11-442 / 11-642 / 11-742:
Search Engines

Feature-Based Retrieval and Authority Metrics

Jamie Callan and Chenyan Xiong
Carnegie Mellon University
callan@cs.cmu.edu

1

Outline

- Introduction to feature-based methods
- Three approaches to training learning algorithms
  - Pointwise
  - Pairwise
  - Listwise
- **Benchmark datasets**
- Sample results

2

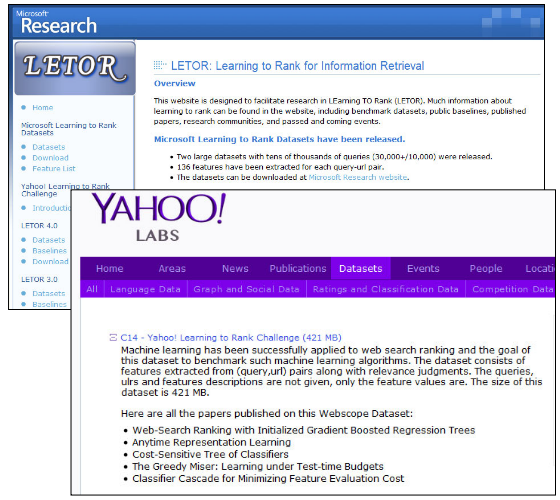
Benchmark Datasets

Three popular LeToR benchmark datasets

1. The LeToR collection (Microsoft) (2007)
2. Yahoo! Challenge datasets (2010)
3. Microsoft Learning to Rank datasets (2010)

Old, but still used in research publications

- There aren't many newer datasets
- Newer datasets have the same characteristics



3

3

Benchmark Dataset #1: The LeToR Collections

Created by Microsoft using existing publicly-available data

- **The .gov corpus**
  - 1M .gov web documents from 2002
  - 350 queries (topic, homepage, named page)
  - Top 1000 documents per query returned by BM25
  - 64 features
- **The OHSUMED corpus**
  - 350K PubMed abstracts from 1987-1991
  - 106 queries (informational)
  - All judged documents
  - 40 features

(http://research.microsoft.com/en-us/projects/mslr/feature.aspx)

(Qin, et al., 2010)

4

4

Benchmark Dataset #1: The LeToR Collections

Field-specific features (title, anchor, url, whole) for (*d*, *q*)

- Tf: Sum over all query terms: $\sum_{q_i \in q} tf_{q_i}$ $tf_{apple} + tf_{pie} + tf_{recipe}$
- Idf: Sum over all query terms: $\sum_{q_i \in q} idf_{q_i}$
- Tf × Idf: Sum over all query terms: $\sum_{q_i \in q} tf_{q_i} \times idf_{q_i}$
- Field length
- Retrieval model scores
  - Boolean, VSM, BM25, LM_{Abs}, $LM_{Dirichlet}$, LM_{JM}
- Hyperlink-based features, HITS, Topic-specific PageRank, …

(Qin, et al., 2010)

5

5

Benchmark Dataset #1: The LeToR Collection

Document-level features for (*d*, *q*)

- PageRank, number of inlinks
- Url: Number of '/', length
- Number of child pages

Note that these features do not depend on *q*

- These features prefer certain types of pages

(Qin, et al., 2010)

6

6

Benchmark Dataset #2:
The Yahoo Challenge! Datasets

Created by Yahoo using proprietary data

- **Set 1**
  - 710K feature vectors
  - 30K queries
  - 519 features (not described)
  - Relevance scale with 5 values
- **Set 2**
  - 173K feature vectors
  - 6K queries
  - 596 features (not described)
  - Relevance scale with 5 values

(Liu, 2011)

Benchmark Dataset #3:
Microsoft Learning to Rank (MSLR) Dataset

Created by Microsoft using proprietary data

- 3.7M web documents
- 30K queries
- 136 features
- Relevance scale with 5 values

Example data

```
0   qid:1   1:3 2:0 3:2 4:2 ... 135:0 136:0
2   qid:1   1:3 2:3 3:0 4:0 ... 135:0 136:0
```

relevance **query id** **features**

(http://research.microsoft.com/en-us/projects/mslr/)

Benchmark Dataset #3: Microsoft Learning to Rank Dataset

Field-specific features (title, anchor, body, url, whole) for (*d*, *q*)

- Covered query term number, covered query term ratio
  - Terms in query q that appear in d
- Field length, field length normalized in various ways
- Idf
- Tf: Min, Max, Sum, Mean, Variance
  - "apple pie recipe": Min (tf_{apple} tf_{pie} tf_{recipe})
- Tf × idf: Min, Max, Sum, Mean, Variance
- Retrieval model scores: Boolean, VSM, BM25, $LM_{Dirichlet}$, LM_{JM}

(http://research.microsoft.com/en-us/projects/mslr/feature.aspx)

9

9

Benchmark Dataset #3: Microsoft Learning to Rank Dataset

Document-level features for (*d*, *q*)

- Url: Number of '/', length
- Number of inlinks and outlinks
- PageRank, SiteRank, url click count, url dwell time
- Two quality scores
- Query-url click count

Note that these features do not depend on *q*

- These features prefer certain types of pages

(http://research.microsoft.com/en-us/projects/mslr/feature.aspx)

10

10

Outline

Introduction to feature-based methods

Three approaches to training learning algorithms

- Pointwise
- Pairwise
- Listwise

Benchmark datasets

Sample results

Sample Experimental Results

LeToR Benchmark (.gov dataset, topic distillation (TD) queries)

	Algorithm	NDCG@			P@			MAP
		1	3	10	1	3	10	
Pointwise	Regression	0.320	0.307	0.326	0.320	0.260	0.178	0.241
Pairwise	RankSVM	0.320	**0.344**	0.346	0.320	**0.293**	0.188	0.263
	RankBoost	0.280	0.325	0.312	0.280	0.280	0.170	0.227
	FRank	0.300	0.267	0.269	0.300	0.233	0.152	0.203
Listwise	ListNet	**0.400**	0.337	**0.348**	**0.400**	**0.293**	**0.200**	**0.275**
	AdaRank	0.260	0.307	0.306	0.260	0.260	0.158	0.228
	SVM^{Map}	0.320	0.320	0.328	0.320	0.253	0.170	0.245

RankSVM is similar to ListNet except at rank 1

(Liu, 2011)

Sample Experimental Results

LeToR Benchmark (.gov dataset, named page (NP) queries)

	Algorithm	NDCG@ 1	NDCG@ 3	NDCG@ 10	P@ 1	P@ 3	P@ 10	MAP
Pointwise	Regression	0.447	0.614	0.665	0.447	0.220	0.081	0.564
Pairwise	RankSVM	0.580	0.765	0.800	0.580	**0.271**	0.092	0.696
	RankBoost	**0.600**	0.764	**0.807**	**0.600**	0.269	**0.094**	**0.707**
	FRank	0.540	0.726	0.776	0.540	0.253	0.090	0.664
Listwise	ListNet	0.567	0.758	0.801	0.567	0.267	0.092	0.690
	AdaRank	0.580	0.729	0.764	0.580	0.251	0.086	0.678
	SVM^{Map}	0.560	**0.767**	0.798	0.560	0.269	0.089	0.687

RankSVM, and RankBoost are best

(Liu, 2011)

Lessons Learned

Observations about the effectiveness of different algorithms

- Many learning algorithms perform relatively well
- Relative effectiveness: Listwise ≈ Pairwise > Pointwise
  - As expected

Many ML algorithms work with pointwise & pairwise LeToR

- Easy to develop, reasonably effective

Listwise algorithms may be more effective

- But, there are fewer off-the-shelf solutions
- Still an open research topic

Lessons Learned

Much of the LeToR literature uses lots of training data

- Research is driven by web companies that have a lot of data
- But … you may not have a lot of data
  - Their conclusions may not apply to your situation

Lessons Learned

Observation from academic research

- 100-200 labeled queries can support 50-100 features
  - Surprising compared to other classification/regression tasks, which need a higher ratio of examples to features
- The theory behind LeToR is still an open research topic

Use large numbers of features cautiously

- Start with a small set of high-quality features, then grow it
- Design features carefully

Lessons Learned

Much research is driven by machine learning researchers …their focus is on machine learning algorithms

The features used in most systems are surprisingly simple

- Simple statistics (e.g., tf, idf, tf × idf, field length)
- Obvious variations of existing ranking algorithms
- A few page quality metrics

Better understanding of search can produce better features … and better search accuracy

- A nice opportunity for future improvement

(Liu, 2011)

Outline

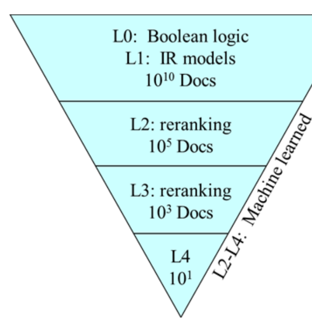
Introduction to feature-based methods

Three approaches to training learning algorithms

- Pointwise
- Pairwise
- Listwise

Benchmark datasets

Sample results



11-442 / 11-642 / 11-742:
Search Engines

Authority Metrics

Jamie Callan
Carnegie Mellon University
callan@cs.cmu.edu

Introduction

Until now, retrieval models considered mostly only page content

- Title, url, meta, body
- We also considered inlink text, which is provided by other pages

Similar content from different sources has different value

- Consider two pages with advice about how to treat a cold
  - A famous medical site
  - Some unknown individual's web page
- Which would you trust more?

Today's topic: Authority metrics

- Which information sources are more trustworthy

Outline

We will cover two authority metrics

- PageRank
- HITS

Goals

- Provide familiarity with some widely-known metrics
- Illustrate issues that authority metrics must address

21

PageRank

PageRank is a metric for estimating a web page's importance

- Developed by Larry Page (Google co-founder)
- PageRank is related to citation analysis in Library Science
  – Which scientific journals or authors have the greatest impact

PageRank is <u>query-independent</u>

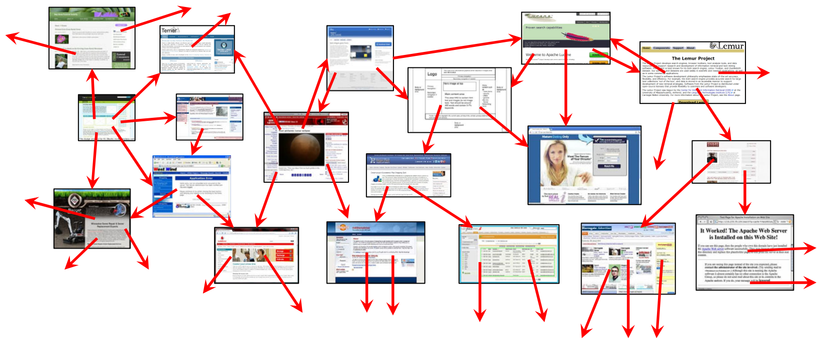
- The Kanye West Wikipedia page has high PageRank
  … but it isn't a good choice for the query "obama family tree"

22

PageRank

How does it work?

- Web pages contain hyperlinks to other web pages
- These links form a directed graph ("the web graph")



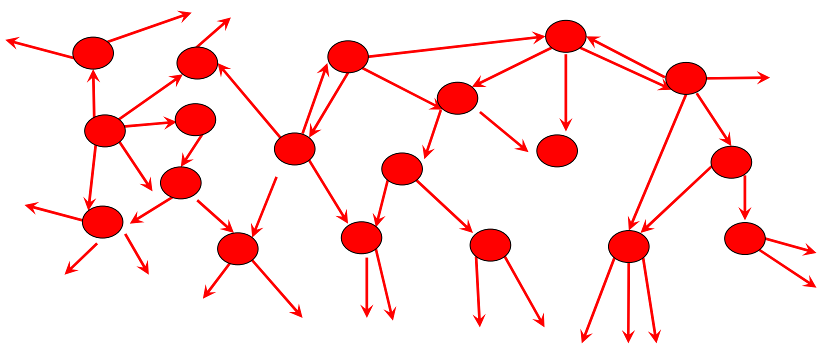
23

23

PageRank

How does it work?

- Web pages contain hyperlinks to other web pages
- These links form a directed graph ("the web graph")



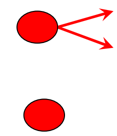
24

24

PageRank

PageRank can be viewed as a random walk algorithm

- Start at an arbitrary web page
- When viewing a page *w* that has *n* outlinks, there are two possible next steps
  - – Randomly follow one of the outlinks to the next page
  - – Randomly select some other page in the dataset ("teleportation")
- Repeat (many, many times)



Over time, some pages are reached more often than other pages

- These pages are more central, and are considered more important

25

25

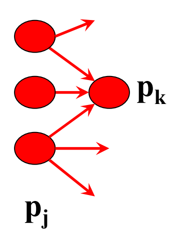
PageRank

Transitions

- The probability of reaching a page p_k is

$$PR(p_k) = \frac{(1-d)}{|C|} + d \sum_{p_j \in \text{InLinks}(p_k)} \frac{PR(p_j)}{|\text{OutLinks}(p_j)|}$$

- d: The damping factor, e.g., d=0.85
- $(1-d)$: The probability of teleporting to a random page
- $|C|$: The size of the corpus



(Brin and Page, 1998)

26

26

PageRank

PageRank can be viewed as a voting algorithm

While (Not done)

For each page p

p votes for each page that it links to

The key idea is how many votes a page p is allowed to cast

- In iteration 1, each page casts the same number of votes
- In iteration i, a page casts the number of votes it received in iteration i-1
  - I.e., popular pages get to cast more votes

PageRank

Voting

- On each iteration, page p_j is allowed to make $PR(p_j)$ votes

$$PR(p_k) = \frac{(1-d)}{|C|} + d \sum_{p_j \in \text{InLinks}(p_k)} \frac{PR(p_j)}{|\text{OutLinks}(p_j)|}$$

d: a damping factor (smoothing), e.g., d=0.85

|C|: size of the corpus

- p_j divides its votes equally among every page that it links to
- Consider two pages that have equal PageRank in iteration i
  - $PR(p_1) = 0.4$. p_1 has 2 outlinks. Each vote by p_1 is $0.4/2 = 0.2$.
  - $PR(p_2) = 0.4$. p_2 has 4 outlinks. Each vote by p_2 is $0.4/4 = 0.1$.

PageRank

PageRank can be calculated with a simple iterative algorithm

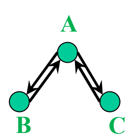
For each page p
  current PR = 1 / |C| C: Number of nodes in the graph
  next PR = 0
While (Not done)
  For each page p
    use p's <u>current</u> PR to update the <u>next</u> PR of each outlink page
  For each page p
    current PR = next PR
    next PR = 0

It can also be calculated using matrix multiplication

PageRank

Consider a web graph with just 3 pages

- A has outlinks to B and C
- B has an outlink to A
- C has an outlink to A



PageRank computation with d=0.5

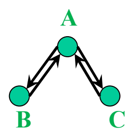
- 14 iterations to converge
  – At 4 decimal places

i	PR(A)	PR(B)	PR(C)
0	0.3333	0.3333	0.3333
1	0.5000	0.2500	0.2500
2	0.4167	0.2917	0.2917
3	0.4583	0.2708	0.2708
4	0.4375	0.2813	0.2813
5	0.4479	0.2760	0.2760
6	0.4427	0.2786	0.2786
7	0.4453	0.2773	0.2773
8	0.4440	0.2780	0.2780
:	: :	: :	: :
14	0.4444	0.2778	0.2778

PageRank

Consider a web graph with just 3 pages

- A has outlinks to B and C
- B has an outlink to A
- C has an outlink to A



PageRank computation with d=0.85

- 58 iterations to converge
  – At 4 decimal places

i	PR(A)	PR(B)	PR(C)
0	0.3333	0.3333	0.3333
1	0.6167	0.1917	0.1917
2	0.3758	0.3121	0.3121
3	0.5805	0.2097	0.2097
4	0.4065	0.2967	0.2967
5	0.5544	0.2228	0.2228
6	0.4287	0.2856	0.2856
7	0.5356	0.2322	0.2322
8	0.4448	0.2776	0.2776
:	: :	: :	: :
58	0.4865	0.2568	0.2568

PageRank Range

PageRank varies over a wide range

Usually the range is compressed and transformed in some way

- E.g., $PR_T = \log_{10}(PR) + 11$
- You could use other functions
- In the past, Google reported a range of 1-10

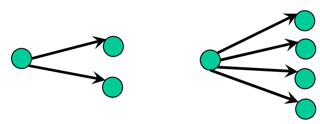
PR	PR	PR_T
0.00000001	1.0E-08	3
0.000001	1.0E-06	5
0.0001	1.0E-04	7
0.01	1.0E-02	9

When people say "PageRank", usually they mean "some transformation of PageRank"

PageRank Observations

What produces a high PageRank for page p?

- Many inlinks (obviously)
- Many inlinks from high PageRank pages
- The pages that link to p have few outlinks
  - During propagation, a page's PR is divided among its outlinks

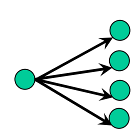


  - I.e., an inlink from a large directory is not very helpful

PageRank Observations

There are many variations on the basic PageRank algorithm

- E.g., should links among pages on the same site count?
  - That makes it easier to manipulate PR for some pages
- E.g., what is the PageRank of new pages?
  - Should they inherit some PageRank from the host?
- E.g., how to handle 'sinks'
  - pages or page groups that have no outgoing links
- E.g., handling link farms and link exchanges



PageRank is topic-independent

- A page may have high PR but be a bad choice for this query

Outline

- PageRank
- **HITS**

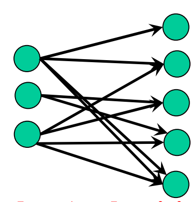
35 © 2021, Jamie Callan

35

Hyperlink-Induced Topic Search (HITS)

Two important types of web pages

- **Hub:** A page that points to pages with good content
- **Authority:** A page that has good content



Initialize *H(p)*=1 and *A(p)*=1 for each page

Hub and Authority scores are calculated iteratively

$$H(p_k) = \sum_{p_j \in OutLinks(p_k)} A(p_j)$$
$$A(p_k) = \sum_{p_j \in InLinks(p_k)} H(p_j)$$

Normalize scores at the end of each iteration

- Divide by $\sqrt{\sum H(p)^2}$ and $\sqrt{\sum A(p)^2}$

(Kleinberg, 1999)

36 © 2021, Jamie Callan

36

Hyperlink-Induced Topic Search (HITS)

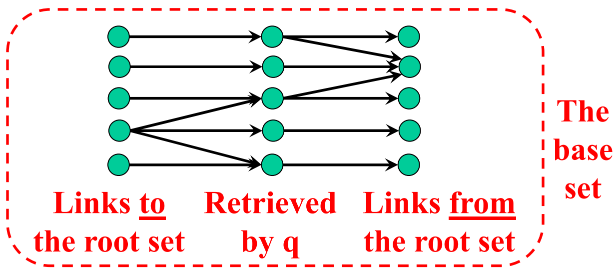
Hubs & authority scores are not calculated over the entire web

Obtain the top n pages for query q ("the root set")

… expand it with some of the pages that point into the root set

… expand it with pages that the root set points to

… calculate hubs and authorities scores over this set

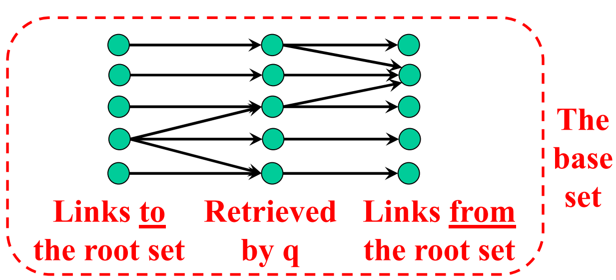


(Kleinberg, 1999)

Hyperlink-Induced Topic Search (HITS)

Notable characteristics

- The base set has a strong, query-specific focus
- The base set is relatively small (so the calculation is efficient)
  - E.g., perhaps 200 pages
- Scores are calculated at query time (so efficiency is important)



(Kleinberg, 1999)

Hyperlink-Induced Topic Search (HITS)

HITS isn't used much in large-scale search engines

- It is a little easier to spam than PageRank
  - – E.g., it is easy to create a page with a high hub score
- Its run-time costs are higher than PageRank

It is often used for other purposes

- E.g., to find communities
  - – They tend to have tightly-bound hubs and authorities
- E.g., finding experts within a community

Outline

- **PageRank**
- **HITS**

Summary

Authority metrics are an important component of web ranking

- Exactly how important is a topic of much debate
- Exactly how it is used is also a topic of much debate

This remains an active area of research

- Spammers and other bad guys keep adapting
- The range of factors that must be considered keeps growing

For More Information

Learning to rank

- B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Addison Wesdley. 2010.
- T.-Y. Liu. *Learning to Rank for Information Retrieval*. Springer. 2011.
- T. Qin, T.-Y. Liu, J. Xu, and H. Li. LETOR: A Benchmark Collection for Research on Learning to Rank for Information Retrieval. *Information Retrieval Journal*. 2010.
- F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank – Theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*. 2008.

Authority metrics

- S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 1998.
- J. M. Kleinberg. Hubs, authorities, and communities. *ACM Computing Surveys*. 1999.