

# Personalized Information Retrieval

Shihn-Yuarn Chen

# Traditional Information Retrieval

- Content-based approaches
  - Statistical and natural language techniques
  - Results that contain a specific set of words or meaning, but cannot differentiate which documents in a collection are the ones really worth reading.
- Citation and hyperlink approaches
  - An implicit measure of importance.
  - Create an authoring bias where the meaning and resources valued by a group of authors determine the results for the entire user population.
- Consensus relevancy, not individual relevancy.

# Apply Usage Data into IR

- Usage-based IR methods
  - Actions of users to compute relevancy.
  - The retrieval process can be infused with different “granularities” of usage data—individual, group/social, and census.
    - More individual usage data → more personalized
    - More social usage data → collaborative filtering → recommendation.

# Directions to Personalized Search

- Query augmentation
  - Old topics
    - Reinforce the query or suggest results from prior searches
    - Query history and query expansion
  - New topics
    - Diverse the search results
    - Filip Radlinski, Susan Dumais, *“Improving Personalized Web Search using Result Diversification”*, SIGIR 2006
- Result processing
  - Filtering
  - Re-ranking

# Old Methods for Collecting Users' Preference

- Force users to input their profile.
- Relevance feedback. (good result, bad result)
- However, all users are lazy.

# Modify PageRank for Personalized Search

# PageRank

- when a page  $p_0$  links to a page  $p$ , it is probably because the author of page  $p_0$  thinks that page  $p$  is important.
- this link ( $p_0 \rightarrow p$ ) adds to the importance score of page  $p$ .
- How much score should be added for each link?
  - Intuitively, if a page itself is very important, then its author's opinion on the importance of other pages is more reliable; and if a page links to many pages, the importance score it confers to each of them is decreased.

$$PR(p) = \sum_{p_0 \in \mathcal{A}_p} PR(p_0) / l_{p_0}$$

$$PR(p) = d * \sum_{p_0 \in \mathcal{A}_p} PR(p_0) / l_{p_0} + (1 - d) * E(p)$$

# Topic Sensitive PageRank

$$TSPR_t(p) = d * \sum_{p_0 \in \mathcal{A}_p} TSPR_t(p_0) / l_{p_0} + (1 - d) * E_t(p).$$

$$E_t(p) = \begin{cases} 1/n_t & \text{if page } p \text{ is related to topic } t \\ 0 & \text{otherwise,} \end{cases}$$



# User's Preference

$$V(p) = \sum_{i=1}^m T(i) * TSPR_i(p)$$

- Prior research\* show that when a user's clicks are affected by search results ranked by  $PR(p)$ , the user's visit probability to page  $p$ ,  $V(p)$ , is proportional to  $PR(p)^{9/4}$ , as opposed to  $PR(p)$  as predicted by the random surfer model.

$$V(p) = \sum_{i=1}^m T(i) * [TSPR_i(p)]^{9/4}$$

[\*] J. Cho and S. Roy. Impact of Web search engines on page popularity. In Proc. of WWW '04, 2004.

# Ranking Search Results Using Topic Preference Vectors

$$\sum_{t=1}^m Pr(T(i)|q) \cdot TSPR_i(p)$$

$$\begin{aligned} Pr(T(i)|q) &= \frac{Pr(q, T(i))}{Pr(q)} \\ &= \frac{Pr(T(i)) * Pr(q|T(i))}{Pr(q)} \\ &\propto Pr(T(i)) * Pr(q|T(i)) \end{aligned}$$

$$PPR_T(p) = \sum_{t=1}^m T(i) \cdot Pr(q|T(i)) \cdot TSPR_i(p)$$

# Query Log & HITS-like Algorithm

# HITS

- Hyperlink-Induced Topic Search (by J. Kleinberg, 1998)
  - Detection of high-score hub and authority web pages.
    - Good authority pages
      - In the context of particular query topics
      - Less out-links, more in-links (especially links from good hub pages)
    - Good hub pages
      - Pages have more links to good authority pages.

# HITS-like iterative algorithm

- Consider unseen pages as authority pages, and representative terms as hub pages.

Approaches	The Directed Graph		
	Nodes		Edges
HITS	Authority Pages	Hub Pages	Hyperlinks
Our Approach	Unseen Search Results	Representative Terms	Occurrence <sup>2</sup>

Table 1. Our approach versus HITS.

# HITS-like iterative algorithm

- Construct a directed graph (of representative terms and unseen pages)
  - $G = (V, E)$
  - $V$ : unseen pages & representative terms
  - $E$ :  $p \rightarrow q$ , is weighted by the freq. of occ. of a representative term  $p$  in an unseen page  $q$ .

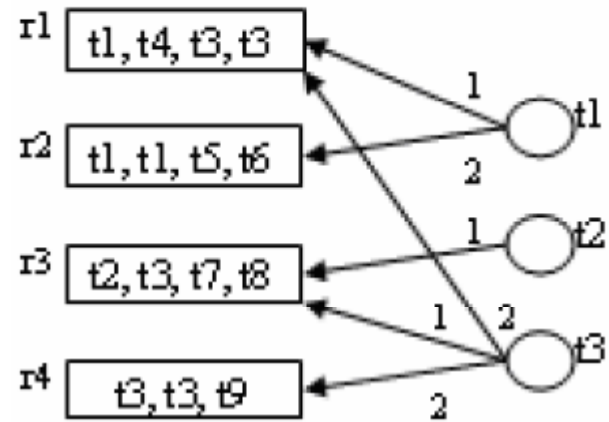


Figure 1. A sample directed graph.

# HITS-like iterative algorithm

- HITS-like iterative algo.

- Initialization

- Equal authoritative for unseen pages.

$$y_1^0 = y_2^0 \cdots = y_{|Y|}^0 = 1/|Y|$$

- Term score: tf in the history query logs

$$x_j^0 = tf_j / \sum_{i=1}^{|X|} tf_i$$

- Associate the weight to each edge.

$$w(t_i \rightarrow r_j) = tf_{i,j}$$

$tf_{i,j}$  is the term freq. of term  $i$  occurring in page  $j$ .

# HITS-like iterative algorithm

- HITS-like iterative algo.
  - Recompute the hub score

$$x'_i{}^{(k+1)} = \sum_{\forall j: t_i \rightarrow r_j} y'_j{}^k \frac{w(t_i \rightarrow r_j)}{\sum_{\forall n: t_n \rightarrow r_j} w(t_n \rightarrow r_j)}$$

- Recompute the authority score

$$y'_j{}^{(k+1)} = \sum_{\forall i: t_i \rightarrow r_j} x'_i{}^k \frac{w(t_i \rightarrow r_j)}{\sum_{\forall m: t_i \rightarrow r_m} w(t_i \rightarrow r_m)}$$

- After recomputing, normalization

$$y_j = \frac{y'_j}{\sum_{k=1}^{|Y|} y'_k} \text{ and } x_i = \frac{x'_i}{\sum_{k=1}^{|X|} x'_k}$$



# HITS-like iterative algorithm

- HITS-like iterative algo.
    - When to stop?
      - The changes of hub scores and the authority scores are smaller than predefined threshold
- $$c = \sum_{j=1}^{|Y|} (\mathbf{y}_j^{(k+1)} - \mathbf{y}_j^k)^2 + \sum_{i=1}^{|X|} (\mathbf{x}_i^{(k+1)} - \mathbf{x}_i^k)^2$$
- Or the # of iterations is larger than a predefined times

# HITS-like iterative algorithm

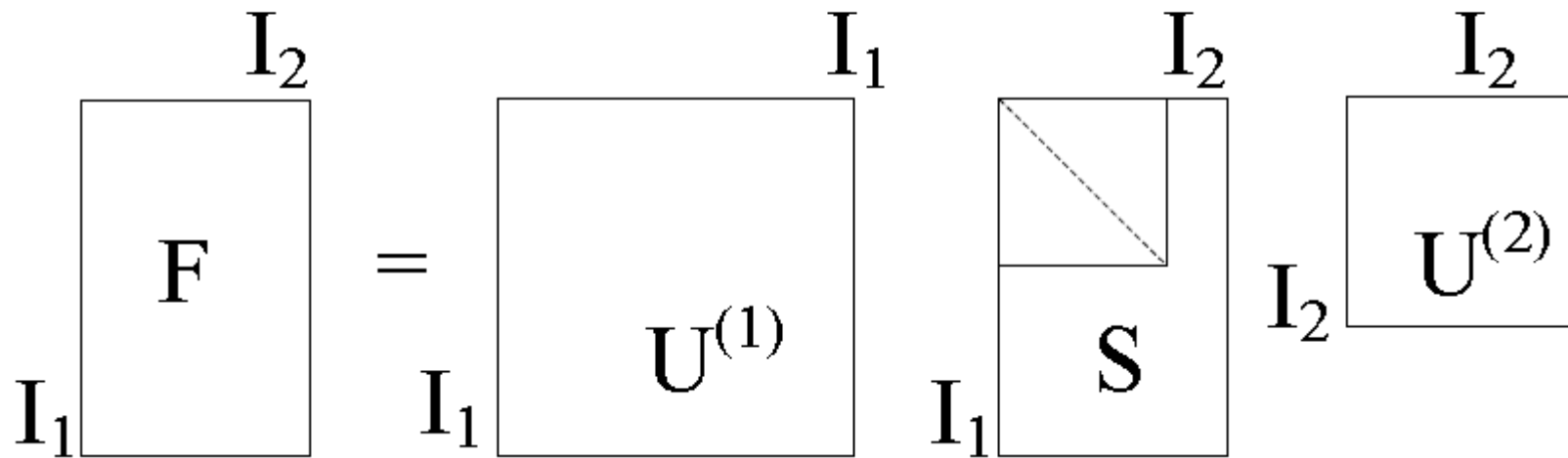
- HITS-like iterative algo.
  - Select result pages and select terms for query expansion.
    - Top  $n$ (predefined) unseen search results with highest authority scores are selected for recommendation
    - Top  $m$  representative terms with highest hub scores are selected to expand the original query.
      - $m$  is determined according to the position of the biggest gap, that is, if  $t_i - t_{i+1}$  is bigger than the gap of any other two neighboring ones of the top half representative terms, then  $m$  is given a value  $i$ .

CubeSVD

# Related Work

- Higher-Order Singular Value Decomposition (HOSVD)
  - L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

# SVD



The diagram illustrates the SVD decomposition of a matrix  $F$ . It shows the equation  $F = U^{(1)} S U^{(2)}$ . Matrix  $F$  is a rectangle with height  $I_1$  and width  $I_2$ . Matrix  $U^{(1)}$  is a square with side length  $I_1$ . Matrix  $S$  is a rectangle with height  $I_1$  and width  $I_2$ , containing a diagonal line from the top-left to the bottom-right. Matrix  $U^{(2)}$  is a square with side length  $I_2$ .

- By setting the smallest  $(\min\{I_1, I_2\} - k)$  singular values in  $S$  to zero, the matrix  $F$  is approximated with a rank- $k$  matrix and this approximation is best measured in reconstruction error.

# HOSVD

- A tensor is a higher order generalization of a vector
  - 1<sup>st</sup> order tensor is a vector
  - 2<sup>nd</sup> order tensor is a matrix
- The order of a tensor  $A \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is  $N$
- Elements of  $A$  are denoted as  $a_{i_1 \dots i_n \dots i_N}$  where  $1 \leq i_n \leq I_n$
- The mode- $n$  vectors of an  $N$ -th order tensor  $A$  are the  $I_n$ -dimensional vectors obtained from  $A$  by varying the index  $i_n$  and keeping the other indices fixed, that is the column vectors of  $n$ -mode matrix unfolding  $A_{(n)} \in \mathbb{R}^{I_n \times (I_1 I_2 \dots I_{n-1} I_{n+1} \dots I_N)}$  of tensor  $A$

# HOSVD

The  $n$ -mode product of a tensor  $\mathcal{A} \in R^{I_1 \times I_2 \times \dots \times I_N}$  by a matrix  $M \in R^{J_n \times I_n}$  is an  $I_1 \times I_2 \times \dots \times I_{n-1} \times J_n \times I_{n+1} \times \dots \times I_N$ -tensor of which the entries are given by

$$(\mathcal{A} \times_n M)_{i_1 \dots i_{n-1} j_n i_{n+1} \dots i_N} = \sum_{i_n} a_{i_1 \dots i_{n-1} i_n i_{n+1} \dots i_N} m_{j_n i_n}$$

- $M_{5 \times 7} \times A_{7 \times 3} = MA_{5 \times 3}$
- $M_{J_n \times I_n} \times A_{I_n \times I_{n+1}} = MA_{J_n \times I_{n+1}}$

# HOSVD

Note that the  $n$ -mode product of a tensor and a matrix is a generalization of the product of two matrices. It can be expressed in terms of matrix unfolding:

$$B_{(n)} = MA_{(n)} \quad (3)$$

where  $B_{(n)}$  is the  $n$ -mode unfolding of tensor  $\mathcal{B} = \mathcal{A} \times_n M$ .

In terms of  $n$ -mode products, the matrix SVD can be rewritten as  $F = S \times_1 V^{(1)} \times_2 V^{(2)}$ . By extension, HOSVD is a generalization of matrix SVD: every  $I_1 \times I_2 \times \cdots \times I_N$  tensor  $\mathcal{A}$  can be written as the  $n$ -mode product [15]:

$$\mathcal{A} = \mathcal{S} \times_1 V_1 \times_2 V_2 \cdots \times_N V_N \quad (4)$$

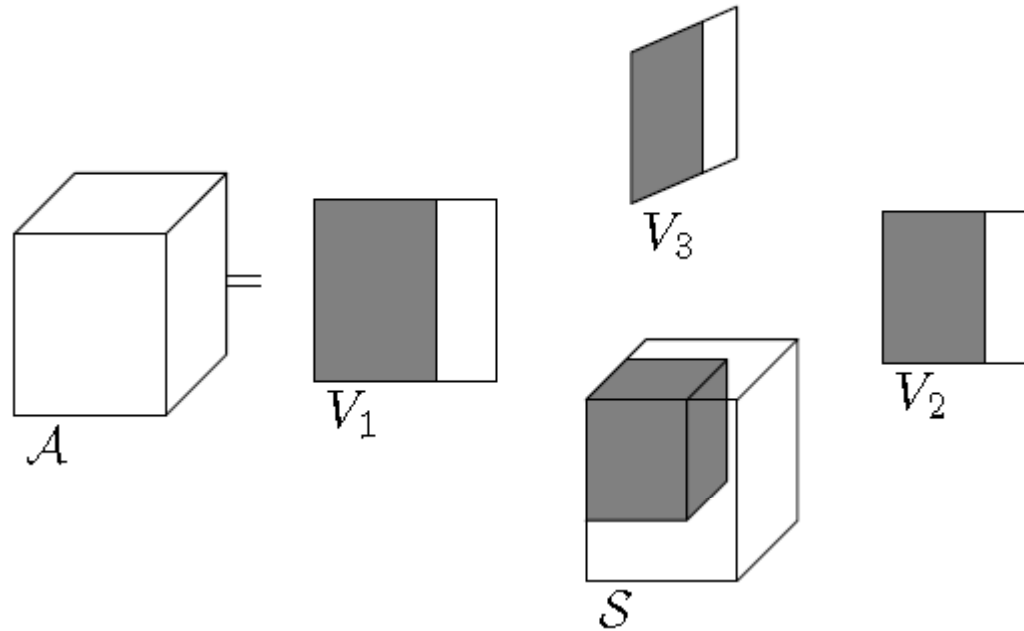


# HOSVD

- *S* is called *core tensor*.
- *Instead of* being pseudodiagonal (nonzero elements only occur when the indices satisfy  $i_1 = i_2 = \dots = i_N$ ),
- *S* has the *property* of all-orthogonality.
  - two subtensors  $S_{i_n=\alpha}$  and  $S_{i_n=\beta}$  are orthogonal for all possible values of  $n$ ,  $\alpha$  and  $\beta$  subject to  $\alpha \neq \beta$ .

# CubeSVD

- (user, query, web page) =  $R^{m \times n \times k}$



1. Construct tensor  $\mathcal{A}$  from the clickthrough data. Suppose the numbers of user, query and Web page are  $m$ ,  $n$ ,  $k$  respectively, then  $\mathcal{A} \in R^{m \times n \times k}$ . Each tensor element measures the preference of a  $\langle user, query \rangle$  pair on a Web page.
2. Calculate the matrix unfolding  $A_u$ ,  $A_q$  and  $A_p$  from tensor  $\mathcal{A}$ .  $A_u$  is calculated by varying user index of tensor  $\mathcal{A}$  while keeping query and page index fixed.  $A_q$  and  $A_p$  are computed in a similar way. Thus  $A_u$ ,  $A_q$ ,  $A_p$  is a matrix of  $m \times nk$ ,  $n \times mk$ ,  $k \times mn$  respectively.
3. Compute SVD on  $A_u$ ,  $A_q$  and  $A_p$ , set  $V_u$ ,  $V_q$  and  $V_p$  to be the left matrix of the SVD respectively.
4. Select  $m_0 \in [1, m]$ ,  $n_0 \in [1, n]$  and  $k_0 \in [1, k]$ . Remove the right-most  $m - m_0$ ,  $n - n_0$  and  $k - k_0$  columns from  $V_u$ ,  $V_q$  and  $V_p$ , then denote the reduced left matrix by  $W_u$ ,  $W_q$  and  $W_p$  respectively. Calculate the core tensor as follows:

$$\mathcal{S} = \mathcal{A} \times_1 W_u^T \times_2 W_q^T \times_3 W_p^T \quad (5)$$

5. Reconstruct the original tensor by:

$$\hat{\mathcal{A}} = \mathcal{S} \times_1 V_u \times_2 V_q \times_3 V_p \quad (6)$$