

Quality Measurement

Quality of fRank measured by *Pairwise Accuracy* (PA)

\[
PA = \frac{|H_p \cap S_p|}{|H_p|}
\]

PA is the portion of \( H_p \) that is also contained in \( S_p \)

\( H_p \sim \) Human ranking
\( S_p \sim \) Static ranking

- PA is the fraction of pairs of documents: when humans claim one is better than the other, the static rank algorithm orders them correctly
Experiments

Method

fRank is divided in several steps:

- Training phase:
  1. Model testing step: testing the model
  2. Validation step: compare static ranking with juman judged ranking
  3. Adjustment step: input weight (TR) adjusted to obtain better results in the next cycle

\[
TR = \frac{K}{\varepsilon + 1}
\]

- Repeat the training until obtained a good static ranking
- Final evaluating that model with the highest PA

\[ K \sim \text{initial rate (0.0001)} \]
\[ \varepsilon \sim \text{number of times the training set error increased} \]
Result: fRank vs. PageRank

- Comparison of fRank and PageRank accuracy

<table>
<thead>
<tr>
<th>Technique</th>
<th>Accuracy (%)</th>
<th>Accuracy Increasing (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (baseline)</td>
<td>50.00</td>
<td></td>
</tr>
<tr>
<td>PageRank</td>
<td>56.70</td>
<td>13.40</td>
</tr>
<tr>
<td>fRank</td>
<td><strong>67.43</strong></td>
<td><strong>34.86</strong></td>
</tr>
</tbody>
</table>

- fRank more than doubles the accuracy of PageRank
Result: Comparison of individual features

- Take every single feature set individually and measure the accuracy

<table>
<thead>
<tr>
<th>Feature</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page</td>
<td>63.93</td>
</tr>
<tr>
<td>Popularity</td>
<td>60.82</td>
</tr>
<tr>
<td>Anchor</td>
<td>59.09</td>
</tr>
<tr>
<td>Domain</td>
<td>59.03</td>
</tr>
<tr>
<td>PageRank</td>
<td>56.70</td>
</tr>
<tr>
<td>All Features</td>
<td>67.43</td>
</tr>
</tbody>
</table>

- Page-level and popularity features are the most important factors
Result: Ablation Comparison (amputation)

• consider all features, amputate the specified feature and measure the accuracy:

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Decrease in Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page</td>
<td>5.42</td>
</tr>
<tr>
<td>Popularity</td>
<td>0.78</td>
</tr>
<tr>
<td>Anchor, PageRank &amp; Domain</td>
<td>0.60</td>
</tr>
<tr>
<td>Anchor</td>
<td>0.47</td>
</tr>
<tr>
<td>PageRank</td>
<td>0.18</td>
</tr>
<tr>
<td>Domain</td>
<td>0.10</td>
</tr>
</tbody>
</table>

➢ the highest accuracy is in Page and Popularity features
➢ the lowest accuracy is in PageRank and Domain features
Result: Comparison

- Comparison of previous studies

<table>
<thead>
<tr>
<th>Feature</th>
<th>Accuracy (%)</th>
<th>Decrease in Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page</td>
<td>63.93</td>
<td>5.42</td>
</tr>
<tr>
<td>Popularity</td>
<td>60.82</td>
<td>0.78</td>
</tr>
<tr>
<td>Anchor</td>
<td>59.09</td>
<td>0.47</td>
</tr>
<tr>
<td>PageRank</td>
<td>56.70</td>
<td>0.18</td>
</tr>
<tr>
<td>Domain</td>
<td>59.03</td>
<td>0.10</td>
</tr>
<tr>
<td>All Features</td>
<td>67.43</td>
<td></td>
</tr>
</tbody>
</table>

- Features with biggest impact are the same:
  - page
  - popularity
Result: Greedy comparison

- Find at what point adding more feature sets becomes relatively useless

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Accuracy (%)</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>+Page</td>
<td>63.93</td>
<td>13.93</td>
</tr>
<tr>
<td>+Popularity</td>
<td>66.83</td>
<td>2.9</td>
</tr>
<tr>
<td>+Anchor</td>
<td>67.25</td>
<td>0.42</td>
</tr>
<tr>
<td>+PageRank</td>
<td>67.31</td>
<td>0.06</td>
</tr>
<tr>
<td>+Domain</td>
<td>67.43</td>
<td>0.12</td>
</tr>
</tbody>
</table>

- Adding the features PageRank and Domain causes only a marginal impact
Top 10 URLs for PageRank and fRank:

<table>
<thead>
<tr>
<th>PageRank</th>
<th>fRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.com</td>
<td>google.com</td>
</tr>
<tr>
<td>apple.com/quicktime/download</td>
<td>yahoo.com</td>
</tr>
<tr>
<td>amazon.com</td>
<td>americanexpress.com</td>
</tr>
<tr>
<td>yahoo.com</td>
<td>hp.com</td>
</tr>
<tr>
<td>microsoft.com/windows/ie</td>
<td>target.com</td>
</tr>
<tr>
<td>apple.com/quicktime</td>
<td>bestbuy.com</td>
</tr>
<tr>
<td>mapquest.com</td>
<td>dell.com</td>
</tr>
<tr>
<td>ebay.com</td>
<td>autotrader.com</td>
</tr>
<tr>
<td>mozilla.org/products/firefox</td>
<td>dogpile.com</td>
</tr>
<tr>
<td>ftc.gov</td>
<td>bankofamerica.com</td>
</tr>
</tbody>
</table>

- fRank contains more consumer-oriented pages
- PageRank weighted towards technology pages
- Duplicate domains in PageRanking (Apple)
**Popularity Data:**

- Popularity data comes from MSN toolbar users
- URL functions used to compute Popularity data:

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact URL</td>
<td>cnn.com/2005/tech/wikipedia.html?v=mobile</td>
</tr>
<tr>
<td>No Params</td>
<td>cnn.com/2005/tech/wikipedia.html</td>
</tr>
<tr>
<td>Page</td>
<td>wikipedia.html</td>
</tr>
<tr>
<td>URL-1</td>
<td>cnn.com/2005/tech</td>
</tr>
<tr>
<td>URL-2</td>
<td>cnn.com/2005</td>
</tr>
<tr>
<td>Domain</td>
<td>cnn.com</td>
</tr>
<tr>
<td>Domain+1</td>
<td>cnn.com/2005</td>
</tr>
</tbody>
</table>

- Effect of adding backoff to the popularity feature set

<table>
<thead>
<tr>
<th>Features</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL count</td>
<td>58.15</td>
</tr>
<tr>
<td>URL and Domain counts</td>
<td>59.31</td>
</tr>
<tr>
<td>All backoff functions</td>
<td>60.82</td>
</tr>
</tbody>
</table>
Summary of results and of the paper

- fRank performs significantly better than PageRank
  - accuracy: increasing of 21.6%
  - computation: 100 times faster (only linear to the number of input features) for ranking 5 billions Web pages
- Page level and Popularity are the most significant contributors to pairwise accuracy
- By collecting more popularity data, we can continue to improve fRank's performance
- PageRank can be improved by ignoring pages with a very low static rank
Future work:

- Add more features to fRank
  - complexity of the page
  - the number of non-terminal nodes
- Investigate a machine learning for crawl prioritization
- Incorporate fRank into the PageRank computation
- Divide popularity data into several segments
- Explore users behaviour in the page:
  - spent time in page
  - the way of leaving page
Positive points:

- Overall good understandable
- Qualitative good paper, because of
  - Detailed experiment section
  - Approach testified with real world data
Paper Valuation

Negative points:

✗ fRank approach in comparison to PageRank's is completely different
  ✗ to say that it outperforms if they follow a different target
✗ Background knowledge explained less
✗ Weak explanation how fRank works and what are the main differences between this paper and RankNet
✗ Some errors in the formula
✗ Missing some measurements
  ✗ how less memory required in comparison to PageRank