Graph 1 & 2.

Social Popularity Analysis

(Link Analysis)

Outline

- Part I
  - Hubs and Authorities (HITS)
  - PageRank
- Part II
  - Personalized PageRank
  - Topic-sensitive PageRank
- Part III. Evaluation of Ranked Lists

Enriched View of IR in the Internet Era

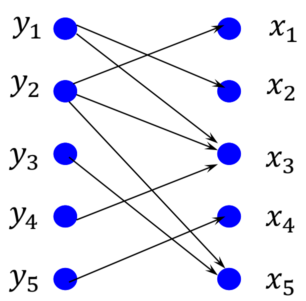
- What is a document anyway?
  - A bag of words?
  - A bag of links?
  - A bag of linked pages?
  - A node in a connected graph?
- Retrieval criteria?
  - **Traditional IR**: Find the most relevant documents for each query
  - **Newer View**: Find the most relevant & authoritive documents for each query (relevance + popularity)

Motivative Examples

- **Retrieval**: If two documents are equally relevant, we want the more popular one to be ranked higher.
- **Web browsing**: Which web sites are more authoritive? Where are the good hubs?
- **Literature overview**: Which are the seminal papers on certain topic?
- **Social networks**: Who are the most important persons in a community?
- All those questions require to analyze the linked structure over a graph.

Bipartite Graph & Adjacency Matrix

Out-links In-links



Each node is a web page;
Each edge is a hyperlink.

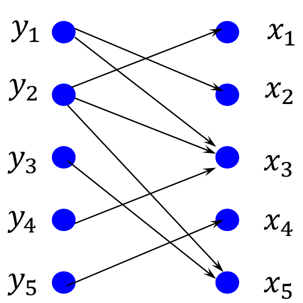
Adjacency Matrix A

	x_1	x_2	x_3	x_4	x_5
y_1	0	1	1	0	0
y_2	1	0	1	0	1
y_3	0	0	0	0	1
y_4	0	0	1	0	0
y_5	0	0	0	1	0

A[i,j] = 1 if there is a link from i to j.

Hubs & Authorities

Out-links **In-links**



Good Hub

- Having many out links (e.g., y_2)
- Pointing to many good authorities (e.g., $y_4 > y_5$)

Good Authority

- Having many in links (e.g., x_3)
- Pointed by many good hubs (e.g., $x_1 > x_2$)

Each node receives two scores (hub & authority scores).

H & A: mutually reinforce each other

Authority score update

$$x_j := \sum_{i=1}^{n} a_{ij}\, y_i = A_{:j}^T y$$

$A_{:j}$ is a column of A *and* $y = (y_1 \quad \cdots \quad y_n)^T$.

Hub score update

$$y_i := \sum_{j=1}^{n} a_{ij}\, x_j = A_{i:} x$$

$A_{i:}$ is a row of A *and* $x = (x_1 \quad \cdots \quad x_n)^T$.

The Compact Notion

vector of authority scores

$$x := A^T y \quad \text{where} \quad y = (y_1 \quad \cdots \quad y_n)^T$$

vector of hub scores

$$y := Ax \quad \text{where} \quad x = (x_1 \quad \cdots \quad x_n)^T$$

Iterative update

$$\begin{cases} x^{(k)} := A^T y^{(k-1)} \\ y^{(k)} := A x^{(k)} \end{cases} \Rightarrow \begin{cases} x^{(k)} := A^T A x^{(k-1)} \\ y^{(k)} := A A^T y^{(k-1)} \end{cases}$$

Updating Rule (Power Iteration)

Letting $B_a = A^T A$ and $B_h = AA^T$, we have:

$$x^{(k)} := B_a x^{(k-1)} = \cdots = B_a{}^{k-1} x^{(1)}$$
$$y^{(k)} := B_h y^{(k-1)} = \cdots = B_h{}^{k} y^{(0)}$$

- We have a chicken-egg problem: Where shall we start?
- It converges when k is sufficiently large. (Where and why?)

Convergence of Power Iteration

- https://en.wikipedia.org/wiki/Power_iteration
  - "If we assume the matrix has an eigenvalue that is strictly greater in magnitude than its other eigenvalues and the starting vector has a nonzero component in the direction of an eigenvector associated with the dominant eigenvalue, then a subsequence converges to the eigenvector associated with the dominant eigenvalue."
- We will revisit the convergency property later (in the lecture on SVD of matrices).

Kleinberg's HITS (Jon Kleinberg, JCAM 1999)

Let *q* be a single-word query.

1. Use a text-based search engine to retrieve top-***t*** pages (***R*** **=** "root set") for the query.
2. Expand ***R*** to R' (up to 50 pages, for example) with the pages that have an in-link to ***R*** or an out-link from ***R.***
3. For set R', compute the authority (A) and hub (H) scores iteratively (usually 10 to 20 iterations would be sufficient)
4. Rank the documents in R' based their authority or hub scores.

Kleinberg's HITS (cont'd)

Iterate(G, K):

Initial settings $z = (1,1,\ldots,1) \in R^n, \quad y^{(0)} = z$

For k = 1 to K

$$x^{(k)} := A^T y^{(k-1)} , \quad y^{(k)} := A x^{(k)}$$

$$\boxed{x^{(k)} := \frac{x^{(k)}}{\|x^{(k)}\|} , \quad y^{(k)} := \frac{y^{(k)}}{\|y^{(k)}\|}}$$

Resulting in $x^{(k)} \propto (\underbrace{A^T A}_{B_a})^{k-1} \underbrace{A^T z}_{x^{(1)}}, \quad y^{(k)} \propto (\underbrace{A A^T}_{B_h})^k z$

Outline

- ✓ Part I
  - ✓ Hubs and Authorities (HITS)
  - ○ PageRank
- Part II
  - Personalized PageRank
  - Topic-sensitive PageRank
- Part III. Evaluation of Ranked Lists

PageRank (S. Brin and L. Page, WWW 1998)

- Probabilistic Transition Matrix M (n by n) is obtained by normalizing each row vector of the adjacency matrix, making its elements sum to 1.

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \Rightarrow M = \begin{pmatrix} 0 & 1/2 & 0 & 1/2 \\ 1/3 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \\ 1/2 & 1/2 & 0 & 0 \end{pmatrix}$$

- Teleportation Matrix E (n by n)

$$E = \frac{1}{n}\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} = \frac{1}{n}\vec{1}\vec{1}^T \quad \text{that is,} \quad \forall i,j: E_{ij} = \frac{1}{n}$$

- Weighted Combination

$$B_{pr} = ((1-\alpha)M + \alpha E)^T \qquad 0 < \alpha < 1 \text{ (typically set } \alpha \text{ to 0.1 ~ 0.2)}$$

Iterative Update

- Initial vector (a probabilistic distribution)

$$r^{(0)} = (\mathrm{r}_1, \mathrm{r}_2, \dots, r_n) \qquad r_i \geq 0,\ \ \sum_{i=1}^{n} r_i = 1$$

- Iterative update

$$r^{(k)} := B_{pr} r^{(k-1)} := {B_{pr}}^{k} r^{(0)}$$

- It converges to a stationary vector (the principal eigenvector of B_{pr}) which does not necessarily depend on the initial vector.

The Random Walk Metaphor

$$r^{(k)} := \underbrace{\left((1-\alpha)M^T + \alpha E^T\right)}_{B} r^{(k-1)}$$

- Start from a randomly picked web page (according to initial $r^{(0)}$).
- Follow the probabilistic transitions in B (either M or E by flipping a coin with the head/tail probabilities of α and 1 - α).
- Repeat the above until r is stabilized (as the 1st eigenvector of B).
- The resulted vector consists of the PageRank scores of nodes, i.e., the expected probability for each page being visited.
- $r^{(k)}$ (for k= 0, 1, 2, …) is always a probabilistic distribution, i.e., the elements are always non-negative and summing to 1.

HITS vs. PageRank (PR)

- HITS $x^{(k)} \propto B_a x^{(k-1)} \propto \underbrace{(A^T A)}_{B_a}^{k-1} x^{(1)}$

  $y^{(k)} \propto B_h y^{(k-1)} \propto \underbrace{(AA^T)}_{B_h}^{k} y^{(0)}$

- PageRank $r^{(k)} = B_{pr} r^{(k-1)} = \underbrace{((1-\alpha)M^T + \alpha E^T)}_{B_{pr}}^{k} r^{(0)}$

$$x \in R^n, \quad y \in R^n, \quad r \in [0,1]^n, \quad \sum_{i=1}^{n} r_i = 1, \quad 0 < \alpha < 1$$

Notice that B_{pr} is not sparse, thus the update might be costly.

@Yiming Yang, Lecture on Link Analysis 17

17

17

Efficient Computation

Originally: $r^{(k)} := \underbrace{((1-\alpha)M^T + \alpha E^T)}_{B} r^{(k-1)}$

Equivalently: $r^{(k)} := (1-\alpha)M^T r^{(k-1)} + \alpha E^T r^{(k-1)}$

Simplified: $E^{\mathrm{T}} r^{(k-1)} = \frac{1}{n} \vec{1}\vec{1}^T r^{(k-1)} = \left(\frac{1}{n}\vec{1}\right) \underbrace{\vec{1}^T r^{(k-1)}}_{=1} = \frac{1}{n}\vec{1}$

$\boxed{r^{(k)} := (1-\alpha)M^T r^{(k-1)} + \alpha p_0,}$ $p_0 \triangleq \left(\frac{1}{n} \quad \cdots \quad \frac{1}{n}\right)^T$

Computationally efficient by leveraging the sparsity of matrix M.

@Yiming Yang, Lecture on Link Analysis 18

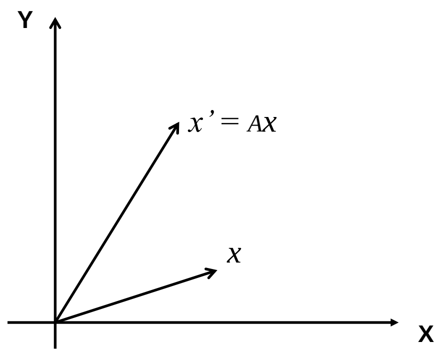
18

18

Propertiy of the Stationary r

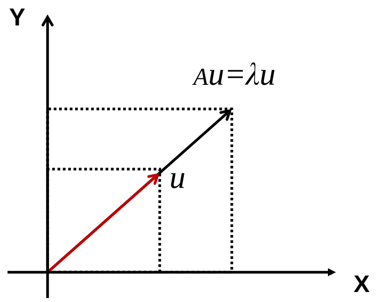
- At the stationary point $B_{pr}r = r$ (as it is converged)
  - Obviously, $\lambda = 1$ is an **eigenvalue** and r is an **eigenvector** of B_{pr}.
  - In fact, a necessary condition for PageRank to converge is that $\lambda = 1$ is strictly larger than any other eigenvalues of B_{pr} in absolution value.

Matrix-Vector Multiplication as a Linear Transformation



Eigenvalue & Eigenvector

- For the eigenvectors if A, the linear transformation can only change their scales but not their directions.



Markov Matrix B_{pr}

- Definition
  - A matrix with nonnegative elements, where each column (or row) summing to 1.
- Both M and E are Markov matrices. Why?
- PageRank matrix is also a Markovian. Why?
  - $B_{pr} = ((1-\alpha)M + \alpha E)^T$

Markov Chains

- Def. A matrix is said to be *strictly positive* (denoted as $B > 0$) if all the elements are positive.
- Def. A **Markov chain** (B^k) is said to be *irreducible* if it is possible to reach every state from any state, i.e.

$$P\left(S^{(k)} = j | S^{(0)} = i\right) > 0, \forall (i,j)$$

- Def. A Markov chain (B^k) is said to be *aperiodic* if for any state i there exist k such that for all k' > k,

$$P\left(S^{(k')} = i | S^{(0)} = i\right) > 0, \forall i$$

- Def. A Markov chain is said to be *regular* if $\exists k \ s.t. \ B^k > 0$

More about Markov Chains

- If B defines a regular Markov chain with finite states, then

$$\lim_{k\to\infty} B^k p = r$$

$$\begin{cases} p \text{ is an arbitrary probability column vector (whose elt's sum up to 1);} \\ r \text{ is a unique stationary distribution (column vector) s.t. } Br = r. \end{cases}$$

- According to the ***Perron-Frobenius theorem,*** any positive square matrix has a unique largest eigenvalue, s.t.

$$\lambda_1 > 0 \quad and \quad \lambda_1 > |\lambda_2|$$

- Any positive Markov matrix has a unique largest eigenvalue of 1 (a special case the ***Perron-Frobenius theorem***), s.t.

$$1 = \lambda_1 > |\lambda_2|$$

Strictly Diagonally Dominant Matrix

- Define $Q \equiv I - (1-\alpha)M$ where M is row-wise stochastic.
- **Proposition**. Matrix Q is *strictly diagonally dominant*, i.e.,

$$|Q_{ii}| > \sum_{j \neq i} |Q_{ij}| \text{ for all } i$$

(You may try to prove it if you wish.)

- **Levy_Desplanques Theorem**. A strictly diagonally dominant matrix is non-singular (i.e., always invertible).
- This can be used to show why the stationary *r* in PageRank is unique.

Closed-form solution for r

- Updating Rule

$$r^{(k)} := (1-\alpha)M^T r^{(k-1)} + \alpha p_0 \quad \text{where } p_0 \equiv \frac{1}{n}1.$$

- At the stationary point where $r^{(k)} = r^{(k-1)}$, we have

$$r = (1-\alpha)M^T r + \alpha p_0$$

$$r - (1-\alpha)M^T r = \alpha p_0$$

$$\underbrace{(I - (1-\alpha)M^T)}_{Q^T} \mathrm{r} = \alpha p_0$$

$$r = (Q^T)^{-1}\alpha p_0 = (I - (1-\alpha)M^T)^{-1}\, \alpha p_0$$

Note: Q is invertible implies that Q^T is also invertible.

Two ways of computing r

- Solving r using the inverse of matrix Q^T

$$r = \alpha\underbrace{\left(I-(1-\alpha)M^T\right)}_{Q^T}{}^{-1}p_0 \qquad \text{where } p_0 \equiv \frac{1}{n}1$$

- Solving r using Power Iteration (until convergence):

$$\begin{aligned} r^{(k)} &:= Br^{(k-1)} \\ &:= (1-\alpha)M^T r^{(k-1)} + \alpha\ p_0 \end{aligned}$$

The latter is computationally more efficient.

PageRank for IR at Google

- Combining two types of scores for each document

$$score(d,q) = f(IRscore(d,q), PageRank(d))$$

-- IRscore(d, q) is the dotproduct of their vectors
-- the function f is not described in the paper

- Rich representation of document (page)

-- title, anchor text or “complete” text as options
-- position, font, capitalization, etc., are indexed for each term
-- word TF, anchor TF, url TF jointly used

Make ranking sensitive to query

- HITS
  - By sampling a subset of web pages nearby each query
- Google

$$score(d,q) = f(IRscore(d,q), PageRank(d))$$

- Other way to make PageRank sensitive to a query?

Outline

✓ Part I
  ✓ Hubs and Authorities (HITS)
  ✓ PageRank
- Part II
  - Personalized PageRank
  - Topic-sensitive PageRank
- Part III. Evaluation of Ranked Lists

How to inject personal preferences to PageRank?

- A user may have personal interests on some links or topics over others
  - E.g., on recent articles on COVID, election, financial markets, school shooting events, …
- How can we modify the PageRank method to reflect such preferences?

How to inject personal preference in PageRank?

- Standard PageRank formulation

$$r^{(k+1)} = B_{pr} r^{(k)} = \left((1-\alpha)M + \alpha E\right)^T r^{(k)}$$

- Shall we change (personalize) the initial vector $r^{(0)}$? Or shall we change (personalize) M or E, instead?
- Let's try ChatGPT ☺ Judge its answers by yourself!
  - Q1. How to inject personal preference into pagerank?
  - Q2. Should I personalize the initial vector $r^{(0)}$?

Personalized PageRank (PPR)

(Haveliwala et al., 2003, Stanford TR)

$$r^{(k+1)} = B_u\, r^{(k)} = \left((1-\alpha)M^T + \alpha E_u{}^T\right) r^{(k)}$$

$$E_u = \vec{1}p_u{}^T = \begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix} \quad \underbrace{(p_{u1},\ p_{u2},\ \cdots, p_{un})}_{p_u} = \begin{pmatrix}p_{u1}, & p_{u2}, & \cdots, p_{un}\\ p_{u1}, & p_{u2}, & \cdots, p_{un}\\ & \vdots & \\ p_{u1}, & p_{u2}, & \cdots, p_{un}\end{pmatrix}$$

$$p_{ui} \in [0,1]\,, \quad \sum_{i=1}^{n} p_{ui} = 1$$

- Personalization vector p_u defines the probabilistic distribution of the web sites the user prefers.

Personalized PageRank (PPR) (cont'd)

- Iterative updating as

$$\begin{aligned} r_u{}^{(k)} &:= B_u\ \ r_u{}^{(k-1)} \\ &= ((1-\alpha)M + \alpha E_u)^T r_u{}^{(k-1)} \\ &= (1-\alpha)M^T r_u{}^{(k-1)} + \alpha\, \underbrace{E_u{}^T r_u{}^{(k-1)}}_{p_u \underbrace{\vec{1}^T r_u{}^{(k-1)}}_{=1}} \end{aligned}$$

$$= (1-\alpha)M^T r_u{}^{(k-1)} + \alpha\, p_u$$

- But can we remove the uniform teleportation part in standard PageRank?

Is the ergodic assumption violated?

- Ergodic Markov chain (Intro. IR, p427)
  - Irreducibility: There is a sequence of transitions with nonzero probability from any state to all other states.
  - Aperiodicity: …
- As the condition for convergence to the steady-state probabilities
- Is this condition violated in our formula?

Remedy

$$B_u^{\ T} = \alpha M + \beta E_u + \gamma E$$

$$where \quad E_u = \vec{1} p_u^{\ T}, \ E = \frac{1}{n} \vec{1}\vec{1}^T,$$

$$\alpha, \beta, \gamma \in (0,1) \quad and \quad \alpha + \beta + \gamma = 1$$

Topic-sensitive PageRank (TSPR)

- For each topic (t), define the topic-specific matrix as

  $${B_t}^T = \alpha M + \beta E_t + \gamma E$$

- Matrix $E_t = \vec{1}p_t^T$ *and* $p_t \in [0,1]^n$ is defined as follows.
  - If a note does not have any output link to an on-topic page, set all the element of p_t with the value of $\frac{1}{n}$;
  - Otherwise, each on-topic page has a weight of $\frac{1}{n_t-1}$ where n_t is the total number of on-topic pages; each not-on-topic page has a weight of zero.

Topic Sensitive PageRank (TSPR)

For each topic, compute the topic-specific pagerank vector as

$$r_t^{(k)} \coloneqq B_t r_t^{(k-1)} = (\alpha M + \beta E_t + \gamma E)^T r_t^{(k-1)}$$

$$= \alpha\ M^T r_t^{(k-1)} + \beta p_t + \gamma\ p_0$$

where $E = \frac{1}{n}\vec{1}\vec{1}^T$, $E_t = \vec{1}{\vec{p}_t}^T$, $p_0 = \frac{1}{n}\vec{1}$, p_t is defined in previous slide, and α+$\beta + \gamma = 1$.

Merging topic-specific ranked lists for each query

Offline computation of the TSPR vectors for t = 1, …, T as:

$$r_t^{(k)} := B_t r_t^{(k-1)} = \alpha\ M^T r_t^{(k-1)} + \beta\ p_t + \gamma\ p_0$$

Online computation given a query (q):

The weighted sum of the TSPR vectors is computed as:

$$r_q{}^{(TSPR)} = \sum_{t=1}^{T} \Pr(t|q) \times r_t$$

Online, query based

Offline, link based

@Yiming Yang, Lecture on Link Analysis

39

39

39

How do we estimate P(t|q)?

- Method 1: NB (binary NB)

$$\Pr(t|q) = \frac{\Pr(t)\Pr(q|t)}{\Pr(q)} = \frac{\Pr(t)\Pr(q|t)}{\sum_{t' \in T}\Pr(t')\Pr(q|t')}$$

$$\propto \Pr(t)\Pr(q|t) = \Pr(t)\ \prod_{w \in q} P(w_j|t)^{TF(w,q)}$$

- Method 2: kNN

$$P_r(t|q) = \frac{\sum_{x_i \in kNN(q)} \delta(y_i, t)}{k}, \quad \delta(y_i, t) = \begin{cases} 1 & \text{if } y_i = \mathrm{t} \\ 0 & \text{otherwise} \end{cases}$$

- Method 3: SoftMax Logistic Regression

$$\Pr(t = k|q) = \frac{\exp(w_k^T q)}{\sum_{k'=1}^{K} \exp(w_{k'}^T q)} \qquad k = 1, 2, \dots, K.$$

@Yiming Yang, Lecture on Link Analysis

40

40

Outline

- ✓ Part I
  - ✓ Hubs and Authorities (HITS)
  - ✓ PageRank
- ✓ Part II
  - ✓ Personalized PageRank
  - ✓ Topic-sensitive PageRank
- Part III. Evaluation of Ranked Lists

Metrics for Evaluating Ranked Lists

- P@n: the proportion of relevant documents among the top n of the ranked list per query, averaged over queries.
- Mean Reciprocal Rank (MRR): RR is the inverse of the rank of the 1^{st} relevant doc in the ranked list for each query; MRR is the average of the RR scores over queries.
- Mean Average Precision (MAP): AP is the average of the precision scores at all relevant doc's in each ranked list; MAP is the average of the AP scores over all queries.
- Normalized Discounted Cumulated Gain (NDCG): allowing multi-scale relevance judgments (omit)
- Precision-Recall Curves, ROC curves, AUC of ROC (omit)

A Toy Example of Ranked List

- Query: Ski areas in Pennsylvania
- Ranked List (red for relevant; gray for irrelevant)
  1. GoSki Pennsylvania, USA - Pennsylvania ski areas, snow ...
  2. Pennsylvania Ski Areas on SkiOdyssey Resort Guide
  3. Press Releases
  4. Ski Areas in the Pocono Mountains and Eastern Pennsylvania
  5. Ski Areas in the United States
  6. Ski areas wrap up season
  7. Ski Areas For Downhill, Cross-country Skiing, other Winter ...
  8. HI-AYH Hostels Near Ski Areas

43

Precision @n

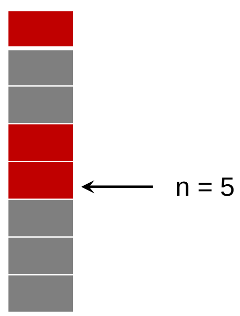
- Precision (P)

$$P = \frac{\text{Number of retrieved and relevant items}}{\text{Number of retrieved items}}$$

- Example

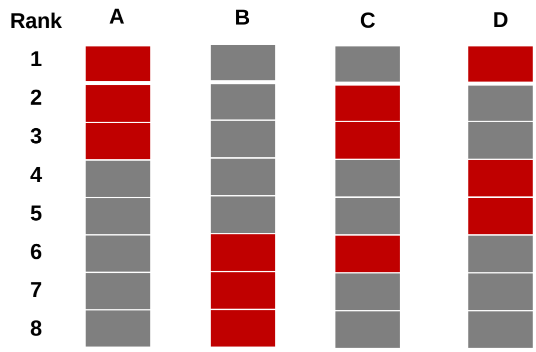
$$P@1 = 100\%$$
$$P@2 = 50\%$$
$$P@5 = 60\%$$
$$P@8 = 37.5\%$$

Ranked List



44

Consider a toy dataset of 8 documents in total: 3 relevant ones (red), 5 irrelevant ones (gray)



Lists A, B, C and D: which is better?

Rank sum of the red boxes: 6 (A), 21 (B), 11 (C), 10 (D)

Rank Sum Statistic

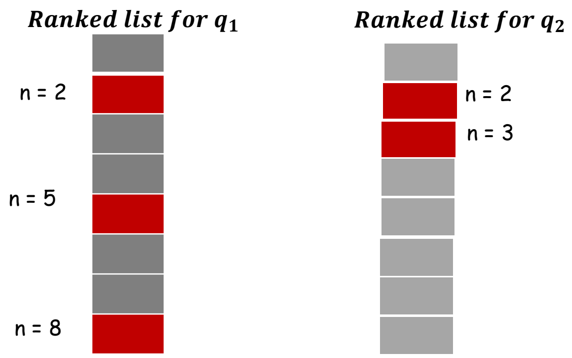
Properties:

- Sufficient for comparing systems on each query (smaller is better)
- Not comparable across queries if the number of relevant documents is different for each query.

Desirable Properties:

- A normalized score per query as between 1 and 0 (higher is better)
- Averaging the per-query scores for each system.

Mean Average Precision (MAP)
(averaging the AP scores over queries)



$AP = (50\% + 40\% + 37.5\%)/3 = 42.5\%$ $\quad$ $AP = (50\% + 66.7\%)/2 = 58.35\%$

$MAP = (42.5\% + 58.35\%)/2 = 50.425\%$

Mean Average Precision (MAP)

- Most common in IR evaluations
- Mimic the rank-sum metric by focusing on the locations of relevant documents only in a ranked list.
- Should be evaluated over the *complete* ranked list (in theory) per query
- Giving more weights to higher-ranking rel. doc's

Average Precision (AP)

Giving more weights to red ones in higher positions

	1st doc	2nd doc	3nd doc
n = 2	1/2		
n = 5	1/5	1/5	
n = 8	1/8	1/8	1/8

$$\text{AP} = (1/2 + 2/5 + 3/8)/3 = 42.5\%$$

Summary of Link Analysis

- Popularity can be defined recursively.
- Popularity can be personalized and made topic-specific.
- We have focused on hard links, but the methods can be applied to soft links as well (e.g., citation graphs, similarity-based graphs, social networks)
- What you should know: the formulation of the matrices, how to evaluate ranked lists, and why HITS/PageRank scores converge (more explanation in the SVD lecture).

References

- Kleinberg. “Authoritative sources in a hyperlinked environment.” ACM-SIAM pp. 668-677. 1998. Extended version in J. ACM 1999.
- S. Brin and L. Page. “The anatomy of a large-scale hypertextual Web search engine”. *Computer Networks*, vol. 30 (1998), pp. 107-117
- Haveliwala, Taher H. "Topic-sensitive pagerank." *Proceedings of the 11th international conference on World Wide Web*. ACM, 2002.