SVD Applications: LSI and Link Analysis

Advanced Machine Learning Course
Shaghayegh Sahebi (Sherry)
Outline

- QR Factorization
- Latent Semantic Indexing (LSI)
- Kleinberg’s Algorithm (HITS)
- PageRank Algorithm (Google)
Vector Space Model

Documents

D1: How to bake bread without recipes
D2: The classic art of Viennese pastry
D3: Numerical recipes: The art of scientific computing
D4: Breads, pastries, pies and cakes: quantity baking recipes
D5: Pastry: A book of best french recipes

Terms

T1: bak(e,ing)  T2: recipes  T3: bread  T4: cake  T5: pastr(y,ies)  T6: pie
Vector Space Model

- Vector space model represents database as a vector space
  - In indexing terms space

- Each document represented as a vector
  - weight of the vector: semantic importance of indexing term in the document

- queries are modeled as vectors
Vector Space Model

- Whole database: \(d\) documents described by \(t\) terms
  - \(t \times d\) term-by-document matrix

\[
A = \begin{pmatrix}
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{pmatrix}
\]

\[
\hat{A} = \begin{pmatrix}
0.5774 & 0 & 0 & 0.4082 & 0 \\
0.5774 & 0 & 1 & 0.4082 & 0.7071 \\
0.5774 & 0 & 0 & 0.4082 & 0 \\
0 & 0 & 0 & 0.4082 & 0 \\
0 & 1 & 0 & 0.4082 & 0.7071 \\
0 & 0 & 0 & 0.4082 & 0
\end{pmatrix}
\]

- the semantic content of the database is wholly contained in the column space of \(A\)
Similarity Measure

• How to identify relevant documents?

• Using spatial proximity for semantic proximity
  - Most relevant documents for a query ≈ those with vectors closest to the query

• **Cosine measure**: the most widespread similarity measure
  - the cosine of the angle between two vectors.
  - Unit vectors ➔ cosine measure = a simple dot product

\[
\cos(\overline{x}, \overline{y}) = \frac{\overline{x} \cdot \overline{y}}{||\overline{x}|| \cdot ||\overline{y}||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}
\]
Example

- A vector space with two dimensions
- Three documents and one query (unit vectors)
- D2 is the most similar document to query q
Term weighting

- Simplest term (vector component) weightings:
  - count of number of times word occurs in document
  - binary: word does or doesn’t occur in document

- A document is a better match if a word occurs three times than once, but not a three times better match
  - → a series of weighting functions e.g., $1+\log(x)$ if $x > 0$

- Significance of a term:
  - occurrence of a term in a document is more important if that term does not occur in many other documents
  - Solution: weight=global weight x local weight
QR-Factorization

- Some information are redundant in vector space model → QR factorization

- How it works?
  - Identify a basis for the column space
  - Low rank approximation
Identify a Basis for Column Space

- For a rank $r_A$ matrix $A$:
  - $R$: $t \times d$ upper triangular matrix
  - $Q$: $t \times t$ orthogonal matrix

$$A = QR$$

$$A = (Q_A Q_A^\perp) \begin{pmatrix} R_A \\ 0 \end{pmatrix} = Q_A R_A$$

$$\cos \theta_j = \frac{a_j^T q}{\|a_j\|_2 \|q\|_2} = \frac{(Q_A r_j)^T q}{\|Q_A r_j\|_2 \|q\|_2} = \frac{r_j^T (Q_A^T q)}{\|r_j\|_2 \|q\|_2}$$

$$\|

\begin{pmatrix}
\end{pmatrix}$$
Example

\[
Q = \begin{pmatrix}
-0.5774 & 0 & -0.4082 & 0 & -0.7071 & 0 \\
-0.5774 & 0 & 0.8165 & 0 & 0.0000 & 0 \\
-0.5774 & 0 & -0.4082 & 0 & 0.7071 & 0 \\
0 & 0 & 0 & -0.7071 & 0 & -0.7071 \\
0 & -1.0000 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.7071 & 0 & 0.7071
\end{pmatrix},
\]

\[
R = \begin{pmatrix}
-1.0001 & 0 & -0.5774 & -0.7070 & -0.4082 \\
0 & -1.0000 & 0 & 0 & 0 \\
0 & 0 & 0.8165 & 0 & 0.5774 \\
0 & 0 & 0 & -0.5774 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}.
\]
Query Matching

\[ q = Iq = QQ^T \]
\[ = [Q_A Q_A^T + Q_A (Q_A^T)^T] q \]
\[ = Q_A Q_A^T q + Q_A (Q_A^T)^T q \]
\[ = q_A + q_A^\perp. \]

\[ \cos \theta_j = \frac{a_j^T q_A + a_j^T q_A^\perp}{\| a_j \|_2 \| q \|_2} = \frac{a_j^T q_A + a_j^T Q_A^\perp (Q_A^T)^T q}{\| a_j \|_2 \| q \|_2}. \]

\[ \cos \theta_j = \frac{a_j^T q_A + 0 \cdot (Q_A^T)^T q}{\| a_j \|_2 \| q \|_2} = \frac{a_j^T q_A}{\| a_j \|_2 \| q \|_2}. \]

\[ \cos \theta_j' = \frac{a_j^T q_A}{\| a_j \|_2 \| q_A \|_2}. \]
Low Rank Approximation

- Change in the DB or not being precise:
  - Use approximation of $A$: $A + E$ (Uncertainty Matrix)

- What if adding $E$ reduces the rank of $A$
  - $A$ can be partitioned to isolate smaller parts of the entries

\[
R = \begin{pmatrix}
-1.0001 & 0 & -0.5774 \\
0 & -1.0000 & 0 \\
0 & 0 & 0.8165 \\
\end{pmatrix}
\begin{pmatrix}
-0.7070 & -0.4082 \\
-0.4082 & -0.7071 \\
0 & 0 & 0.5774 \\
\end{pmatrix}
= \begin{pmatrix}
R_{11} & R_{12} \\
0 & R_{22} \\
\end{pmatrix}.
\]
QR- Factorization Problem

- Gives no information of row space
- Doesn’t choose the smallest values → Could be more precise
- Inability to address two problems
  - Synonymy: two different words (say car and automobile) have the same meaning
  - Polysemy: a term such as charge has multiple meanings
- Synonymy → underestimate true similarity
- Polysemy → overestimate true similarity
- Solution: LSI
  - Use the co-occurrences of terms to capture the latent semantic associations of terms?
Latent Semantic Indexing (LSI)

- Approach: Employing a low rank approximation to the vector space representation

- Goal: Cluster similar documents which may share no terms in the latent semantic space, which is a low-dimensional subspace. (improves recall)

- LSI projects queries and documents into a space with latent semantic dimensions.
  - co-occurring words are projected on the same dimensions
  - non-co-occurring words are projected onto different dimensions

- Thus, LSI can be described as a method for dimensionality reduction
Latent Semantic Indexing (LSI)

- Dimensions of the reduced semantic space correspond to the axes of greatest variation in the original space (closely related to PCA)
- LSI is accomplished by applying SVD to term-by-document matrix

Steps:
- Preprocessing: Compute optimal low-rank approximation (latent semantic space) to the original term-by-document matrix with help of SVD
- Evaluation: Rank similarity of terms and docs to query in the latent semantic space via a usual similarity measure

Optimality dictates that the projection into the latent semantic space should be changed as little as possible measured by the sum of the squares of differences
Example

- $A$: term-by-document matrix with rank 5
- Reduced to two dimensions (latent dimensions, concepts)
- In the original space the relation between $d_2$ and $d_3$ is not clear
Singular Value Decomposition (SVD)

- Decomposes \( A_{txd} \) into the product of three matrices \( T_{txn} \), \( S_{nxn} \) and \( D_{dxn} \):

\[
A_{txd} = T_{txn} S_{nxn} (D_{dxn})^T
\]

- \( T \) and \( D \): have orthonormal columns

- \( S \): diagonal matrix containing singular values of \( A \) in descending order. The number of non-zero singular values equals the rank of \( A \).
Singular Value Decomposition (SVD)

- Columns of $T$: orthogonal eigenvectors of $AA^T$
- Columns of $D$: orthogonal eigenvectors of $A^TA$
- LSI defines:
  - $A$ as term-by-document matrix
  - $T$ as term-to-concept similarity matrix
  - $S$ as concept strengths
  - $D$ as concept-to-doc similarity matrix
- If rank of $A$ is smaller than term count, we can directly project into a reduced dimensionality space. However, we may also want to reduce the dimensionality of $A$ by setting small singular values of $S$ to zero.
Dimensionality Reduction

- Compute SVD of $A_{txd} = T_{txn} S_{nxn} (D_{dxn})^T$
- Form $A_{txk}^\wedge = T_{txk} S_{kxk} (D_{kxn})^T$ by replacing the $r-k$ smallest singular values on the diagonal by zeros, which is the optimal reduced rank-$k$ approximation of $A_{txd}$
- $B_{txk}^\wedge = S_{kxk} (D_{kxn})^T$ builds the projection of documents from the original space to the reduced rank-$k$ approximation
  - in the original space, $n$ dimensions correspond to terms
  - in the new reduced space, $k$ dimensions correspond to concepts
- $Q_k = (T_{txk})^T Q_t$ builds the projection of the query from the original space to the reduced rank-$k$ approximation
- Then we can rank similarity of documents to query in the reduced latent semantic space via a usual similarity measure
Example-SVD

\[
A = \begin{pmatrix}
\text{dim}1 & \text{dim}2 & \text{dim}3 & \text{dim}4 & \text{dim}5 & \text{dim}6 \\
\text{cosmonaut} & 1 & 0 & 1 & 0 & 0 & 0 \\
\text{astronaut} & 0 & 1 & 0 & 0 & 0 & 0 \\
\text{moon} & 1 & 1 & 0 & 0 & 0 & 0 \\
\text{car} & 1 & 0 & 0 & 1 & 1 & 0 \\
\text{truck} & 0 & 0 & 0 & 1 & 0 & 1
\end{pmatrix}
\]

\[
T = \begin{pmatrix}
\text{dim}1 & \text{dim}2 & \text{dim}3 & \text{dim}4 & \text{dim}5 \\
\text{cosmonaut} & -0.44 & -0.30 & 0.57 & 0.58 & 0.25 \\
\text{astronaut} & -0.13 & -0.33 & -0.59 & 0.00 & 0.73 \\
\text{moon} & -0.48 & -0.51 & -0.37 & 0.00 & -0.61 \\
\text{car} & -0.70 & 0.35 & 0.15 & -0.58 & 0.16 \\
\text{truck} & -0.26 & 0.65 & -0.41 & 0.58 & -0.09
\end{pmatrix}
\]

\[
S = \begin{pmatrix}
2.16 & 0 & 0 & 0 & 0 \\
0 & 1.59 & 0 & 0 & 0 \\
0 & 0 & 1.28 & 0 & 0 \\
0 & 0 & 0 & 1.00 & 0 \\
0 & 0 & 0 & 0 & 0.39
\end{pmatrix}
\]

\[
D^T = \begin{pmatrix}
\text{dim}1 & \text{dim}2 & \text{dim}3 & \text{dim}4 & \text{dim}5 & \text{dim}6 \\
\text{dim}1 & -0.75 & -0.28 & -0.20 & -0.45 & -0.33 & -0.12 \\
\text{dim}2 & -0.29 & -0.53 & -0.19 & 0.63 & 0.22 & 0.41 \\
\text{dim}3 & 0.28 & -0.75 & 0.45 & -0.20 & 0.12 & -0.33 \\
\text{dim}4 & 0 & 0 & 0.58 & 0 & -0.58 & 0.58 \\
\text{dim}5 & -0.53 & 0.29 & -0.63 & 0.19 & 0.41 & -0.22
\end{pmatrix}
\]
**Example-Reduction (rank-2 approx.)**

\[
S' = \begin{pmatrix}
2.16 & 0 & 0 & 0 & 0 \\
0 & 1.59 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

We can get rid of zero valued columns and rows

And have a 2 x 2 concept strength matrix

\[
T' = \begin{pmatrix}
\hline
\text{cosmonaut} & -0.44 & -0.30 & 0 & 0 & 0 \\
\text{astronaut} & -0.13 & -0.33 & 0 & 0 & 0 \\
\text{moon} & -0.48 & -0.51 & 0 & 0 & 0 \\
\text{car} & -0.70 & 0.35 & 0 & 0 & 0 \\
\text{truck} & -0.26 & 0.65 & 0 & 0 & 0 \\
\hline
\end{pmatrix}
\]

We can get rid of zero valued columns

And have a 5 x 2 term-to-concept similarity matrix

\[
D'_{vT} = \begin{pmatrix}
\hline
d1 & d2 & d3 & d4 & d5 & d6 \\
dim1 & -0.75 & -0.28 & -0.20 & -0.44 & -0.33 & -0.12 \\
dim2 & -0.29 & -0.53 & -0.19 & 0.65 & 0.22 & 0.41 \\
dim3 & 0 & 0 & 0 & 0 & 0 & 0 \\
dim4 & 0 & 0 & 0 & 0 & 0 & 0 \\
dim5 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{pmatrix}
\]

We can get rid of zero valued columns

And have a 2 x 6 concept-to-doc similarity matrix

**dim1 and dim2 are the new concepts**
Example—Projection

Original space

\[ Q = \begin{pmatrix}
\cos\text{monaut} & 1 \\
\astronaut & 0 \\
\text{moon} & 0 \\
\text{car} & 0 \\
\text{truck} & 0
\end{pmatrix} \]

Reduced latent semantic space

\[ Q' = \begin{pmatrix}
d\dim1 & -0.44 \\
d\dim2 & -0.30
\end{pmatrix} \]

\[ A = \begin{pmatrix}
\cos\text{monaut} & d1 & 0 & 1 & 0 & 0 & 0 \\
\astronaut & 0 & 1 & 0 & 0 & 0 & 0 \\
\text{moon} & 1 & 1 & 0 & 0 & 0 & 0 \\
\text{car} & 1 & 0 & 0 & 1 & 1 & 0 \\
\text{truck} & 0 & 0 & 0 & 1 & 0 & 1
\end{pmatrix} \]

\[ B = \begin{pmatrix}
d\dim1 & -1.62 & -0.60 & -0.44 & -0.97 & -0.70 & -0.26 \\
d\dim2 & -0.46 & -0.84 & -0.30 & 1.00 & 0.35 & 0.65
\end{pmatrix} \]

\[ \cos(Q, d2) = 0 \]

\[ \cos(Q', d2) = 0.88 \]

We see that query is not related to the d2 in the original space but in the latent semantic space they become highly related, which is true

\[ \text{Max } [\cos(x, y)] = 1 \]
Database as in Graph Model

- Building citation graph and its adjacency matrix
- Represent documents and terms as nodes of the graph
- There is a link from each document to each term if the term appears in that document
- An authoritative word: a commonly used word
- Connected components in the linkage graph: distinct document topics
Kleinberg’s Algorithm

- Extracting information from link structures of a hyperlinked environment
- Basic essentials
  - Authorities
  - Hubs
- For a topic, authorities are relevant nodes which are referred by many hubs
- For a topic, hubs are nodes which connect many related authorities for that topic
- Authorities are defined in terms of hubs and hubs defined in terms of authorities
  - Mutually enforcing relationship (global nature)
Authorities and Hubs

- The algorithm can be applied to arbitrary hyperlinked environments
  - World Wide Web (nodes correspond to web pages with links)
  - Publications Database (nodes correspond to publications and links to co-citation relationship)
Kleinberg’s Algorithm (WWW)

- Is different from clustering
  - Different meanings of query terms

- Addressed problems by the text-based model
  - Self-description of page may not include appropriate keywords
  - Distinguish between general popularity and relevance

- Three steps
  - Create a focused sub-graph of the Web
  - Iteratively compute hub and authority scores
  - Filter out the top hubs and authorities
Root and Base Set

- For the success of the algorithm base set (sub-graph) should be:
  - relatively small
  - rich in relevant pages
  - contains most of the strongest authorities

- Start first with a root set:
  - obtained from a text-based search engine
  - does not satisfy third condition of a useful subgraph

- Solution: extending root set:
  - add any page pointed by a page in the root set to it
  - add any page that points to a page in the root set to it (at most d)
  - the extended root set becomes our base set
Root and Base Set
Two Operations

- $a[p]$ ... authority weight for page $p$
- $h[p]$ ... hub weight for page $p$
- Iterative algorithm
  1. set all weights for each page to 1
  2. apply both operations on each page from the base set and normalize authority and hub weights separately (sum of squares=1)
  3. repeat step 2 until weights converge
Matrix Notation

- $G$ (root set) is a directed graph with web pages as nodes and their links.

- $G$ can be presented as a connectivity matrix $A$
  - $A(i,j)=1$ only if $i$-th page points to $j$-th page.

- Authority weights can be represented as a unit vector $a$
  - $a(i)$ is the authority weight of the $i$-th page.

- Hub weights can be represented as a unit vector $h$
  - $h(i)$ is the hub weight of the $i$-th page.

\[
A = \begin{pmatrix}
  n1 & n2 & n3 & n4 & n5 \\
  n1 & 0 & 1 & 1 & 1 & 0 \\
  n2 & 0 & 0 & 0 & 1 & 0 \\
  n3 & 0 & 0 & 0 & 0 & 1 \\
  n4 & 0 & 0 & 0 & 0 & 0 \\
  n5 & 0 & 0 & 0 & 1 & 0 \\
\end{pmatrix}
\]
Convergence

- Two mentioned basic operations can be written as matrix operations (all values are updated simultaneously)
  - Updating authority weights: $a = A^T h$
  - Updating hub weights: $h = A a$

- After $k$ iterations:
  $$
  a_1 = A^T h_0 \\
  h_1 = A a_1 \\
  h_1 = A A^T h_0 \rightarrow h_k = (A A^T)^k h_0
  $$

- Thus
  - $h_k$ is a unit vector in the direction of $(A A^T)^k h_0$
  - $a_k$ is a unit vector in the direction of $(A^T A)^{k-1} h_0$

- Theorem
  - $a_k$ converges to the principal eigenvector of $A^T A$
  - $h_k$ converges to the principal eigenvector of $A A^T$
Convergence

- \((A^T A)^k x v' \approx (\text{const}) v_1\) where \(k \gg 1\), \(v'\) is a random vector, \(v_1\) is the eigenvector of \(A^T A\)

- Proof:

\[
(A^T A)^k = (A^T A) x (A^T A) x \ldots = (V \Lambda^2 V^T) x (V \Lambda^2 V^t) x \ldots
\]

\[
= (V \Lambda^2 V^T) x \ldots = (V \Lambda^4 V^T) x \ldots = (V \Lambda^{2k} V^T)
\]

Using spectral decomposition:

\[
(A^T A)^k = (V \Lambda^{2k} V^T) = \lambda_1^{2k} v_1 v_1^T + \lambda_2^{2k} v_2 v_2^T + \ldots + \lambda_n^{2k} v_n v_n^T
\]

because \(\lambda_1 > \lambda_{i \neq 1} \Rightarrow \lambda_1^{2k} \gg \lambda_{i \neq 1}^{2k}\)

thus \((A^T A)^k \approx \lambda_1^{2k} v_1 v_1^T\)

now \((A^T A)^k x v' = \lambda_1^{2k} v_1 v_1^T x v' = (\text{const}) v_1\)

because \(v_1^T x v'\) is a scalar.
Sign of Eigenvector

- We know that \((A^TA)^k \approx \lambda_1^{2k} v_1 v_1^T\)
- Since A is the adjacency matrix, elements of \((A^TA)^k\) are all positive
- \(\Rightarrow \lambda_1^{2k} v_1 v_1^T\) should be positive
- \(\lambda_1^{2k}\) is positive \(\Rightarrow v_1 v_1^T\) is positive \(\Rightarrow\) all elements of \(v_1\) should have the same sign (either all elements are positive or all are negative)
Sub-communities

- Authority vector converges to the principal eigenvector of $A^T A$, which lets us choose strong authorities.
- Hub vector converges to the principal eigenvector of $A A^T$ which lets us choose strong hubs.
- These chosen authorities and hubs build a cluster in our network.
- However, there can exist different clusters of authorities and hubs for a given topic, which correspond to:
  - different meanings of a term (e.g. jaguar $\rightarrow$ animal, car, team)
  - different communities for a term (e.g. randomized algorithms)
  - polarized thoughts for a term (e.g. abortion)
- Extension:
  - each eigenvector of $A^T A$ and $A A^T$ represents distinct authority and hub vectors for a sub-community in Graph G, respectively.
PageRank

- PageRank is a link analysis algorithm that assigns weights to nodes of a hyperlinked environment.

- It assigns importance scores to every node in the set which is similar to the authority scores in Kleinberg algorithm.

- It is an iterative algorithm like Kleinberg algorithm.

- Main assumptions:
  - In-degree of nodes are indicators of their importance.
  - Links from different nodes are not counted equally. They are normalized by the out-degree of its source.
Simplified PageRank (WWW)

\[ \Pr(u) = \sum_{v \in B(u)} \frac{\Pr(v)}{L(v)} \]

- \( B(u) \) is the set of nodes which have a link to \( u \)

- PageRank Algorithm simulates a random walk over web pages.
- \( \Pr \) value is interpreted as probabilities
- In each iteration we update \( \Pr \) values of each page simultaneously
- After several passes, \( \Pr \) value converges to a probability distribution used to represent the probability that a person randomly clicking on links will arrive at any particular page
Matrix Notation

Update step

\[ \text{Pr}_{kx1} = M_{kxk} \times \text{Pr}_{kx1} \]

\[ M_{ij} = \begin{cases} 
1 & \text{, if } i \in B_j \\
\frac{1}{|B_j|} & \text{, else} \\
0 & \text{, else}
\end{cases} \]

- M(i,j) is the transition matrix and defines fragment of the j-th page's Pr value which contributes to the Pr value of the i-th page.

k is the number of total pages

B_i is the set of pages which have a link to i-th page
PageRank and Markov Chain

- PageRank defines a Markov Chain on the pages with transition matrix $M$ and stationary distribution $\Pr$.
  - states are pages
  - transitions are the links between pages (all equally probable)

- As a result of Markov theory, $\Pr$ value of a page is the probability of being at that page after lots of clicks.
Matrix Notation

Update step

\[
\begin{pmatrix}
  x \\
  y \\
  z
\end{pmatrix} =
\begin{pmatrix}
  0 & 0.5 & 1 \\
  1 & 0 & 0 \\
  0 & 0.5 & 0
\end{pmatrix}
\times
\begin{pmatrix}
  x \\
  y \\
  z
\end{pmatrix}
\]

\[
x = 0 \cdot x + 1/2 \cdot y + 1 \cdot z
\]

\[
y = 1 \cdot x + 0 \cdot y + 0 \cdot z
\]

\[
z = 0 \cdot x + 1/2 \cdot y + 0 \cdot z
\]
Non-Simplified PageRank (WWW)

\[ Pr(u) = \frac{1 - d}{k} + d \cdot \sum_{v \in B(u)} \frac{Pr(v)}{L(v)} \]

**Matrix Notation**

\[
Pr_{kx1} = M_{kxk} \times Pr_{kx1}
\]

\[
M_{ij} = \begin{cases} 
\frac{1 - d}{k} + \frac{d}{|B_j|}, & \text{if } i \in B_j \\
\frac{1 - d}{k}, & \text{else}
\end{cases}
\]

- (1-d) defines the probability to jump to a page, to which there is no link from the current page.
- Pr converges to the principal eigenvector of the transition matrix M.
Randomized HITS

- Random walk on HITS
- Odd time steps: update authority
- Even time steps: update hubs

\[
a^{(t+1)} = \epsilon \mathbf{1} + (1 - \epsilon) A_{\text{row}}^T h^{(t)} \\
h^{(t+1)} = \epsilon \mathbf{1} + (1 - \epsilon) A_{\text{col}} a^{(t+1)}
\]

- \( t \): a very large odd number, large enough that the random walk converged ➔ The authority weight of a page = the chance that the surfer visits that page on time step \( t \)
Stability of Algorithms

- Being stable to perturbations of the link structure.

- HITS: if the eigengap is big, insensitive to small perturbations; If it’s small there may be a small perturbation that can dramatically change its results.

- PageRank: if the perturbed/modified web pages did not have high overall PageRank, then the perturbed PageRank scores will not be far from the original.

- Randomized HITS: insensitive to small perturbations
Thank You!

Special thanks to Cem