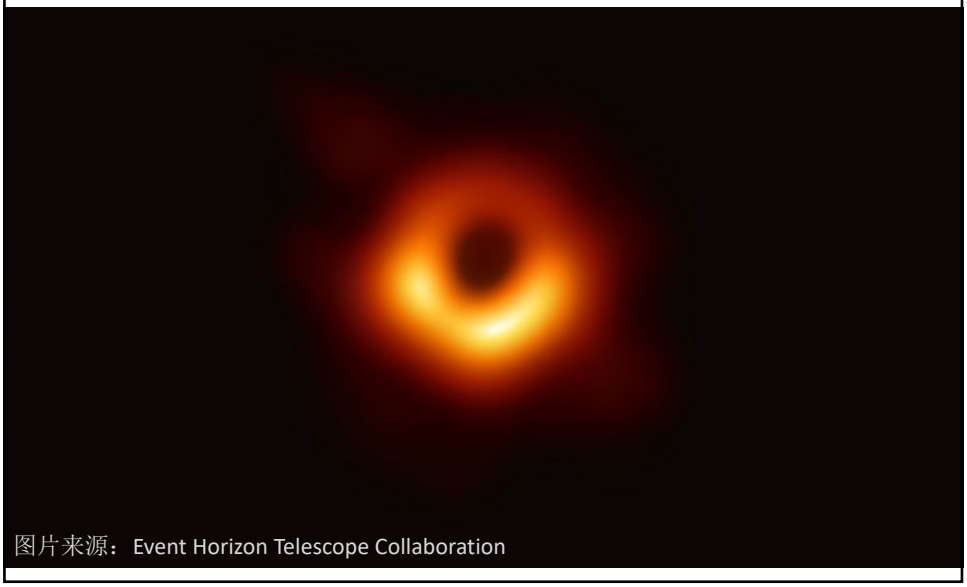
Welcome to the class of

Advanced Topics in Information Retrieval !

Min ZHANG （张敏）
z-m@tsinghua.edu.cn

Tea Time (1): The 1st Image of Black Hole



图片来源：Event Horizon Telescope Collaboration

Tea Time （2）

检索和推荐应用中的Privacy问题

Shao Yunqiu 邵韵秋

Introduction on IR fundamental techniques (IV) – Evaluation (Cont.)

Evaluation

- Evaluation methodology
- Standard benchmarks for IR
- Annotation consistency
- Common metrics for relevance

- Precision, Recall, MAP, MRR, F
- DCG and NDCG

Discounted Cumulative Gain

- Popular measure for evaluating web search and related tasks
- Two assumptions:
  - Highly relevant documents are more useful than marginally relevant document
  - The lower the ranked position of a relevant document, the less useful it is for the user, since it is less likely to be examined

Discounted Cumulative Gain

- Uses *graded relevance* as a measure of the usefulness, or *gain,* from examining a document
  - $rel_1 + rel_2 + rel_3 + \ldots$
- Gain is accumulated starting at the top of the ranking and may be reduced, or *discounted*, at lower ranks
  - $rel_1 + \text{discounted}(rel_2) + \text{discounted}(rel_3) + \ldots$
  - Typical discount is $1/log\ (rank)$
  - With base 2, the discount at rank 4 is 1/2, and at rank 8 it is 1/3
  - $rel_1 + rel_2 / \log_2 2 + rel_3 / \log_2 3 + \ldots$

Discounted Cumulative Gain

- *DCG* is the total gain accumulated at a particular rank *p*:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

- Alternative formulation:

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log(1+i)}$$

  - Popularly leveraged by web search companies
  - Emphasis on retrieving highly relevant documents

DCG Example

- 10 ranked documents judged on 0-3 relevance scale:

  3, 2, 3, 0, 0, 1, 2, 2, 3, 0

- Discounted gain: $(1/\log_2 i)$

  3, 2/1, 3/1.59, 0, 0, 1/2.59, 2/2.81, 2/3, 3/3.17, 0

  = 3, 2, 1.89, 0, 0, 0.39, 0.71, 0.67, 0.95, 0

- Discounted cumulative gain (DCG@*n*):

  3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

Normalized DCG – NDCG

- **Doc: 3, 2, 3, 0, 0, 1, 2, 2, 3, 0**
  DCG@*n*: 3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61
- *Normalized* by comparing the DCG at each rank with that of the *perfect ranking*
- Perfect ranking: 3, 3, 3, 2, 2, 2, 1, 0, 0, 0

Ideal DCG@n: 3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10.88

- NDCG@*n* values (divide by ideal):

$$\left(\frac{3}{3}, \frac{5}{6}, \frac{6.89}{7.89}, \frac{6.89}{8.89}, \frac{6.89}{9.75}, \frac{7.28}{10.52}, \frac{7.99}{10.88}, \frac{8.66}{10.88}, \frac{9.61}{10.88}, \frac{9.61}{10.88}\right)$$

1, 0.83, 0.87, 0.76, 0.71, 0.69, 0.73, 0.8, 0.88, 0.88

Notes on the "Perfect Ranking"

- It is NOT the ideal sequence of THIS SE's results, but the ideal sequence of the results by ALL SEs, if we are comparing multiple SEs.
- SE A: 0 3 2 0 2 1 0 0 0 0 (pages are: a b c d e f g h i j)
- SE B: 3 2 0 1 1 1 0 2 0 0 (pages are: b c a l m f i n k o)
- Then we have:
  a0 b3 c2 d0 e2 f1 g0 h0 i0 j0 k0 l1 m1 n2 o0
- So the ideal sequence of top 10 results should be:
  3 2 2 2 1 1 1 0 0 0

NDCG@*n*

- *Normalized* by comparing the DCG at each rank with that of the *perfect ranking*
  - Makes averaging easier for queries with different number of relevant documents
- NDCG ≤ 1 at any rank position
- Takes grade relevance and position information into consideration.
- One of the most popular used metrics in IR research and industry.

Overview: Evaluation

- **Evaluation methodology**
  - Cranfield-like evaluation
- **Annotation Consistency**
  - Kappa Coefficient
- **Common metrics for relevance**
  - Precision, Recall, MAP, MRR, F-measure
  - DCG and NDCG

Part II. Search Engine Techniques

— （I） Link-based Analysis

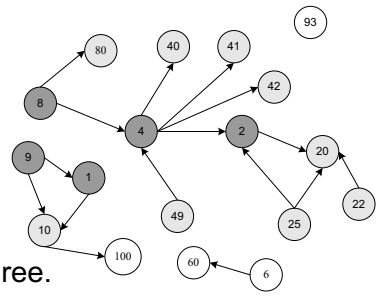
Web IR Techniques: Link-based IR

- Counting node degree
- PageRank
- HITS (authorities and hubs)
- TrustRank
- Spreading activation
- Anchor text

Web IR Techniques: Link-based IR

- Counting node degree
- PageRank
- HITS (authorities and hubs)
- TrustRank
- Spreading activation
- Anchor text

Counting node degree



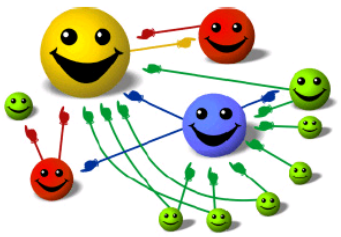
- Index node:
  Whose out-degree is significantly larger than the average out-degree.
- Reference node:
  Whose in-degree is significantly larger than the average in-degree
- Propose measures of centrality
  - Based on node-to-node distances in the link structure graph
  - Rank = in-degree + out-degree
- A "counting" notion of WWW link structure

Web IR Techniques: Link-based IR

- Counting node degree
- PageRank
- HITS (authorities and hubs)
- TrustRank
- Spreading activation
- Anchor text

PageRank -- background

- Measure the importance of the Web pages
- Recommendation assumption:
  - If A has a link that points to B, then the author of A recommend page B.
  - The more being recommended, the better.
  - Good recommender gives better recommendations.



PageRank – basic idea

- Sergey Brin and Larry Page 1998 (Google)
  - 15925 Citations on Google Scholar
- A model of user behavior
- A "Random surfer" – random walk model:
  - Randomly given a webpage
  - Keep clicking
  - Never hitting "back"
  - Eventually get starts on another random page
- Simulate the user's navigation procedure with Markov chain
  - t→∞, the probability of each page that the user stays: PR

The anatomy of a large-scale hypertextual Web search engine, S. Brin, L. Page, www7, 1998
L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998

PageRank -- computation

- Computes a document's score based on the scores of documents that link to it.

$$PR(A) = \frac{d}{N} + (1-d)\left(\frac{PR(T_1)}{L(T_1)} + \cdots\cdots + \frac{PR(T_n)}{L(T_n)}\right)$$

T_i links to A, $L(T_i)$: out-degree of page T_i N: # of pages
d: probability of re-start (i.e. not following links), (0,1) ~ generally 0.15

- A probability distribution over web pages, Sum of all PR(A) = 1
  - A simple iterative algorithm
  - Corresponding to the principal eigenvector of the normalized link matrix of the web

PageRank – computation (cont.)

- Don't forget to dealing with pages with NO out-link

Algorithm:

- Initialize $PR(T_i)$ for each page T_i, $\sum_{T_i} PR(T_i) = 1$
- In each loop:
  1. For each page *A*, $I(A) = \frac{d}{N}$
  2. For each page T_i
     - if outdegree$(T_i) > 0$,
       - *For each A that* $T_i \rightarrow$ A , $I(A) = I(A) + (1-d)\frac{PR(T_i)}{L(T_i)}$
     - else (i.e. outdegree$(T_i) = 0$)
       - *For each A,* $I(A) = I(A) + (1-d)\frac{PR(T_i)}{N}$
  3. For each page *A*, $PR(A) = I(A)$

You needn't normalize PR scores by the end of each loop. It's already been guaranteed.

Improvements on PageRank

- On speed-up of the computation
- On refinement and enrichment of the model
  - PageRank – nothing about content
    - Topic-sensitive PageRank, Query-dependent PageRank
  - Block-based PageRank, ……
- Others
  - Modifying the personalized vector
  - Introducing inter-domain and intra-domain link weights
    - HostRank, SiteRank, ……
  - TrustRank, anti-TrustRank

Web IR Techniques: Link-based IR

- Counting node degree
- PageRank
- HITS
- TrustRank
- Spreading activation
- Anchor text

HITS: Hypertext-Induced Topic Search

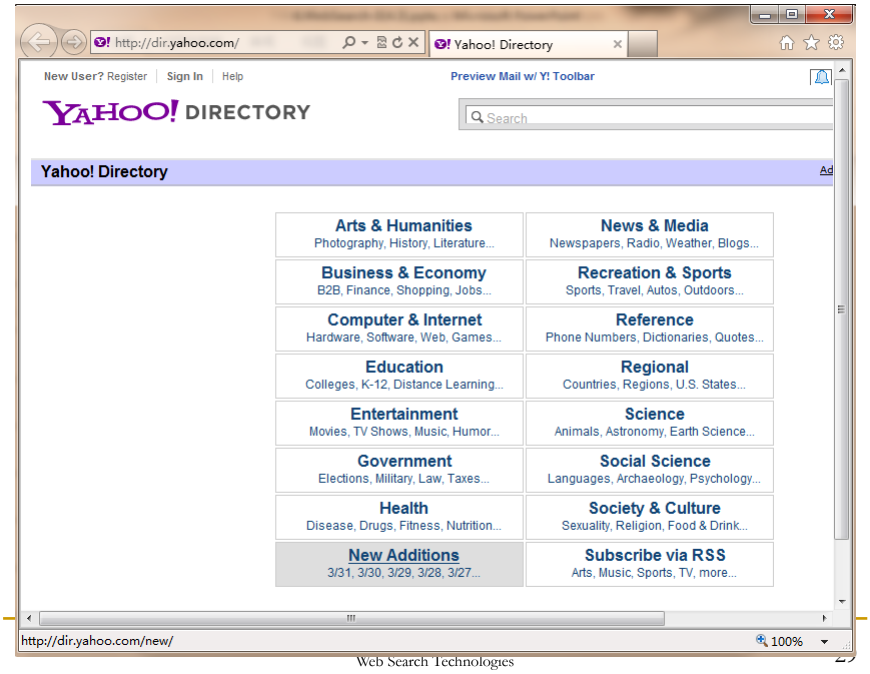
- John Kleinberg 1998 (Cornell)
  - *"Authoritative Sources in a Hyperlinked Environment"*
  - 10783 citations on Google Scholar
- ONR (*Office of Naval Research*) Young Investigator Award
- A MacArthur Foundation Fellowship
- A Packard Foundation Fellowship
- A Sloan Foundation Fellowship
- Member of the National Academy of Engineering
- And the American Academy of Arts and Sciences

HITS: Hypertext-Induced Topic Search

- 3 types of queries (examples from the original paper)
  - Specific queries
    - "Does Netscape support the JDK 1.1 code-signing API?" *far too large #relevant_docs, need authoritative or definitive one*
  - Broad-topic queries
    - "Find information about the Java programming language."
  - Similar-page queries
    - "Find pages 'similar' to java.sun.com."

HITS: Hypertext-Induced Topic Search

- Authority page
  - Good sources of **content**
  - Large **in-degree**, e.g.
  - www.amazon.com, www.wikipedia.com , zhidao.baidu.com
- Hub pages
  - good sources of **links**
  - Pull together authorities on a given topic
  - Throw out unrelated pages of large in-degree
  - E.g. **dir.yahoo.com**



HITS: Hypertext-Induced Topic Search

- Authority page
  - Good sources of **content**
  - Large **in-degree**, e.g.
  - www.amazon.com, www.wikipedia.com , zhidao.baidu.com
- Hub pages
  - good sources of **links**
  - Pull together authorities on a given topic
  - Throw out unrelated pages of large in-degree
  - E.g. **dir.yahoo.com**
- Relationship of authorities and hubs
  - **Mutually reinforcing relationship**
  - A good hub points to many good authorities
  - A good authority is pointed by many good hubs

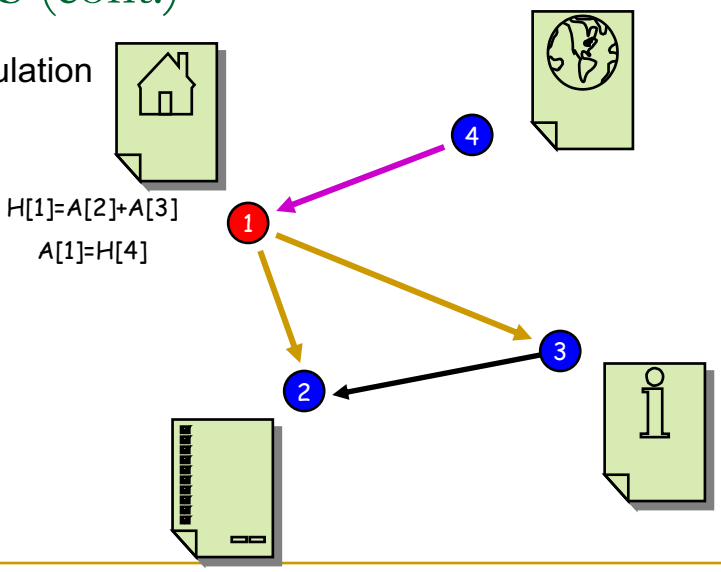
HITS

- A good *Base Set* R_σ is quite important
  - *a)* R_σ is relatively small.
  - *b)* R_σ is rich in relevant pages.
  - *c)* R_σ contains most (or many) of the strongest authorities.
- A root set
  - Collect the *t* highest-ranked pages for the query σ (e.g. t=200)
    - Good for a) and b), not enough for c)
- Expand root set to base set
  - Add all pages that the root set link to
  - Add 50 pages that link to the root set
  - Remove in-site links

Then its size is generally 1000-5000

HITS (cont.)

- Calculation



HITS (cont.)

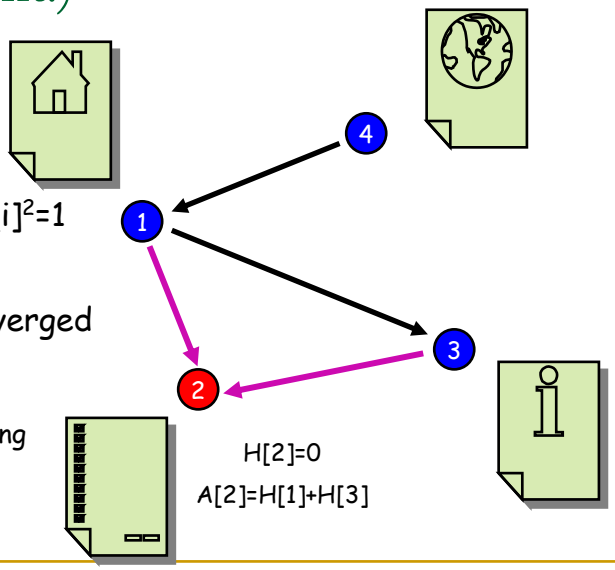
- Calculation

Normalization:

$\sum A[i]^2=1, \quad \sum H[i]^2=1$

Repeat until converged

Generally be used within the searching result documents -- topic specific / query dependent



H[2]=0

A[2]=H[1]+H[3]

Improvements on HITS

- Problems: can not work well in all cases
  - Mutually reinforcing relationships between hosts
  - Automatically generated links
  - Non-relevant nodes (topic drifting problem)

    e.g. "mango fruit" → "fruit"
- Some ideas
  - **Hosts problem** solution: A(n) → A(n)/k H(n) → H(n)/l
  - 2 basic approaches to tackle **topic drift**
    - Elimination non-relevant nodes from the graph
    - Regulating the influence of a node based on its relevance

Web IR Techniques: Link-based IR

- Counting node degree
- PageRank
- HITS (authorities and hubs)
- TrustRank
- Spreading activation
- Anchor text

TrustRank – link-based spam page detection

- Trust Rank
  - Rationale: The approximate isolation of the good set
    - Good pages seldom point to bad ones.
  - Basic Idea: in-link based trust propagation
  - Select seed set
    - With high Invert PageRank (outlink-based PR)
      - To spread the trust score quickly
    - (And) high PageRank
      - To guarantee the quality of seed pages
    - Initial score t* = d
      - *d: normalize static score distribution, seed:1 others: 0*

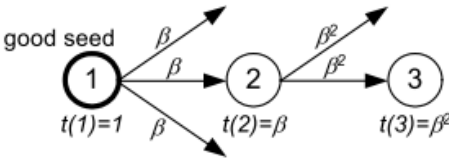
Trust Attenuation

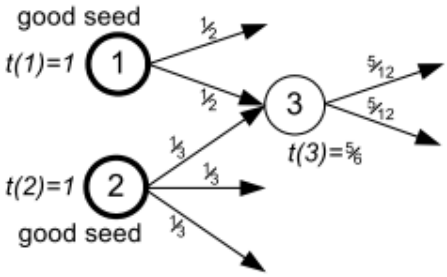
Type I:

- Trust dampening

Type II:

- Trust splitting





$0 < \beta < 1$

TrustRank Algorithm

function TrustRank
input

	$\mathbf{T}$	transition matrix
	N	number of pages
	L	limit of oracle invocations
	α_B	decay factor for biased PageRank
	M_B	number of biased PageRank iterations

output

	$\mathbf{t}^*$	TrustRank scores

α_B: generally set to ~0.85,

```
begin
        // evaluate seed-desirability of pages
(1)     s = SelectSeed(...)
        // generate corresponding ordering
(2)     σ = Rank({1,...,N}, s)
        // select good seeds
(3)     d = 0_N
        for i = 1 to L do
                if O(σ(i)) == 1 then
                        d(σ(i)) = 1
        // normalize static score distribution vector
(4)     d = d/|d|
        // compute TrustRank scores
(5)     t* = d
        for i = 1 to M_B do
                t* = α_B · T · t* + (1 − α_B) · d
        return t*
end
```

Link Based Detection

- Trust Rank Result
  - Precision and recall, PageRank vs. TrustRank

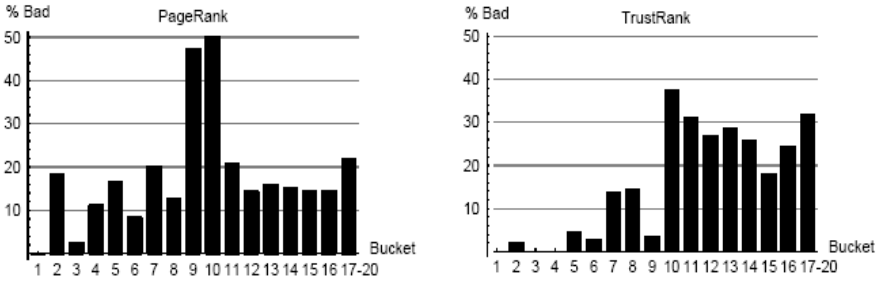


Figure 10: Bad sites in PageRank and TrustRank buckets.

Rank pages with scores. Rank 1: highest score.
Bucket setting: The sum of scores in each bucket is equal.

Z. Gyongyi, et al. Combating web spam with trustrank. In *VLDB '04, 576–587, 2004.*

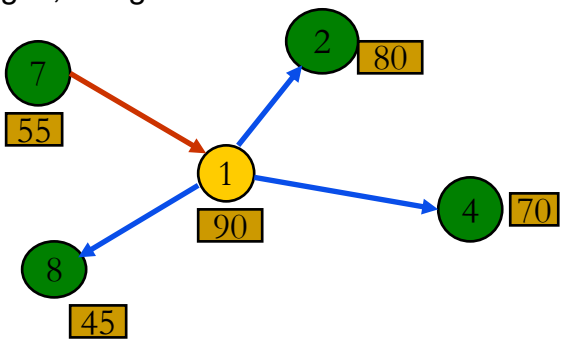
Web IR Techniques: Link-based IR

- Counting node degree
- PageRank
- HITS (authorities and hubs)
- TrustRank
- Spreading activation
- Anchor text

Spreading Activation

- For result re-ranking:

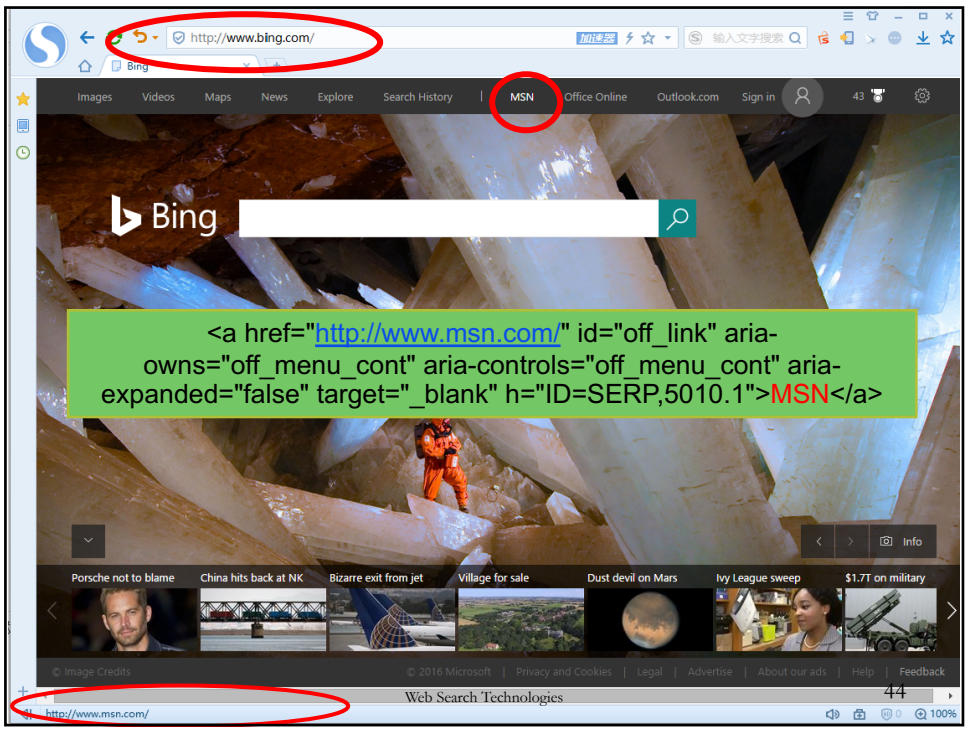
  When the initial relevance scores have been generated for result pages, using SA to re-rank the results.



$$RSV(D_1) = 90 + \lambda_1 * (80 + 45 + 70) + \lambda_2 * 55$$

Web IR Techniques: Link-based IR

- Counting node degree
- PageRank
- HITS (authorities and hubs)
- TrustRank
- Spreading activation
- Anchor text



On the use of anchor text

- Three ways to use anchor text – I
  - Use anchor text as the complementary document information
    - $Document_i' = Document_i \cup Anchor_i$

Where $Anchor_i = \{anchor\_text_{ij} \mid$ where doc j has link to doc $i\}$

- $Final\_score_i$ = ranking score on $Document_i'$

On the use of anchor text

- Three ways to use anchor text – II
  - Use anchor text search result to re-rank
    - $Final\_score = f(Doc\_search\_score, Anchor\_search\_score)$
    - Selection of $f()$ is quite heuristics, by experiential parameters
    - E.g. Linear combination:
      - $Final\_score = \alpha * Doc\_search\_score + (1- \alpha)\ Anchor\_search\_score$

On the use of anchor text

- Three ways to use anchor text – III
  - Only use anchor text

    $Final\_score = Anchor\_search\_score$
- An observation
  - Only use anchor text is significantly better than use the original webpage on site finding tasks

    -- by Nick Craswell, David Hawking and Stephen Robertson, SIGIR'01

Improvements on the use of anchor text

- Filtering anchor text noise
- Use anchor text for finding web synonyms
- For query expansion
- For abbreviations
- Anchor text weighting with click information (clicked anchor)
- ……

Web IR Techniques: Link-based IR

- Counting node degree
- PageRank
- HITS (authorities and hubs)
- TrustRank
- Spreading activation
- Anchor text

References

- "The Anatomy of a Large-Scale Hypertextual Web Search Engine", S. Brin, L. Page, www7, 1998
- "Authoritative Sources in a Hyperlinked Environment",Jon M. Kleinberg, Proc. of the 9th annual ACM-SIAM symposium on Discrete Algorithms, pp 668-677, 1997
- L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998
- "Experiments in Topic Distillation", S. Chakrabarti, B. Dom, D. Gibson, S.R. Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins, ACM SIGIR workshop on hypertext information retrieval on the web, 1998
- "Improved Algorithms for Topic Distillation in a Hyperlinked Environment", K. Bharat, M. R. Henzinger, SIGIR 1998
- "Effective site finding using link anchor information", N. Craswell, D. Hawking, S. E. Robertson, SIGIR 2001
- Does "Authority Mean Quality? Predicting Expert Quality Ratings of Web Documents", B. Amento, L. Terveen, W. Hill, SIGIR 2000
- Daniel E. Rose, Danny Levinson, Understanding User Goals in Web Search, www2004.

- Yiqun Liu, *Min Zhang,* Liyun Ru, Shaoping Ma. Data Cleansing for Web Information Retrieval using Query Independent Features. Journal of the American Society for Information Science and Technology (JASIST), Volume 58, Issue 12, Pages 1884-1898, 2007
- Ricardo B. Yates, et al, The Intention Behind Web Queries, SPIRE 2006.
- T. Haveliwala. Efficient computation of pageRank. Technical Report 1999-31, 1999.
- T. H. Haveliwala. Topic-sensitive pagerank. In *WWW ' 02,* Honolulu, Hawaii, May 2002.
- M. Richardson and P. Domingos. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In *Advances in Neural Information Processing Systems 14. 2002.*
- A. N. Langville and C. D. Meyer. Deeper inside pagerank. *Internet Mathematics, 1(3):335–400, 2004.*
- Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *VLDB '04, pages 576–587, 2004.*
- F. McSherry. A uniform approach to accelerated pagerank computation. In *WWW '05, pages 575–582,* USA, 2005.

FINAL PROJECT PROPOSAL

- Please add a post on *learn.tsinghua.edu.cn* "*course discussion*" （课程讨论） section, with:
  - The title (including 4 parts):
    - [Name1] [Student ID1] [Name2] [Student ID2] [Proj. Name]
  - The content
    - Short description on what you will do in the project (100~200 words).

Submission deadline: by April 17th 11:59pm (Wed.)

EXTENDED READING

Precision, Recall, MAP, MRR, F

DCG and NDCG

ROC curve and AUC

Accuracy: not good enough for classifiers with rank

- Two classifiers

Rank list 1	+	+	+	+	–	+	–	–	–	–
Rank list 2	–	+	+	+	+	–	–	–	–	+

+: relevant, –: non-relevant

Accuracy of Classifier1: 4/5

Accuracy of Classifier2: 4/5

But in application and intuition, Classifier 1 is better!

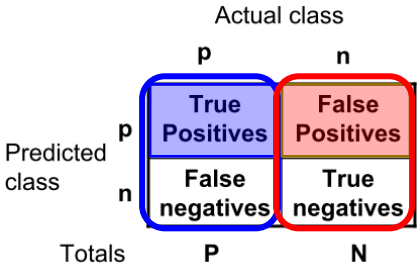
Accuracy vs ranking

- Accuracy-based: making two assumptions:
  - Balanced class distribution, and
  - Equal costs for misclassification
- Ranking: step aside these assumptions
  - Problem: Training examples are labeled, not ranked
- How to evaluate ranking?

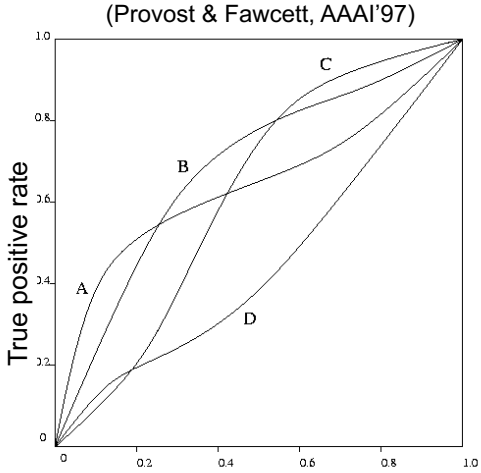
ROC curves

- *ROC* = *Receiver Operating Characteristic*
- Started in electronic signal detection theory (1940s - 1950s)
- Has become very popular in biomedical applications, particularly radiology and imaging
- Also used in machine learning applications to assess classifiers
- Can be used to compare tests/procedures

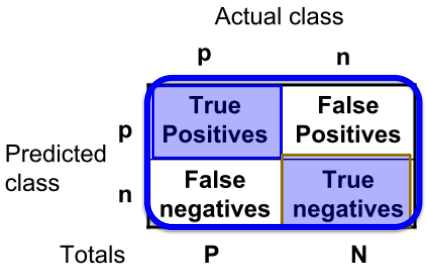
ROC curves



(Provost & Fawcett, AAAI'97)



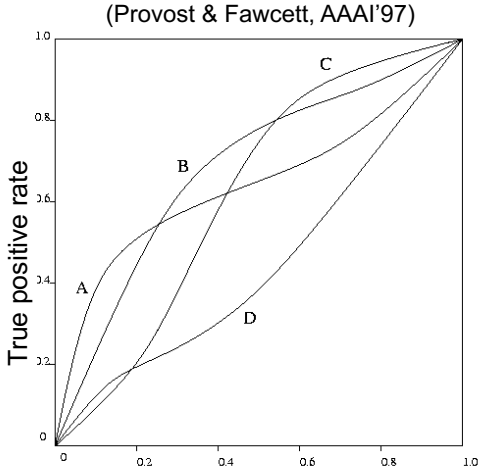
ROC curves and Accuracy



TP rate = TP / P recall (hit rate)

FP rate = FP / N (false alarm rate)

(Provost & Fawcett, AAAI'97)



ROC curves, Accuracy and Precision

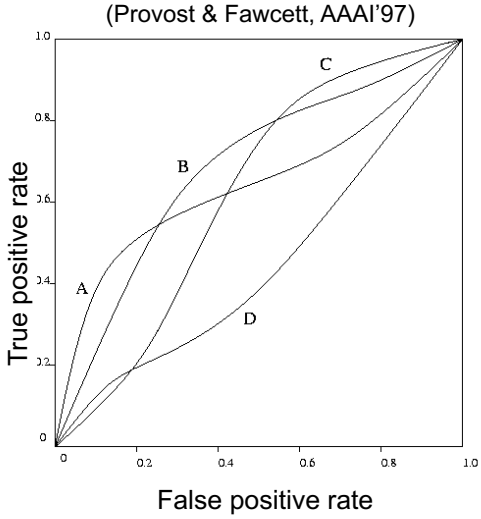
(Provost & Fawcett, AAAI'97)

		Actual class: p	Actual class: n
Predicted class	p	True Positives	False Positives
	n	False negatives	True negatives
Totals		P	N

TP rate = TP / P recall (hit rate)

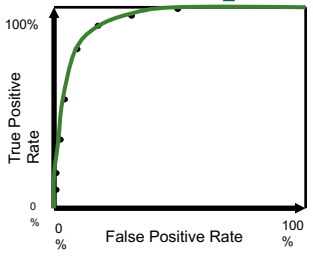
FP rate = FP / N (false alarm rate)

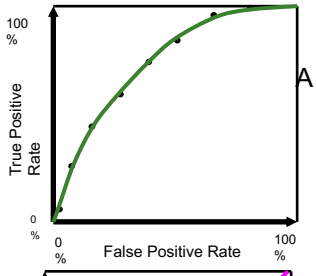
Accuracy = (TP + TN) / (P + N)



ROC curve comparison

A good test

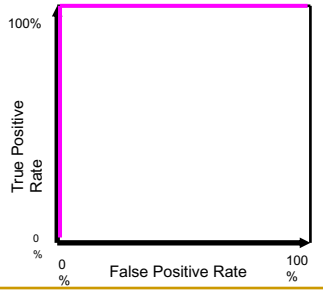


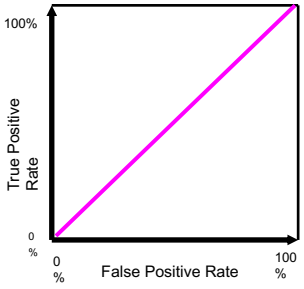


A poor test

Best Test

The distributions don't overlap at all





Worst test

The distributions overlap completely

ROC curve example

- T: threshold

Actual class	score
1	0.98
0	0.80
1	0.67
1	0.65
0	0.54
0	0.32

T=0.5

Actual class	score	class
1	0.98	1
0	0.80	1
1	0.67	1
1	0.65	1
0	0.54	1
0	0.32	0

predicted \ actual	0	1
0	1	0
1	2	3

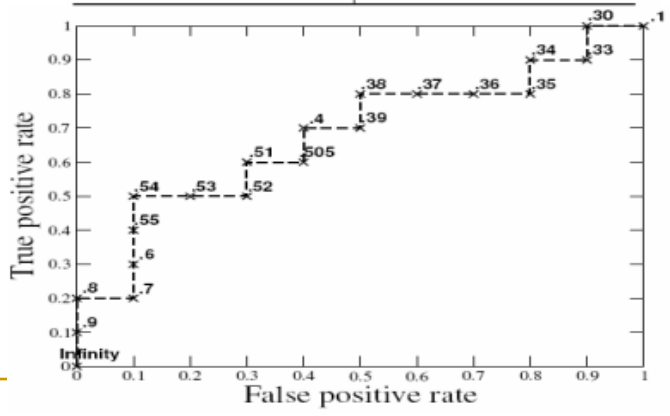
T=0.7

Actual class	score	class
1	0.98	1
0	0.80	1
1	0.67	0
1	0.65	0
0	0.54	0
0	0.32	0

predicted \ actual	0	1
0	2	2
1	1	1

For a model f, each threshold value T gives a point in ROC space

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

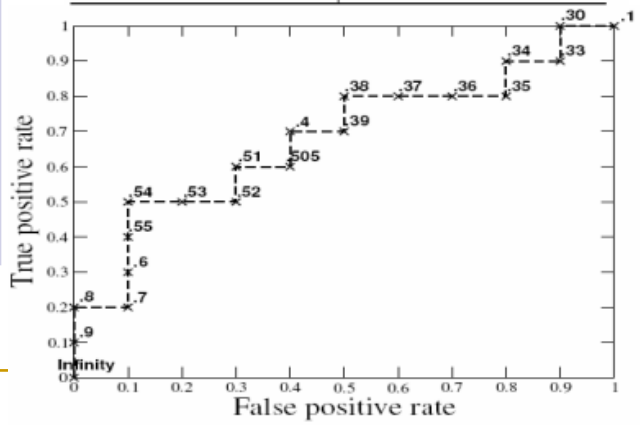


1. T = min-score
2. TP=0, FP=0
3. For each obs. i
   if score-i > T
      increment TP
   else
      increment FP
4. Add point (FP/N, TP/P)
   to ROC graph

-Increment T from min-score to max-score (by smallest difference between two scores)

- Repeat 1. to 4. above to add a point to the graph.

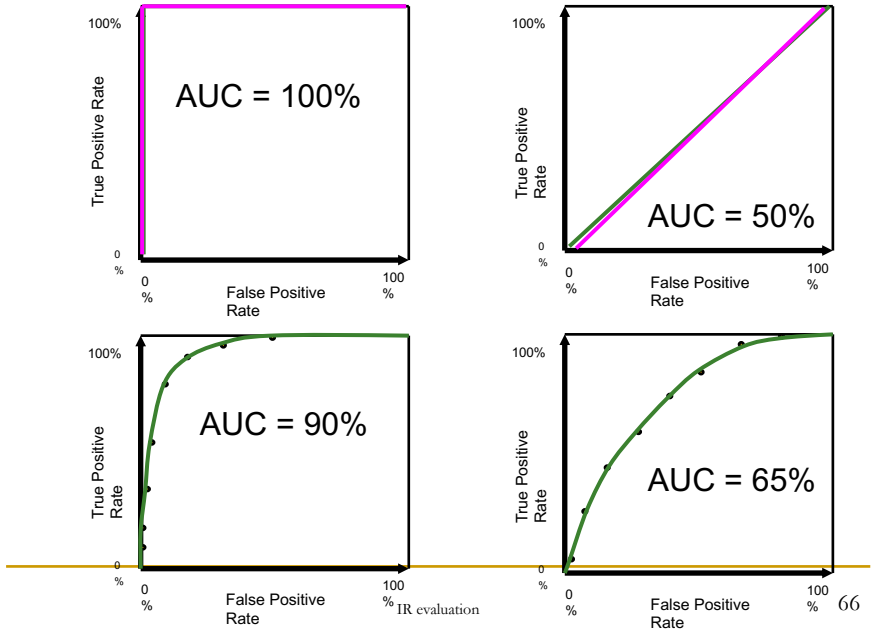
Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1



AUC: Area under ROC curve

- *Overall measure* of test performance
- *Comparisons* between two tests based on differences between (estimated) AUC
  - One ROC curve dominates the other, its AUC must be larger.
  - Intuitively, the larger the AUC of a ROC, the better.
- For continuous data, AUC equivalent to *Mann-Whitney U-statistic* (nonparametric test of difference in location between two populations)
- It represents the possibility to put positive examples before negative ones.

AUC for ROC curves



How to calculate AUC

- Rank test examples in an increasing order $\{g_1, \ldots, g_{n1}, f_1, \ldots, f_{n0}\}$. g_i: positive; f_i: negative
  - n_1: total number of positive examples
  - n_0: total number of negatives examples
  - Let r_i be the rank of ith negative example
  - $S_0 = \sum r_i$
- AUC: $\hat{A} = \dfrac{S_0 - n_0(n_0+1)/2}{n_0 n_1}$

	+ + – + – + – +
i	1 2 3
r_i	3 5 7

- (Hand & Till, 2001, MLJ)

AUC examples

- Two classifiers (ranklist):

Ranklist 1	+	+	+	+	–	+	–	–	–	–
Ranklist 2	–	+	+	+	+	–	–	–	–	+

The AUC of ranklist 1: $\frac{(5+7+8+9+10)-5\times 6/2}{5\times 5}=\frac{24}{25}$

The AUC of ranklist 2: $\frac{(1+6+7+8+9)-5\times 6/2}{5\times 5}=\frac{16}{25}$

Ranklist 1 is better than 2!