Outline

- Part I
  - Hubs and Authorities (HITS)
  - PageRank

- Part II
  - Personalized PageRank
  - Topic-sensitive PageRank

- Part III. Evaluation of Ranked Lists
Evolution of Retrieval Models

- Traditional IR based on a bag-of-words representation
- Boolean models: TF-IDF, Rocchio, BM25, regression
- Language models
- Neural networks
- Latent Semantic Indexing

IR methods enriched by link analysis:
- HITS
- PageRank
- Topic sensitive PageRank
- Personalized PageRank
- TrustRank
- BrowseRank
- RankSVM
- RankNet
- LambdaRank
- SVM-MAP
- ListNet
- LETOR

Traditional IR methods

Red: Covered in this course

An Enriched View of IR

What is a document anyway?
- A bag of words?
- A bag of links?
- A bag of linked pages?
- A node in a connected graph?

How can we leverage links in IR?
- Find the documents well connected to the (potentially) relevant documents given a query.
- Rank documents based on both relevance estimate (content based) and popularity (link based).
Motivation & Examples

- **Retrieval**: If the retrieved documents are equally relevant, we may want the popular ones ("authorities") to be ranked higher.
- **Web browsing**: Which web sites are "good hubs" (with many links to authority pages), and which web pages are the authorities (with many in-links from good hubs)?
- **Literature overview**: Which papers are the seminal ones in a topic? Which authors are most influential in learning to rank with SVMs, for example?
- **Social Networks**: Who are most important ones in a community (not necessarily the boss)?

Bipartite Graph & Adjacency Matrix

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

Adjacency Matrix $A$

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$y_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$y_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$y_4$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$y_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$A[i,j] = 1$ if there is a link from $i$ to $j$. 
**Hubs & Authorities**

A good hub:
- having many out-links (e.g., $y_2$)
- pointing to many good authorities (e.g., $y_4 > y_5$)

A 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 -- the hub score and the authority score.

**H & A: Mutually Reinforcing**

Authority update

\[
x_j := \sum_{i=1}^{n} a_{ij} y_i = A_j^T y \]

\[\begin{align*}
A_j^T & \text{ is a column of } A \\
y &= (y_1 \cdots y_n)^T
\end{align*}\]

Hub score

\[
y_i := \sum_{j=1}^{n} a_{ij} x_j = A_i \cdot x
\]

\[\begin{align*}
A_i & \text{ is a row of } A \\
x &= (x_1 \cdots x_n)^T
\end{align*}\]
The Compact Notion

Authority

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

Hub

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

We have a chicken-egg problem: Where shall we start?

The Updating Rules

\[
\begin{align*}
x^{(k)} &:= A^T y^{(k-1)} \\
y^{(k)} &:= Ax^{(k)}
\end{align*}
\]

It converges when \( k \) is sufficiently large. (Where and why?)

Letting \( B_a = A^T A \) and \( B_h = AA^T \) we have:

\[
\begin{align*}
x^{(k)} &:= B_a x^{(k-1)} = \cdots = B_a^{k-1} x^{(1)} \\
y^{(k)} &:= B_h y^{(k-1)} = \cdots = B_h^{k} y^{(0)}
\end{align*}
\]

Power Iteration
Convergence of 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 need to talk about the eigenvalues of eigenvectors of a squared matrix (later).

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 ($R = \text{"root set"}$) for the query.

2. Expand $R$ (up to 50 pages, for example) using the pages with an in-link to $R$ and an out-link from $R$

3. For the expanded set of $R$, compute the authority ($A$) and hub ($H$) scores iteratively (10 to 20 iterations are usually sufficient)

4. Rank the documents in $R$ based their authority or hub scores.
Kleinberg's HITS (cont’d)

\[\text{Iterate}(G, k):
\]
Initial settings \[z = (1,1,\ldots,1) \in \mathbb{R}^n, \quad y^{(0)} = z\]
For \(k = 1\) to \(K\)
\[x^{(k)} := A^T y^{(k-1)}, \quad y^{(k)} := A x^{(k)}\]
\[x^{(k)} := \frac{x^{(k)}}{\|x^{(k)}\|}, \quad y^{(k)} := \frac{y^{(k)}}{\|y^{(k)}\|}\]
Resulting in \[x^{(k)} \propto (A^T A)^{k-1} \frac{A^T z}{x^{(k)}}, \quad y^{(k)} \propto (A A^T)^{k} z\]

Revision of HITS
(Bharat & Henzinger, SIGIR-98)

- **Base**: Kleinberg’s HITS algorithm
  -- for query \(q\), fetch \(R\) with up to 50 documents
  -- compute the \(H\) & \(A\) scores for each document in \(R\)
- **Med**: Using pruned start set \(R'\)
  \[\bar{\mu} = \frac{1}{|R|} \sum_{d \in R} \tilde{\mu}, \quad t = \text{median}\{\cos(\tilde{d}, \bar{\mu})\}, \quad R' = \{\tilde{d} \in R | \cos(\tilde{d}, \bar{\mu}) > t\}\]
- **Medr**: Combining \(H/A\) scores (Med) with text-similarity
  \[s_H(\tilde{d}, \tilde{q}) = H(\tilde{d} | \tilde{q}) \times \cos(\tilde{d}, \tilde{q}), \quad s_A(\tilde{d}, \tilde{q}) = A(\tilde{d} | \tilde{q}) \times \cos(\tilde{d}, \tilde{q})\]
Revision of HITS (cont’d)

- Search engine: AltaVista (1997)
- Queries: 28 (used by Chakrabarti et al.)
- Relevance judgments: 3 assessors, majority vote
- Metric: average precision at top 5 doc. per query

<table>
<thead>
<tr>
<th></th>
<th>Base</th>
<th>Med</th>
<th>Medr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authority</td>
<td>0.52</td>
<td>0.73</td>
<td>0.72</td>
</tr>
<tr>
<td>Hub</td>
<td>0.60</td>
<td>0.87</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Outline

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

- Denoting by $n_i$ the number of links from node $i$, the probabilistic transition matrix $M$ ($n$ by $n$) is defined as:
  - If $n_i > 0$
    - if there is a link from $i$ to $j$ and $j \neq i$, then $M_{ij} := \frac{1}{n_i}$;
    - if there is no link from $i$ to $j$ or if $j = i$, then $M_{ij} := 0$;
  - If $n_i = 0$ then $M_{ij} := \frac{1}{n}$ for $j = 1$ to $n$.

- Teleportation Matrix $E$ ($n$ by $n$)
  \[
  E = \begin{pmatrix}
  1 & 1 & \cdots & 1 \\
  1 & 1 & \cdots & 1 \\
  \vdots & \vdots & \ddots & \vdots \\
  1 & 1 & \cdots & 1 
  \end{pmatrix} = \frac{1}{n} \mathbf{1} \mathbf{1}^T
  \]

- Weighted Combination
  \[
  B = (1 - \alpha)M + \alpha E^T \quad 0 < \alpha < 1 \quad \text{(typically set \(\alpha\) to 0.1 ~ 0.2)}
  \]

Iterative Update

- Initial PageRank vector:
  - Any nx1 vector whose elements are in [0,1] and sum to 1.

- Iterative update
  \[
  r^{(k)} := Br^{(k-1)} := B^k r^{(0)} \quad \text{where} \quad B = (1 - \alpha)M^T + \alpha E^T
  \]

- It converges to a stationary vector (the principle eigenvector of $B$) which does not depend on the initial vector.
HITS vs. PageRank (PR)

- **HITS**
  \[ x^{(k)} \propto B_a x^{(k-1)} \propto \left( A^T A \right)^{-1} x^{(1)} \]
  \[ y^{(k)} \propto B_h y^{(k-1)} \propto \left( A A^T \right)^k y^{(0)} \]

- **PageRank**
  \[ r^{(k)} = B_{pr} r^{(k-1)} = \left( (1 - \alpha) M^T + \alpha E^T \right)^k r^{(0)} \]

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

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

Efficient Computation

Initially:
\[ r^{(k)} := (1 - \alpha) M^T r^{(k-1)} + \alpha E^T r^{(k-1)} \]

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

Notice:
\[ E^T r^{(k-1)} = \frac{1}{n} \bar{1}^T (1 - \alpha) M^T r^{(k-1)} = \frac{1}{n} \bar{1} \]

\[ r^{(k)} := (1 - \alpha) M^T r^{(k-1)} + \alpha E^T r^{(k-1)} \]
\[ p_0 \equiv \left( \frac{1}{n} \cdots \frac{1}{n} \right)^T \]

Computationally efficient by leveraging the sparsity of matrix \( M \).
The Random Walk Metaphor

- Start from a randomly picked web page (according to any p).
- Next, follow the probabilistic transitions in B (either M or E by flipping a coin with the head/tail probabilities of $\alpha$ and $1-\alpha$)

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

- Repeat the above until $r$ is stabilized, whose elements are the PageRank scores, i.e., the expected probability for each page being visited.

Properties of Matrix $B_{pr}$

- Markov matrix
  - Nonnegative elt's, with each column (or row) adding to 1;
- Is $B_{pr}$ Markovian?
  - Both $M^T$ and $E^T$ are Markovian;
  - $B_{pr} = (1-\alpha) M^T + \alpha E$ is also a Markovian. (Why?)
Properties of Matrix $B_{pr}$

- At the stationary point $B_{pr}r = r$ (if it converges)
  - Obviously, $\lambda_1 = 1$ is an eigenvalue of $B_{pr}$, and $r$ is an eigenvector;
  - Convergence requires that all the eigenvalues $\lambda_k$ satisfy $|\lambda_k| \leq 1$
    (proof omitted).

| $|\lambda_1| < 1$ |

Interesting Properties of Markov Matrices

- **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.
  \[ P(S^{(k)} = j \mid S^{(0)} = i) > 0, \forall (i, j) \]

- **Def.** A Markov chain $(B^k)$ is said to be **aperiodic** if for any state $i$ there exist $k$ such that for all $k' > k$,
  \[ P(S^{(k')} = i \mid S^{(0)} = i) > 0, \forall i \]

- **Def.** A Markov chain is said to be **regular** if $\exists k$, s.t. $B^k > 0$
Interesting Properties (cont'd)

- If $B$ defines a regular Markov chain with finite states, then
  \[
  \lim_{k \to \infty} B^k p = r
  \]
  where $p$ is an arbitrary probability column vector (whose elt's sum up to 1); $r$ is a unique stationary distribution (column vector) s.t. $Br = r$.

- According to the Perron-Frobenius theorem, any positive square matrix $B > 0$ has a unique largest eigenvalue, s.t.
  \[\lambda_1 > |\lambda_2| \quad (\text{and } \lambda_1 > 0)\]

- Any positive Markov matrix $B > 0$ has a unique largest eigenvalue of 1 (a special case the Perron-Frobenius theorem), s.t.
  \[1 = \lambda_1 > |\lambda_2|.
  \]

Interesting Properties (cont'd)

- Define $Q \equiv I - (1 - \alpha)M$ where $M$ is row-wise stochastic.

- **Proposition.** Matrix $Q$ is *strictly diagonally dominant*, i.e.,
  \[|Q_{ii}| > \sum_{j \neq i} |Q_{ij}|\]
  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 theorem can be used to show the stationary $r$ in PageRank is unique.
The Closed-form Solution for \( r \)

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

- **At the stationary point**
  \[
  r = (1 - \alpha)M^T r + \alpha p_0 \quad \text{(because } r^{(k)} = r^{(k-1)})
  \]
  \[
  r - (1 - \alpha)M^T r = \alpha p_0
  \]
  \[
  (I - (1 - \alpha)M^T) r = \alpha p_0
  \]
  \[
  \frac{Q^T}{Q} r = (Q^T)^{-1} \alpha p_0 = (I - (1 - \alpha)M^T)^{-1} \alpha p_0
  \]

  **Note:** \( Q \) is invertible implies that \( Q^T \) is also invertible.

---

Two ways of computing \( r \)

- **Solving the inverse of matrix \( Q \)**
  \[
  r = \alpha \left( I - (1 - \alpha)M^T \right)^{-1} p_0 \quad \text{where } p_0 \equiv \frac{1}{n}
  \]

- **Using the Power Iteration (until convergence):**
  \[
  r^{(k)} := Br^{(k-1)}
  \]
  \[
  := (1 - \alpha)M^T r^{(k-1)} + \alpha p_0
  \]

  The latter is computationally more efficient.
PageRank in IR Application: Google’s Approach (Brin & Page, WWW 1998)

- Combining two types of scores for each document
  \[ \text{score}(d, q) = f(\text{IRscore}(d, q), \text{PageRank}(d)) \]
  -- \text{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

Outline

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

- **HITS**
  \[ x^{(k)} \propto B_a x^{(k-1)} \propto B_a^{(k-1)} x^{(1)}, \quad B_a = A^T A \]
  \[ y^{(k)} \propto B_h y^{(k-1)} \propto B_h^k y^{(0)}, \quad B_h = AA^T \]
- **PageRank**
  \[ r^{(k)} = B_{pr} r^{(k-1)} = B_{pr} r^{(0)} \quad B_{pr} = \left( (1 - \alpha) M^T + \alpha E^T \right) \]
- All use power iteration

Making the ranking sensitive to query

- **HITS**
  - By sampling a subset of web pages nearby each query
- **Google**
  \[ \text{score}(d,q) = f(\text{IRscore}(d,q), \text{PageRank}(d)) \]
- **Other way to make the ranking sensitive to a query?**
  \[ r^{(k)} := B_{pr} r^{(k-1)} = B_{pr}^k r^{(0)} \quad \text{where} \quad B_{pr} = \left( (1 - \alpha) M^T + \alpha E^T \right) \]
  - Let's change the above formula.
How can we inject personal preference in PageRank?

\[ r^{(k)} := B_{pr} r^{(k-1)} = B_{pr} r^{(0)} \]

- Should we change matrix B or initial vector \( r^{(0)} \)?
  - Personalize the initial vector?
  - Personalize matrix B?
    - Which part (M or E) in B shall we personalize?

Personalized PageRank (PPR) (Haveliwala et al., 2003, Stanford TR)

- Personalized the teleportation matrix as:

\[
E_u = \tilde{1} p_u^T = \begin{cases} 
1 & \text{if } u = \tilde{1} \\
1 & \text{otherwise}
\end{cases}
\]

where \( p_{ui} \in [0,1] \), \( \sum_{i=1}^{n} p_{ui} = 1 \)

- The personalization vector is defined as the probabilistic distribution of the user-interested web pages

Standard PageRank is just a specific case, i.e., making p uniform.
Personalized PageRank (PPR) (cont’d)

- Iterative updating as
  \[ r_u^{(k)} := B_u^T r_u^{(k-1)} \]
  \[ = (1 - \alpha)M + \alpha E_u r_u^{(k-1)} \]
  \[ = (1 - \alpha)M^T r_u^{(k-1)} + \alpha E_u^T r_u^{(k-1)} = (1 - \alpha)M^T r_u^{(k-1)} + p \]

- Equivalently
  \[ r_u^{(k)} := B_u^k r_u^{(0)} \]

---

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

- Would this condition be violated in PPR?

---
A Remedy

\[ B_u^T = (1 - \alpha)M + \alpha E_u \quad \text{where} \quad E_u = \bar{1} p_u^T \]

\[ \downarrow \]

\[ B_u^T = \alpha M + \beta E_u + \gamma E \quad \text{where} \quad E = \frac{1}{n} 1 1^T \]

\[ \alpha, \beta, \gamma \in (0,1) \quad \text{and} \quad \alpha + \beta + \gamma = 1 \]

Outline

✓ Part I
  ✓ Hubs and Authorities (HITS)
  ✓ PageRank

▪ Part II
  ✓ Personalized PageRank
    ▪ Topic-sensitive PageRank

▪ Part III. Evaluation of Ranked Lists
**Topic Sensitive PageRank (TSPR)**

1) Denote by \( t \) a topic, and by \( n_t \) be the number of web pages in topic \( t \).

2) For each topic, construct a topic-specific teleportation vector as:
   \[
   p_t = \left( p_{i1}, p_{i2}, \ldots, p_{in_t} \right)^T, \quad p_n = \begin{cases} 
   \frac{1}{n_t} & \text{if page } i \in t \\
   0 & \text{otherwise}
   \end{cases}
   \]

3) For each topic, construct the hybrid matrix as
   \[
   B_t^T = \alpha M + \beta E_t + \gamma E
   \]
   where \( E = \frac{1}{n} \mathbf{1} \mathbf{1}^T \), \( E_t = \frac{1}{n_t} \tilde{1} \tilde{p}^T_t \) and \( \alpha + \beta + \gamma = 1. \)

---

**The topic-specific matrix in TSPR**

\[
B_t^T = \alpha M + \beta E + \gamma E_t \quad \text{where} \quad E_t = \tilde{1} p_t^T
\]

*e.g.* \( E_t = \begin{bmatrix}
0.33 & 0 & 0.33 & 0.33 & 0 \\
0.33 & 0 & 0.33 & 0.33 & 0 \\
: & : & : & : & : \\
: & : & : & : & : \\
0.33 & 0 & 0.33 & 0.33 & 0
\end{bmatrix} \quad \text{with} \quad n_t = 3.\)
The random walk intuition

- The updating rule
  \[ r_t^{(k)} := B_t r_t^{(k-1)} = (\alpha M + \beta E_t + \gamma E) r_t^{(k-1)} \]
  where \( E_t = 1p_t^T \) and \( E = \frac{1}{n} 11^T \).

- Efficient computation
  \[ r_t^{(k)} := \alpha M^T r_t^{(k-1)} + \beta p_t + \gamma p_0 \quad \text{where} \quad p_0 = \frac{1}{n} \]

Merging topic-specific ranked lists

- Offline computation of the TSPR vectors for \( t = 1, \ldots, T \) 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:
  \[ r_q^{(TSPR)} = \sum_{t=1}^{T} \Pr(t \mid q) \times r_t \]
How do we estimate \( P(t | q) \)?

- **Method 1: NB**
  \[
  \Pr(t | q) = \frac{\Pr(t) \Pr(q | t)}{\Pr(q)} = \frac{\Pr(t) \Pr(q | t)}{\sum_{t' \in T} \Pr(t') \Pr(q | t')}
  \]

- **Method 2: kNN**
  \[
  \hat{\Pr}(t | q) = \frac{\sum_{x_i \in kNN(q)} \delta(y_i, t)}{k}, \quad \delta(y_i, t) = \begin{cases} 1 & \text{if } y_i = t \\ 0 & \text{otherwise} \end{cases}
  \]

- **Method 3: Logistic Regression**
  \[
  \Pr(t = k | q) = \frac{1}{1 + \exp(-w_{k0} - w_k^T \cdot q)} \quad w_{k0} \in R, \ w_k \in R^m, \ q \in R^m
  \]

- ...
Naïve Bayes Model (cont’d)

- Estimate topic prior
  \[ \hat{P}(c_k) = \frac{\text{count}(c_k, D_v)}{|D_v|} \equiv w_{k0} \]

- Estimate conditional probabilities:
  \[ P(x_i | c_k) \propto \prod_{j=1}^{J} P(t_j | c_k)^{n_j}, \quad \hat{P}(t_j | c_k) = \frac{n(t_j, D^{(k)})}{\sum_{j=1}^{J} n(t_j, D^{(k)})} \equiv w_{kj} \]

- Model parameters of each class:
  \[ (w_{k0}, w_{k1}, \ldots, w_{kJ}), \quad \forall k = 1, 2, \ldots, K \]

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 doc’s among the top n in the ranked list for each query is averaged over queries.
- **Mean Reciprocal Rank (MRR)**: The inverse of the rank of the 1st relevant doc in each ranked list is averaged over queries.
- **Mean Average Precision (MAP)**: The mean of the precision scores at all relevant doc’s in each ranked list is averaged over queries.
- **Normalized Discounted Cumulated Gain (NDCG)**: allowing multi-scale relevance judgments
- **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
Precision and Recall @n

- Precision \( (P) \)
  \[
P = \frac{\text{Number of retrieved and relevant items}}{\text{Number of retrieved items}}
\]

- Recall \( (R) \)
  \[
R = \frac{\text{Number of retrieved and relevant items}}{\text{Number of relevant items in collection}}
\]

**Ranked List**

\[
P @ 5 = \frac{2}{5} = 40\%
\]

\[
R @ 5 = \frac{2}{3} = 67\%
\]

Consider a toy dataset with 8 doc's in total: with 3 relevant (red) and 5 irrelevant (gray)

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

<table>
<thead>
<tr>
<th>Rank</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Rank sum of the red ones: 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.

We can modify the method as follows:

- Normalize the scores using the maximum (8) and the minimum (1), making them ranging between 1 and 0
- Average the scores over all the positions of relevant doc's (+'s)
- Average the per-query scores

Average Precision (AP)

Ranked List (assuming complete)

\[
P @ 2 = \frac{1}{2} = 50\
\]

\[
P @ 5 = \frac{2}{5} = 40\
\]

\[
P @ 8 = \frac{3}{8} = 37.5\
\]

\[
AP = \frac{(50\% + 40\% + 37.5\%)}{3} = 42.5\%
\]
The Precision-Recall Graph: An Example

Equivalent to the volume of the shaded area underlying the projected line of each precision value to the left

\[
AP = \frac{(0.5 + 0.4 + 0.375)}{3} = 0.425
\]

Mean Average Precision (MAP)

\[
\text{System on query 1: } \quad AP = \frac{(0.5 + 0.4 + 0.375)}{3} = 0.425
\]

\[
\text{System on query 2: } \quad AP = \frac{(0.5 + 0.67)}{2} = 0.585
\]

\[
\text{System on queries 1 and 2: } \quad MAP = \frac{(0.425 + 0.585)}{2} = 0.55
\]
RAW RECALL-PRECISION CURVE

Total # of relevant doc's (red) = 4

Recall | Precision
--------|--------
        |        
        |        
        |        
        |        
        |        
        |        
        |        
        |        

11-point Interpolated Precision
(at recall levels of 0%, 10%, ..., 100%)

\[ p_{\text{interp}} = \max_{r', r' \leq r} p(r') \]
\[ AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1.0\}} p_{\text{interp}}(r) \]
\[ MAP = \frac{1}{|Q|} \sum_{q \in Q} AP(q) \]
11-pt Average Precision Curves

Example of performance curves (averaged over all queries)

Mean Average Precision (MAP)

- Most common in IR evaluations
- Based on the precision values at the positions of relevant documents in each ranked list
- Spiritually similar to the rank-sum based metric for comparing ranked lists
- Should be evaluated over the complete ranked list (in theory) per query
- Averaged AP values over all queries
- Giving more weights to higher-ranking rel. doc’s
**Average Precision (AP)**

*Gives more weights to rel. doc's in higher positions*

<table>
<thead>
<tr>
<th>n</th>
<th>1st doc</th>
<th>2nd doc</th>
<th>3rd doc</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1/2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1/5</td>
<td>1/5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1/8</td>
<td>1/8</td>
<td>1/8</td>
</tr>
</tbody>
</table>

\[
AP = \frac{1/2 + 2/5 + 3/8}{3} = 42.5\%
\]

---

**Normalized Discounted Cumulative Gain (NDCG, Jarvelin & Kekalainen, ACM TOIS 2002)**

**Graded Relevance:** Documents may not be equally relevant, so we want graded relevance as

- 3: The best choice
- 2: A very good choice
- 1: Acceptable
- 0: Not relevant
Normalized Discounted Cumulative Gain

Gain from reading a document
\[ G(d_i | L_q) = 2^{R(d_i, q)} - 1 \]
- \( q \) is a query
- \( L_q \) is a ranked list given \( q \)
- \( d_i \in L_q \) is the document with rank \( i \)
- \( R(d_i, q) \in \{0, 1, 2, 3\} \) is the graded relevance

Discounted Cumulated Gain from the list
\[ DCG(L_q) = \sum_{i=1}^{|L_q|} \frac{2^{R(d_i, q)} - 1}{\log_2 (1 + i)} \text{ Discount} \]

Normalized
\[ NDCG(L_q) = \frac{DCG(L_q)}{DCG(L_q^*)} \]
\( L_q^* \) is an ideal list for \( q \).

NDCG Calculation: Ideal List

<table>
<thead>
<tr>
<th>Rank (i)</th>
<th>Rel (r)</th>
<th>( 2^r - 1 )</th>
<th>( \log_2 (i) )</th>
<th>DG</th>
<th>DCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>1.00</td>
<td>7.00</td>
<td>7.00</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1.58</td>
<td>1.89</td>
<td>8.89</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2.00</td>
<td>1.50</td>
<td>10.39</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2.32</td>
<td>0.43</td>
<td>10.82</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2.58</td>
<td>0.00</td>
<td>10.82</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>2.81</td>
<td>0.00</td>
<td>10.82</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>3.00</td>
<td>0.00</td>
<td>10.82</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>3.17</td>
<td>0.00</td>
<td>10.82</td>
</tr>
</tbody>
</table>

Ideal DCG
NDCG Calculation: An Example

<table>
<thead>
<tr>
<th>Rank (i)</th>
<th>Rel (r)</th>
<th>$2^r - 1$</th>
<th>$\log_2 (i)$</th>
<th>DG</th>
<th>DCG</th>
<th>NDCG</th>
<th>Ideal DCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>1.00</td>
<td>7.00</td>
<td>7.00</td>
<td>1.00</td>
<td>7.00</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1.58</td>
<td>1.89</td>
<td>8.89</td>
<td>1.00</td>
<td>8.89</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2.00</td>
<td>0.00</td>
<td>8.89</td>
<td>0.86</td>
<td>10.39</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2.32</td>
<td>0.43</td>
<td>9.32</td>
<td>0.86</td>
<td>10.82</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2.58</td>
<td>0.00</td>
<td>9.32</td>
<td>0.86</td>
<td>10.82</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>3</td>
<td>2.81</td>
<td>1.07</td>
<td>10.39</td>
<td>0.96</td>
<td>10.82</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>3.00</td>
<td>0.00</td>
<td>10.39</td>
<td>0.96</td>
<td>10.82</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>3.17</td>
<td>0.00</td>
<td>10.39</td>
<td>0.96</td>
<td>10.82</td>
</tr>
</tbody>
</table>

Summary

- Popularity can be defined recursively.
- Popularity can be personalized and 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, why the results converge, and how to engage link analysis with user’s interests in IR and other applications (e.g., social network analysis).
References