LSI vs Link Analysis 
(A Survey)

C. D. Meyer and A. N. Langville

Department of Mathematics
North Carolina University
Raleigh, NC

1/23/2003
Outline

- Background & History
Outline

- Background & History
- Vector Space Approach
Outline

• Background & History

• Vector Space Approach

• Link Analysis Approach
Outline

- Background & History
- Vector Space Approach
- Link Analysis Approach
- Hybrid Approaches
Background

Goal

- Identify documents that best match users query
Background

Goal

- Identify documents that best match users query

Measures

- Recall = \( \frac{\text{relevant docs retrieved}}{\text{docs in collection}} \) (max # useful docs)

- Precision = \( \frac{\text{relevant docs retrieved}}{\text{docs retrieved}} \) (min # useless docs)
Background

Goal

• Identify documents that best match users query

Measures

• Recall = \frac{\text{#relevant docs retrieved}}{\text{#docs in collection}} \quad \text{(max # useful docs)}

• Precision = \frac{\text{#relevant docs retrieved}}{\text{#docs retrieved}} \quad \text{(min # useless docs)}

Do it \textit{FAST}!
SMART
(System for the Mechanical Analysis and Retrieval of Text)
SMART

(System for the Mechanical Analysis and Retrieval of Text)

Harvard 1962 – 1965

- IBM 7094 & IBM 360
SMART
(System for the Mechanical Analysis and Retrieval of Text)

Harvard 1962 – 1965
• IBM 7094 & IBM 360

Gerard Salton
• Implemented at Cornell (1965 – 1970)
SMART
(System for the Mechanical Analysis and Retrieval of Text)

Harvard 1962 – 1965
- IBM 7094 & IBM 360

Gerard Salton
- Implemented at Cornell (1965 – 1970)
- Based on matrix methods
Term–Document Matrix

Start With Dictionary of Terms

- Single words — or short phrases (e.g., landing gear)
Term–Document Matrix

Start With Dictionary of Terms
- Single words — or short phrases (e.g., landing gear)

Index Each Document (by human or by computer)
- Count $f_{ij} = \#$ times term $i$ appears in document $j$
Term–Document Matrix

Start With Dictionary of Terms
  • Single words — or short phrases (e.g., landing gear)

Index Each Document (by human or by computer)
  • Count $f_{ij} =$ # times term $i$ appears in document $j$

Term–Document Matrix

$$
\begin{pmatrix}
\text{TERM 1} \\
\text{TERM 2} \\
\vdots \\
\text{TERM } m
\end{pmatrix}
\begin{pmatrix}
f_{11} & f_{12} & \cdots & f_{1n} \\
f_{21} & f_{22} & \cdots & f_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
f_{m1} & f_{m2} & \cdots & f_{mn}
\end{pmatrix}
= A_{m \times n}
$$
Term–Document Matrix

Start With Dictionary of Terms
- Single words — or short phrases (e.g., *landing gear*)

Index Each Document (by human or by computer)
- Count $f_{ij} = \#$ times term $i$ appears in document $j$

Term–Document Matrix

$$
\begin{bmatrix}
    f_{11} & f_{12} & \cdots & f_{1n} \\
    f_{21} & f_{22} & \cdots & f_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    f_{m1} & f_{m2} & \cdots & f_{mn}
\end{bmatrix}
= A_{m \times n}
$$

Features
- $A \geq 0$
Term–Document Matrix

Start With Dictionary of Terms
• Single words — or short phrases (e.g., landing gear)

Index Each Document (by human or by computer)
• Count $f_{ij} = \#$ times term $i$ appears in document $j$

Term–Document Matrix

$\begin{pmatrix}
\text{Term 1} & \text{Doc 1} & f_{11} & f_{12} & \cdots & f_{1n} \\
\text{Term 2} & \text{Doc 2} & f_{21} & f_{22} & \cdots & f_{2n} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\text{Term m} & \text{Doc n} & f_{m1} & f_{m2} & \cdots & f_{mn}
\end{pmatrix} = A_{m \times n}$

Features
• $A \geq 0$
• $A$ can be really big
Term–Document Matrix

Start With Dictionary of Terms
- Single words — or short phrases (e.g., landing gear)

Index Each Document (by human or by computer)
- Count $f_{ij} = \#$ times term $i$ appears in document $j$

Term–Document Matrix

$$
\begin{pmatrix}
  f_{11} & f_{12} & \cdots & f_{1n} \\
  f_{21} & f_{22} & \cdots & f_{2n} \\
  \vdots  & \vdots  & \ddots & \vdots \\
  f_{m1} & f_{m2} & \cdots & f_{mn}
\end{pmatrix}
= A_{m \times n}
$$

Features
- $A \geq 0$
- $A$ can be really big
- $A$ is sparse — but otherwise unstructured
# Term–Document Matrix

## Start With Dictionary of Terms
- Single words — or short phrases (e.g., *landing gear*)

## Index Each Document (by human or by computer)
- Count $f_{ij} = \#$ times term $i$ appears in document $j$

## Term–Document Matrix

<table>
<thead>
<tr>
<th>Term 1 \ Doc 1</th>
<th>Term 1 \ Doc 2</th>
<th>\cdots</th>
<th>Term 1 \ Doc n</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{11}$</td>
<td>$f_{12}$</td>
<td>$\cdots$</td>
<td>$f_{1n}$</td>
</tr>
<tr>
<td>$f_{21}$</td>
<td>$f_{22}$</td>
<td>$\cdots$</td>
<td>$f_{2n}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\ddots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$f_{m1}$</td>
<td>$f_{m2}$</td>
<td>$\cdots$</td>
<td>$f_{mn}$</td>
</tr>
</tbody>
</table>

$= A_{m \times n}$

## Features
- $A \geq 0$
- $A$ can be really big
- $A$ is sparse — but otherwise unstructured
- $A$ contains a lot of uncertainty
Query Matching

Query Vector

- $q^T = (q_1, q_2, \ldots, q_m)$ where $q_i = \begin{cases} 
1 & \text{if Term } i \text{ is requested} \\
0 & \text{if not} 
\end{cases}$
Query Matching

Query Vector

- \( q^T = (q_1, q_2, \ldots, q_m) \) where \( q_i = \begin{cases} 1 & \text{if Term } i \text{ is requested} \\ 0 & \text{if not} \end{cases} \)

How Close is the Query to Each Document?
Query Matching

Query Vector

- \( q^T = (q_1, q_2, \ldots, q_m) \)  where  \( q_i = \begin{cases} 
1 & \text{if Term } i \text{ is requested} \\
0 & \text{if not}
\end{cases} \)

How Close is the Query to Each Document?

- i.e., how close is \( q \) to each column \( A_i \)?
Query Matching

Query Vector

- \( q^T = (q_1, q_2, \ldots, q_m) \) where \( q_i = \begin{cases} 1 & \text{if Term } i \text{ is requested} \\ 0 & \text{if not} \end{cases} \)

How Close is the Query to Each Document?

- i.e., how close is \( q \) to each column \( A_i \)?

\[ \|q - A_1\| < \|q - A_2\| \text{ but } \theta_2 < \theta_1 \]
Query Matching

Query Vector

- \( q^T = (q_1, q_2, \ldots, q_m) \) where \( q_i = \begin{cases} 1 & \text{if Term } i \text{ is requested} \\ 0 & \text{if not} \end{cases} \)

How Close is the Query to Each Document?

- i.e., how close is \( q \) to each column \( A_i \)?

\[
\| q - A_1 \| < \| q - A_2 \| \quad \text{but} \quad \theta_2 < \theta_1
\]

Use \( \delta_i = \cos \theta_i = \frac{q^T A_i}{\|q\| \|A_i\|} \)
Query Matching

Query Vector

- \( q^T = (q_1, q_2, \ldots, q_m) \) where \( q_i = \begin{cases} 1 & \text{if Term } i \text{ is requested} \\ 0 & \text{if not} \end{cases} \)

How Close is the Query to Each Document?

- i.e., how close is \( q \) to each column \( A_i \)?

\[
\|q - A_1\| < \|q - A_2\| \text{ but } \theta_2 < \theta_1
\]

Use \( \delta_i = \cos \theta_i = \frac{q^T A_i}{\|q\| \|A_i\|} \)

Rank documents by size of \( \delta_i \)
Query Matching

Query Vector

- $q^T = (q_1, q_2, \ldots, q_m)$ where $q_i = \begin{cases} 1 & \text{if Term } i \text{ is requested} \\ 0 & \text{if not} \end{cases}$

How Close is the Query to Each Document?

- i.e., how close is $q$ to each column $A_i$?

\[ \|q - A_1\| < \|q - A_2\| \text{ but } \theta_2 < \theta_1 \]

Use $\delta_i = \cos \theta_i = \frac{q^T A_i}{\|q\| \|A_i\|}$

Rank documents by size of $\delta_i$

Return Document $i$ to user when $\delta_i \geq tol$
Term Weighting

A Defect

- If the term *bank* occurs once in Doc 1 but twice in Doc 2, and if $\|A_1\| \approx \|A_2\|$, then a query containing only *bank* produces $\delta_2 \approx 2\delta_1$ (i.e., Doc 2 is rated twice as relevant as Doc 1).
Term Weighting

A Defect

• If the term \textit{bank} occurs once in Doc 1 but twice in Doc 2, and if $\|A_1\| \approx \|A_2\|$, then a query containing only \textit{bank} produces $\delta_2 \approx 2\delta_1$ (i.e., Doc 2 is rated twice as relevant as Doc 1).

To Compensate

• Set $a_{ij} = \log(1 + f_{ij})$ (other weights also possible)
Term Weighting

A Defect

- If the term *bank* occurs once in Doc 1 but twice in Doc 2, and if $\|A_1\| \approx \|A_2\|$, then a query containing only *bank* produces $\delta_2 \approx 2\delta_1$ (i.e., Doc 2 is rated twice as relevant as Doc 1).

To Compensate

- Set $a_{ij} = \log(1 + f_{ij})$ (other weights also possible)

Query Weights

- Terms *Boeing* and *airplanes* not equally important in queries
Term Weighting

A Defect

- If the term \textit{bank} occurs once in Doc 1 but twice in Doc 2, and if $\|A_1\| \approx \|A_2\|$, then a query containing only \textit{bank} produces $\delta_2 \approx 2\delta_1$ (i.e., Doc 2 is rated twice as relevant as Doc 1).

To Compensate

- Set $a_{ij} = \log(1 + f_{ij})$ (other weights also possible)

Query Weights

- Terms \textit{Boeing} and \textit{airplanes} not equally important in queries
- Importance of Term $i$ tends to be inversely proportional to $\nu_i = \#$ Docs containing Term $i$
Term Weighting

A Defect

- If the term \textit{bank} occurs once in Doc 1 but twice in Doc 2, and if $\|A_1\| \approx \|A_2\|$, then a query containing only \textit{bank} produces $\delta_2 \approx 2\delta_1$ (i.e., Doc 2 is rated twice as relevant as Doc 1).

To Compensate

- Set $a_{ij} = \log(1 + f_{ij})$ (other weights also possible)

Query Weights

- Terms \textit{Boeing} and \textit{airplanes} not equally important in queries
- Importance of Term $i$ tends to be inversely proportional to $\nu_i = \#$ Docs containing Term $i$

To Compensate

- Set $q_i = \begin{cases} 
\log(n/\nu_i) & \text{if } \nu_i \neq 0 \\
0 & \text{if } \nu_i = 0 
\end{cases}$ (other weights also possible)
Uncertainties in A
Uncertainties in A

Ambiguity in Vocabulary
Uncertainties in A

Ambiguity in Vocabulary

- e.g., A *plane* could be \(\cdots\)
Uncertainties in A

Ambiguity in Vocabulary

- e.g., A *plane* could be ⋅⋅⋅

  - A flat geometrical object
Uncertainties in A

Ambiguity in Vocabulary

- e.g., A *plane* could be ⋯
  - A flat geometrical object
  - A woodworking tool
Uncertainties in A

Ambiguity in Vocabulary

• e.g., A plane could be ...
  — A flat geometrical object
  — A woodworking tool
  — A Boeing product
Uncertainties in A

Ambiguity in Vocabulary

- e.g., A *plane* could be ⋅⋅⋅
  - A flat geometrical object
  - A woodworking tool
  - A Boeing product

Variation in Writing Style

- No two authors write the same way
Uncertainties in A

Ambiguity in Vocabulary

- e.g., A *plane* could be
  - A flat geometrical object
  - A woodworking tool
  - A Boeing product

Variation in Writing Style

- No two authors write the same way
  - One author may write *car* and *laptop*
Uncertainties in A

Ambiguity in Vocabulary

- e.g., A *plane* could be
  - A flat geometrical object
  - A woodworking tool
  - A Boeing product

Variation in Writing Style

- No two authors write the same way
  - One author may write *car* and *laptop*
  - Another author may write *automobile* and *portable*
Uncertainties in A

Ambiguity in Vocabulary
  • e.g., A *plane* could be
    — A flat geometrical object
    — A woodworking tool
    — A Boeing product

Variation in Writing Style
  • No two authors write the same way
    — One author may write *car* and *laptop*
    — Another author may write *automobile* and *portable*

Variation in Indexing Conventions
  • No two people index documents the same way
  • Computer indexing is inexact and can be unpredictable
Theory vs Practice

In Theory — it’s easy
Theory vs Practice

In Theory — it’s easy

- Weight terms and normalize cols — Make $\|A_i\| = 1$
In Theory — it’s easy

- Weight terms and normalize cols — Make $\|A_i\| = 1$
- For each new query, weight and normalize — Make $\|q\| = 1$
Theory vs Practice

In Theory — it's easy

- Weight terms and normalize cols — Make $\|A_i\| = 1$
- For each new query, weight and normalize — Make $\|q\| = 1$
- Compute $\delta_i = \cos \theta_i = (q^T A)_i$ to return the most relevant docs
Theory vs Practice

In Theory — it’s easy

- Weight terms and normalize cols — Make $\|A_i\| = 1$
- For each new query, weight and normalize — Make $\|q\| = 1$
- Compute $\delta_i = \cos \theta_i = (q^T A)_i$ to return the most relevant docs

In Practice — it’s not so easy
Theory vs Practice

In Theory — it’s easy

- Weight terms and normalize cols — Make $\|A_i\| = 1$
- For each new query, weight and normalize — Make $\|q\| = 1$
- Compute $\delta_i = \cos \theta_i = (q^T A)_i$ to return the most relevant docs

In Practice — it’s not so easy

- Suppose query = *gas*
Theory vs Practice

In Theory — it’s easy

- Weight terms and normalize cols — Make $\|A_i\| = 1$
- For each new query, weight and normalize — Make $\|q\| = 1$
- Compute $\delta_i = \cos \theta_i = (q^T A)_i$ to return the most relevant docs

In Practice — it’s not so easy

- Suppose query = $\text{gas}$
- $D_1$ indexed by $\text{gas, car, tire}$
Theory vs Practice

In Theory — it’s easy

- Weight terms and normalize cols — Make $\|A_i\| = 1$
- For each new query, weight and normalize — Make $\|q\| = 1$
- Compute $\delta_i = \cos \theta_i = (q^T A)_i$ to return the most relevant docs

In Practice — it’s not so easy

- Suppose query = gas
- $D_1$ indexed by gas, car, tire
Theory vs Practice

In Theory — it’s easy

- Weight terms and normalize cols — Make $\|A_i\| = 1$
- For each new query, weight and normalize — Make $\|q\| = 1$
- Compute $\delta_i = \cos \theta_i = (q^T A)_i$ to return the most relevant docs

In Practice — it’s not so easy

- Suppose query = gas
- $D_1$ indexed by gas, car, tire (found)
- $D_2$ indexed automobile, fuel, and tire
Theory vs Practice

In Theory — it’s easy

- Weight terms and normalize cols — Make $\|A_i\| = 1$
- For each new query, weight and normalize — Make $\|q\| = 1$
- Compute $\delta_i = \cos \theta_i = (q^T A)_i$ to return the most relevant docs

In Practice — it’s not so easy

- Suppose query = gas
- $D_1$ indexed by gas, car, tire (found)
- $D_2$ indexed automobile, fuel, and tire (missed)
Theory vs Practice

In Theory — it’s easy

- Weight terms and normalize cols — Make $\|A_i\| = 1$
- For each new query, weight and normalize — Make $\|q\| = 1$
- Compute $\delta_i = \cos \theta_i = (q^T A)_i$ to return the most relevant docs

In Practice — it’s not so easy

- Suppose query = gas
- $D_1$ indexed by gas, car, tire
- $D_2$ indexed automobile, fuel, and tire

Somehow Reveal Latent Connections

- Find $D_2$ by making the connection through tire
Theory vs Practice

In Theory — it’s easy

- Weight terms and normalize cols — Make $\|A_i\| = 1$
- For each new query, weight and normalize — Make $\|q\| = 1$
- Compute $\delta_i = \cos \theta_i = (q^T A)_i$ to return the most relevant docs

In Practice — it’s not so easy

- Suppose query = gas
- $D_1$ indexed by gas, car, tire (found)
- $D_2$ indexed automobile, fuel, and tire (missed)

Somehow Reveal Latent Connections

- Find $D_2$ by making the connection through tire
- Do it FAST!
Theory vs Practice

In Theory — it’s easy

- Weight terms and normalize cols — Make $\|A_i\| = 1$
- For each new query, weight and normalize — Make $\|q\| = 1$
- Compute $\delta_i = \cos \theta_i = (q^T A)_i$ to return the most relevant docs

In Practice — it’s not so easy

- Suppose query = gas
- $D_1$ indexed by gas, car, tire
- $D_2$ indexed automobile, fuel, and tire

Somehow Reveal Latent Connections

- Find $D_2$ by making the connection through tire
- Do it FAST!
  - Data compression
Contaminated Data (not text data)

\[
\mathbf{x} = \begin{bmatrix}
  x_0 \\
  x_1 \\
  x_2 \\
  \vdots \\
  x_{510} \\
  x_{511}
\end{bmatrix}
\]
Contaminated Data (not text data)

$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{510} \\ x_{511} \end{bmatrix}$
Contaminated Data  (not text data)

\[ x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{510} \\ x_{511} \end{bmatrix} \]

Goal
- Reveal hidden patterns
Contaminated Data (not text data)

\[ \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{510} \\ x_{511} \end{bmatrix} \]

Goal
- Reveal hidden patterns
- Compress the data
Change Of Coordinates

New Basis \( \mathcal{B} = \{ W_0, W_1, \ldots, W_{n-1} \} \)
New Basis  $\mathcal{B} = \{W_0, W_1, \ldots, W_{n-1}\}$

- Find coordinates of $x$ with respect to $\mathcal{B}$
Change of Coordinates

New Basis \( \mathcal{B} = \{W_0, W_1, \ldots, W_{n-1}\} \)

- Find coordinates of \( x \) with respect to \( \mathcal{B} \)
  - Find \( y_k \) so that \( x = \sum y_k W_k \) (Fourier expansion if \( \mathcal{B} \) o.n.)
Change Of Coordinates

New Basis \( \mathcal{B} = \{W_0, W_1, \ldots, W_{n-1}\} \)

- Find coordinates of \( x \) with respect to \( \mathcal{B} \)
  - Find \( y_k \) so that \( x = \sum y_k W_k \) (Fourier expansion if \( \mathcal{B} \) o.n.)
  - \( y_k = \langle W_k | x \rangle = \text{amount of } x \text{ in direction of } W_k \) (if \( \mathcal{B} \) o.n.)
Change Of Coordinates

New Basis \[ \mathcal{B} = \{W_0, W_1, \ldots, W_{n-1}\} \]

- Find coordinates of \( x \) with respect to \( \mathcal{B} \)
  - Find \( y_k \) so that \( x = \sum y_k W_k \) (Fourier expansion if \( \mathcal{B} \) o.n.)
  - \( y_k = \langle W_k | x \rangle = \text{amount of } x \text{ in direction of } W_k \) (if \( \mathcal{B} \) o.n.)
  - \( x = W y \) where \( W = (W_0 | W_1 | \cdots | W_{n-1}) \)
Change Of Coordinates

New Basis \( \mathcal{B} = \{ W_0, W_1, \ldots, W_{n-1} \} \)

- Find coordinates of \( x \) with respect to \( \mathcal{B} \)
  
  - Find \( y_k \) so that \( x = \sum y_k W_k \) (Fourier expansion if \( \mathcal{B} \) o.n.)
  
  - \( y_k = \langle W_k | x \rangle = \text{amount of } x \text{ in direction of } W_k \) (if \( \mathcal{B} \) o.n.)
  
  - \( x = W y \) where \( W = (W_0 | W_1 | \cdots | W_{n-1}) \)
  
  - \( y = W^{-1} x \) (\( y = W^* x \) if \( \mathcal{B} \) o.n.)
Change Of Coordinates

New Basis \( \mathcal{B} = \{W_0, W_1, \ldots, W_{n-1}\} \)

- Find coordinates of \( x \) with respect to \( \mathcal{B} \)
  - Find \( y_k \) so that \( x = \sum y_k W_k \) (Fourier expansion if \( \mathcal{B} \) o.n.)
  - \( y_k = \langle W_k | x \rangle = \text{amount of } x \text{ in direction of } W_k \) (if \( \mathcal{B} \) o.n.)
  - \( x = W y \) where \( W = (W_0 \mid W_1 \mid \cdots \mid W_{n-1}) \)
  - \( y = W^{-1} x \) (\( y = W^*x \) if \( \mathcal{B} \) o.n.)

Oscillatory
Change Of Coordinates

New Basis $\mathcal{B} = \{W_0, W_1, \ldots, W_{n-1}\}$

- Find coordinates of $\mathbf{x}$ with respect to $\mathcal{B}$
  - Find $y_k$ so that $\mathbf{x} = \sum y_k W_k$ (Fourier expansion if $\mathcal{B}$ o.n.)
  - $y_k = \langle W_k | \mathbf{x} \rangle = $ amount of $\mathbf{x}$ in direction of $W_k$ (if $\mathcal{B}$ o.n.)
  - $\mathbf{x} = \mathbf{W} \mathbf{y}$ where $\mathbf{W} = (W_0 \ | \ W_1 \ | \cdots \ | \ W_{n-1})$
  - $\mathbf{y} = \mathbf{W}^{-1} \mathbf{x}$

Oscillatory

- $\mathbf{W} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{n-2} & \cdots & \omega \end{bmatrix}_{n\times n}$
  - $\omega = e^{2\pi i/n}$
\textbf{Change Of Coordinates}

\textbf{New Basis} \quad \mathcal{B} = \{W_0, W_1, \ldots, W_{n-1}\}

- Find coordinates of \( \mathbf{x} \) with respect to \( \mathcal{B} \)
  - Find \( y_k \) so that \( \mathbf{x} = \sum y_k W_k \) (Fourier expansion if \( \mathcal{B} \) o.n.)
  - \( y_k = \langle W_k | x \rangle = \text{amount of } \mathbf{x} \text{ in direction of } W_k \) (if \( \mathcal{B} \) o.n.)
  - \( \mathbf{x} = \mathbf{W} \mathbf{y} \) where \( \mathbf{W} = (W_0 | W_1 | \cdots | W_{n-1}) \)
  - \( \mathbf{y} = \mathbf{W}^{-1} \mathbf{x} \)

\textbf{Oscillatory}

\[ \mathbf{W} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{n-2} & \cdots & \omega \end{bmatrix} \]

\[ \omega = e^{2\pi i/n}, \quad W_k = \frac{e^{2\pi ikt}}{2} \quad t = 0, 1/n, 2/n, \ldots \]
Change Of Coordinates

New Basis \( \mathcal{B} = \{W_0, W_1, \ldots, W_{n-1}\} \)

- Find coordinates of \( x \) with respect to \( \mathcal{B} \)
  - Find \( y_k \) so that \( x = \sum y_k W_k \) (Fourier expansion if \( \mathcal{B} \) o.n.)
  - \( y_k = \langle W_k | x \rangle = \text{amount of } x \text{ in direction of } W_k \) (if \( \mathcal{B} \) o.n.)
  - \( x = W y \) where \( W = (W_0 \mid W_1 \mid \cdots \mid W_{n-1}) \)
  - \( y = W^{-1} x \) (\( y = W^* x \) if \( \mathcal{B} \) o.n.)

Oscillatory

- \( W = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{n-2} & \cdots & \omega \end{bmatrix}_{n \times n} \)
  - \( \omega = e^{2\pi i/n}, \quad W_k = \frac{e^{2\pi i k t}}{2} \quad t = 0, 1/n, 2/n, \ldots \)

- \( W_k + W_{n-k} = \cos 2\pi k t \)
Change Of Coordinates

New Basis $\mathcal{B} = \{W_0, W_1, \ldots, W_{n-1}\}$

- Find coordinates of $\mathbf{x}$ with respect to $\mathcal{B}$
  
  — Find $y_k$ so that $\mathbf{x} = \sum y_k W_k$ (Fourier expansion if $\mathcal{B}$ o.n.)
  
  — $y_k = \langle W_k | \mathbf{x} \rangle = \text{amount of } \mathbf{x} \text{ in direction of } W_k$ (if $\mathcal{B}$ o.n.)
  
  — $\mathbf{x} = W\mathbf{y}$ where $W = (W_0 | W_1 | \cdots | W_{n-1})$
  
  — $\mathbf{y} = W^{-1}\mathbf{x}$

Oscillatory

$W = \frac{1}{2} \begin{bmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\
1 & \omega^2 & \omega^4 & \cdots & \omega^{n-2} \\
: & : & : & \ddots & : \\
1 & \omega^{n-1} & \omega^{n-2} & \cdots & \omega \\
\end{bmatrix}_{n \times n} \quad \omega = e^{2\pi i/n}, \quad W_k = \frac{e^{2\pi i k t}}{2}$

$t = 0, 1/n, 2/n, \ldots$

- $W_k + W_{n-k} = \cos 2\pi k t$
- $W_k - W_{n-k} = i \sin 2\pi k t$
Making The Change
Recall

- \( x = \sum y_k W_k = Wy \)
Making The Change

Recall

- \( x = \sum y_k W_k = Wy \)
- \( y = W^{-1} x \)
Making The Change

Recall

- \( x = \sum y_k W_k = Wy \)
- \( y = W^{-1}x \)

\( W^{-1} = (4/n)\overline{W} = \text{Discrete Fourier Transform} \)
Making The Change

Recall

- \( x = \sum y_k W_k = Wy \)

- \( y = W^{-1}x \)

\( W^{-1} = (4/n)\overline{W} = \text{Discrete Fourier Transform} \)

\[
\begin{bmatrix}
    y_0 \\
    y_1 \\
    y_2 \\
    \vdots \\
    y_{n-1}
\end{bmatrix} = \frac{2}{n} \begin{bmatrix}
    1 & 1 & 1 & \cdots & 1 \\
    1 & \xi & \xi^2 & \cdots & \xi^{n-1} \\
    1 & \xi^2 & \xi^4 & \cdots & \xi^{n-2} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    1 & \xi^{n-1} & \xi^{n-2} & \cdots & \xi
\end{bmatrix} \begin{bmatrix}
    x_0 \\
    x_1 \\
    x_2 \\
    \vdots \\
    x_{n-1}
\end{bmatrix}
\]

\( \xi = e^{-2\pi i/n} = \bar{\omega} \)
Only 4 are significant: $y_{80} = y_{432} = 1$
Only 4 are significant: \( y_{80} = y_{432} = 1 \) and \( y_{50} = -2i = -y_{462} \)
• Only 4 are significant: \( y_{80} = y_{432} = 1 \) and \( y_{50} = -2i = -y_{462} \)

• \( \mathbf{x} = \sum y_k W_k = 1W_{80} + 1W_{432} - 2iW_{50} + 2iW_{462} + \sum \varepsilon_j W_j \)
The New Coordinates

- Only 4 are significant: \( y_{80} = y_{432} = 1 \) and \( y_{50} = -2i = -y_{462} \)

- \( x = \sum y_k W_k = 1W_{80} + 1W_{432} - 2iW_{50} + 2iW_{462} + \sum \varepsilon_j W_j \)

- Small components (noise) are nondirectional
Drop Small Coordinates

\[ \mathbf{x} = \sum y_k W_k = 1W_{80} + 1W_{432} - 2iW_{50} + 2iW_{462} + \sum \varepsilon_j W_j \]
Drop Small Coordinates

\[ x = \sum y_k W_k = 1W_{80} + 1W_{432} - 2iW_{50} + 2iW_{462} + \sum \varepsilon_j W_j \]

\[ \tilde{x} = (W_{80} + W_{432}) - 2i(W_{50} - W_{462}) \]
Drop Small Coordinates

- \( \mathbf{x} = \sum y_k W_k = 1W_{80} + 1W_{432} - 2iW_{50} + 2iW_{462} + \sum \varepsilon_j W_j \)

- \( \tilde{\mathbf{x}} = (W_{80} + W_{432}) - 2i(W_{50} - W_{462}) \)

- \( n = 512 \)

- \( \tilde{\mathbf{x}} = (W_{80} + W_{n-80}) - 2i(W_{50} - W_{n-50}) \)
Drop Small Coordinates

- $\mathbf{x} = \sum y_k W_k = 1W_{80} + 1W_{432} - 2iW_{50} + 2iW_{462} + \sum \varepsilon_j W_j$

- $\mathbf{\tilde{x}} = (W_{80} + W_{432}) - 2i(W_{50} - W_{462})$

- $n = 512$

- $\mathbf{\tilde{x}} = (W_{80} + W_{n-80}) - 2i(W_{50} - W_{n-50})$

Compressed (512 → 4)
Drop Small Coordinates

- \( \mathbf{x} = \sum y_k W_k = 1W_{80} + 1W_{432} - 2iW_{50} + 2iW_{462} + \sum \varepsilon_j W_j \)

- \( \tilde{\mathbf{x}} = (W_{80} + W_{432}) - 2i(W_{50} - W_{462}) \)

- \( n = 512 \)

- \( \tilde{\mathbf{x}} = (W_{80} + W_{n-80}) - 2i(W_{50} - W_{n-50}) \) — Compressed (512 → 4)

- \( W_k + W_{n-k} = \cos 2\pi k \mathbf{t} \)

- \( W_k - W_{n-k} = i \sin 2\pi k \mathbf{t} \)
Drop Small Coordinates

• \( \mathbf{x} = \sum y_k W_k = W_{80} + W_{432} - 2iW_{50} + 2iW_{462} + \sum \varepsilon_j W_j \)

• \( \tilde{\mathbf{x}} = (W_{80} + W_{432}) - 2i(W_{50} - W_{462}) \)

• \( n = 512 \)

• \( \tilde{\mathbf{x}} = (W_{80} + W_{n-80}) - 2i(W_{50} - W_{n-50}) \)  
  Compressed \((512 \rightarrow 4)\)

  \( W_k + W_{n-k} = \cos 2\pi k t \)

  \( W_k - W_{n-k} = i \sin 2\pi k t \)

• \( \tilde{\mathbf{x}} = \cos 2\pi 80 t + 2 \sin 2\pi 50 t \)
Drop Small Coordinates

- \( \mathbf{x} = \sum y_k W_k = 1W_{80} + 1W_{432} - 2iW_{50} + 2iW_{462} + \sum \varepsilon_j W_j \)
- \( \tilde{\mathbf{x}} = (W_{80} + W_{432}) - 2i(W_{50} - W_{462}) \)
- \( n = 512 \)
- \( \tilde{\mathbf{x}} = (W_{80} + W_{n-80}) - 2i(W_{50} - W_{n-50}) \)

Compressed \((512 \rightarrow 4)\)

- \( W_k + W_{n-k} = \cos 2\pi k t \)
- \( W_k - W_{n-k} = i \sin 2\pi k t \)

Cleaned

- \( \tilde{\mathbf{x}} = \cos 2\pi 80t + 2 \sin 2\pi 50t \)
Drop Small Coordinates

- \[ \mathbf{x} = \sum y_k W_k = 1W_{80} + 1W_{432} - 2iW_{50} + 2iW_{462} + \sum \varepsilon_j W_j \]

- \[ \tilde{\mathbf{x}} = (W_{80} + W_{432}) - 2i(W_{50} - W_{462}) \]

- \( n = 512 \)

- \[ \tilde{\mathbf{x}} = (W_{80} + W_{n-80}) - 2i(W_{50} - W_{n-50}) \]
  
  - \( W_k + W_{n-k} = \cos 2\pi k t \)
  
  - \( W_k - W_{n-k} = i \sin 2\pi k t \)

- \[ \tilde{\mathbf{x}} = \cos 2\pi 80t + 2 \sin 2\pi 50t \]

- \( \mathbf{x} = \cos 2\pi 80t + 2 \sin 2\pi 50t + \text{noise} \)
Original Data

\[ \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{510} \\ x_{511} \end{bmatrix} \]
Cleaned & Compressed Data

\[ \tilde{x} = x - \text{noise} = (W_{80} + W_{432}) - 2i(W_{50} - W_{462}) \]

\[ \cos 2\pi 80t + 2 \sin 2\pi 50t \]
The DFT Game

Matrix–Vector Product

\[
y = \frac{2}{n} \begin{bmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & \zeta & \zeta^2 & \cdots & \zeta^{n-1} \\
1 & \zeta^2 & \zeta^4 & \cdots & \zeta^{n-2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \zeta^{n-1} & \zeta^{n-2} & \cdots & \zeta \\
\end{bmatrix} \begin{bmatrix}
x_0 \\
x_1 \\
x_2 \\
\vdots \\
x_{n-1} \\
\end{bmatrix}
\]

\[
\zeta = e^{-2\pi i/n}
\]
The DFT Game

Matrix–Vector Product

\[ y = \frac{2}{n} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \xi & \xi^2 & \cdots & \xi^{n-1} \\ 1 & \xi^2 & \xi^4 & \cdots & \xi^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi^{n-1} & \xi^{n-2} & \cdots & \xi \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix} \]

\[ \xi = e^{-\frac{2\pi i}{n}} \]

Simple in Theory, But \cdots
The DFT Game

Matrix–Vector Product

\[ y = \frac{2}{n} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \xi & \xi^2 & \cdots & \xi^{n-1} \\ 1 & \xi^2 & \xi^4 & \cdots & \xi^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi^{n-1} & \xi^{n-2} & \cdots & \xi \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix} \]

\[ \xi = e^{-2\pi i / n} \]

Simple in Theory, But \cdots

- Must do it \textit{FAST}!
The DFT Game

Matrix–Vector Product

\[ y = \frac{2}{n} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \xi & \xi^2 & \cdots & \xi^{n-1} \\ 1 & \xi^2 & \xi^4 & \cdots & \xi^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi^{n-1} & \xi^{n-2} & \cdots & \xi \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix} \]

\[ \xi = e^{-2\pi i / n} \]

Simple in Theory, But \cdots

- Must do it \textit{FAST}!

Need For Speed \implies Matrix Factorizations \implies FFT
The DFT Game

Matrix–Vector Product

\[
y = \frac{2}{n} \begin{bmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & \xi & \xi^2 & \cdots & \xi^{n-1} \\
1 & \xi^2 & \xi^4 & \cdots & \xi^{n-2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \xi^{n-1} & \xi^{n-2} & \cdots & \xi \\
\end{bmatrix} \begin{bmatrix}
x_0 \\
x_1 \\
x_2 \\
\vdots \\
x_{n-1} \\
\end{bmatrix} = e^{-2\pi i/n}
\]

Simple in Theory, But …

- Must do it \textit{FAST}!

Need For Speed \implies Matrix Factorizations \implies FFT

- \( F_n = B_n (I_2 \otimes F_{n/2}) P_n \)
- \( B_n = \begin{bmatrix}
I_{n/2} & D_{n/2} \\
I_{n/2} & -D_{n/2} \\
\end{bmatrix} \)
- \( D_{n/2} = \begin{bmatrix}
1 & \xi & \xi^2 & \cdots \\
\end{bmatrix} \)
The DFT Game

Matrix–Vector Product

\[ y = \frac{2}{n} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \xi & \xi^2 & \cdots & \xi^{n-1} \\ 1 & \xi^2 & \xi^4 & \cdots & \xi^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi^{n-1} & \xi^{n-2} & \cdots & \xi \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix} \]

\[ \xi = e^{-2\pi i/n} \]

Simple in Theory, But ⋅⋅⋅

- Must do it *FAST*!

Need For Speed ⟷ Matrix Factorizations ⟷ FFT

- \( F_n = B_n (I_2 \otimes F_{n/2}) P_n \)
  - \( B_n = \begin{bmatrix} I_{n/2} & D_{n/2} \\ I_{n/2} & -D_{n/2} \end{bmatrix} \)
  - \( D_{n/2} = \begin{bmatrix} 1 & \xi & \xi^2 & \cdots \end{bmatrix} \)
- FFT changes \( n^2 \) flop requirement into \( (n/2) \log_2 n \)
The DFT Game

Matrix–Vector Product

\[
y = \frac{2}{n} \begin{bmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & \xi & \xi^2 & \cdots & \xi^{n-1} \\
1 & \xi^2 & \xi^4 & \cdots & \xi^{n-2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \xi^{n-1} & \xi^{n-2} & \cdots & \xi \\
\end{bmatrix}
\begin{bmatrix}
x_0 \\
x_1 \\
x_2 \\
\vdots \\
x_{n-1} \\
\end{bmatrix}
\]

\[
\xi = e^{-2\pi i/n}
\]

Simple in Theory, But \cdots

\bullet Must do it FAST!

Need For Speed \implies Matrix Factorizations \implies FFT

\bullet \quad F_n = B_n (I_2 \otimes F_{n/2}) P_n \\
\quad B_n = \begin{bmatrix}
I_{n/2} & D_{n/2} \\
I_{n/2} & -D_{n/2} \\
\end{bmatrix} \\
\quad D_{n/2} = \begin{bmatrix}
1 & \xi & \xi^2 & \cdots \\
\end{bmatrix}

\bullet FFT changes \(n^2\) flop requirement into \((n/2) \log_2 n\)

“The most valuable numerical algorithm in our lifetime.”
Back To IR

Almost the Same Problem

- Reveal hidden patterns & evaluate $q^T A$ fast
Back To IR

Almost the Same Problem

- Reveal hidden patterns & evaluate $q^T A$ fast  
  (clean & compress)
Back To IR

Almost the Same Problem

- Reveal hidden patterns & evaluate $q^T A$ fast (clean & compress)

Data is Now the Term-Doc Matrix in Standard Coordinates

- $A = \sum_{i,j} a_{ij} E_{ij}$
- $E_{ij} = e_i e_j^T$
Back To IR

Almost the Same Problem

- Reveal hidden patterns & evaluate $q^T A$ fast (clean & compress)

Data is Now the Term-Doc Matrix in Standard Coordinates

- $A = \sum_{i,j} a_{ij} E_{ij}$
- $E_{ij} = e_i e_j^T$

Change Basis to $B = \{Z_1, Z_2, \ldots\}$ That Can Squeeze & Clean

- $A = \sum \sigma_i Z_i$ (Fourier Expansion)
Back To IR

Almost the Same Problem

- Reveal hidden patterns & evaluate $q^T A$ fast  
  (clean & compress)

Data is Now the Term-Doc Matrix in Standard Coordinates

- $A = \sum_{i,j} a_{ij} E_{ij}$  
  $E_{ij} = e_i e_j^T$

Change Basis to $B = \{Z_1, Z_2, \ldots\}$ That Can Squeeze & Clean

- $A = \sum \sigma_i Z_i$  
  (Fourier Expansion)

- $B$ o.n. $\Rightarrow \sigma_i = \langle Z_i | A \rangle = \text{amount of } A \text{ in direction of } Z_i$
Back To IR

Almost the Same Problem

- Reveal hidden patterns & evaluate $q^T A$ fast (clean & compress)

Data is Now the Term-Doc Matrix in Standard Coordinates

- $A = \sum_{i,j} a_{ij} E_{ij}$
  - $E_{ij} = e_i e_j^T$

Change Basis to $B = \{Z_1, Z_2, \ldots\}$ That Can Squeeze & Clean

- $A = \sum \sigma_i Z_i$
  - $B$ o.n. $\Rightarrow \sigma_i = \langle Z_i | A \rangle$ = amount of $A$ in direction of $Z_i$
  - (Fourier Expansion)

Matrix Factorizations: $A = URV^T = \sum r_{ij} u_i v^T_j = \sum r_{ij} Z_{ij}$
Back To IR

Almost the Same Problem

- Reveal hidden patterns & evaluate $q^T A$ fast (clean & compress)

Data is Now the Term-Doc Matrix in Standard Coordinates

- $A = \sum_{i,j} a_{ij} E_{ij}, \quad E_{ij} = e_i e_j^T$

Change Basis to $B = \{ Z_1, Z_2, \ldots \}$ That Can Squeeze & Clean

- $A = \sum \sigma_i Z_i$
  
  - $B$ o.n. $\Rightarrow \sigma_i = \langle Z_i | A \rangle = \text{amount of } A \text{ in direction of } Z_i$

Matrix Factorizations: $A = URV^T = \sum r_{ij} u_i v_{ij}^T = \sum r_{ij} Z_{ij}$

- Represent data with as few directions $Z_i$ as possible
Back To IR

Almost the Same Problem

- Reveal hidden patterns & evaluate $q^T A$ fast  
  (clean & compress)

Data is Now the Term-Doc Matrix in Standard Coordinates

- $A = \sum_{i,j} a_{ij} E_{ij}$  
  $E_{ij} = e_i e_j^T$

Change Basis to $B = \{Z_1, Z_2, \ldots \}$ That Can Squeeze & Clean

- $A = \sum \sigma_i Z_i$  
  (Fourier Expansion)

- $B \text{ o.n. } \Rightarrow \sigma_i = \langle Z_i | A \rangle = \text{amount of } A \text{ in direction of } Z_i$

Matrix Factorizations: $A = URV^T = \sum r_{ij} u_i v_j^T = \sum r_{ij} Z_{ij}$

- Represent data with as few directions $Z_i$ as possible

- SVD $\Rightarrow R = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}$  
  $\Rightarrow A = \sum_{i=1}^r \sigma_i Z_i$,  
  $\langle Z_i | Z_j \rangle = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$
Same As Before

Assume Nondirectional Uncertainty
Assume Nondirectional Uncertainty

- Drop small $\sigma_i$'s — replace $A$ with $\tilde{A} = \sum_{i=1}^{k} \sigma_i Z_i$
Same As Before

Assume Nondirectional Uncertainty

- Drop small $\sigma_i$'s — replace $A$ with $\tilde{A} = \sum_{i=1}^{k} \sigma_i Z_i$
- Lose only small part of relevance
Same As Before

Assume Nondirectional Uncertainty

- Drop small $\sigma_i$’s — replace $A$ with $\tilde{A} = \sum_{i=1}^{k} \sigma_i Z_i$
- Lose only small part of relevance
- Lose larger proportion of uncertainty
Same As Before

Assume Nondirectional Uncertainty

- Drop small $\sigma_i$’s — replace $A$ with $\tilde{A} = \sum_{i=1}^{k} \sigma_i Z_i$
- Lose only small part of relevance
- Lose larger proportion of uncertainty

New Query Matching Strategy
Same As Before

Assume Nondirectional Uncertainty

- Drop small $\sigma_i$’s — replace $A$ with $\tilde{A} = \sum_{i=1}^{k} \sigma_i Z_i$
- Lose only small part of relevance
- Lose larger proportion of uncertainty

New Query Matching Strategy

- Normalize
  
  \[
  q \leftarrow q / \|q\| \]

Same As Before

Assume Nondirectional Uncertainty

- Drop small $\sigma_i$'s — replace $A$ with $\tilde{A} = \sum_{i=1}^{k} \sigma_i Z_i$
- Lose only small part of relevance
- Lose larger proportion of uncertainty

New Query Matching Strategy

- Normalize
  - $q \leftarrow q/\|q\|$
  - $\tilde{A} \leftarrow \sum_{i=1}^{k} \sigma_i u_i \tilde{v}_i^T D = \sum_{i=1}^{k} \sigma_i u_i \tilde{v}_i^T$
Same As Before

Assume Nondirectional Uncertainty

- Drop small \( \sigma_i \)'s — replace \( A \) with \( \tilde{A} = \sum_{i=1}^{k} \sigma_i Z_i \)
- Lose only small part of relevance
- Lose larger proportion of uncertainty

New Query Matching Strategy

- Normalize
  - \( q \leftarrow \frac{q}{\|q\|} \)
  - \( \tilde{A} \leftarrow \sum_{i=1}^{k} \sigma_i u_i v_i^T D = \sum_{i=1}^{k} \sigma_i u_i \tilde{v}_i^T \)
- Compare query to each document
  - \( q^T \tilde{A} = \sum_{i=1}^{k} \sigma_i (q^T u_i) \tilde{v}_i^T = (\delta_1, \delta_2, \ldots, \delta_n) \)
Pros & Cons

Advantages

- Compression
  - $A$ replaced with a few singular values & vectors (but dense)
Pros & Cons

Advantages

- Compression
  - $A$ replaced with a few singular values & vectors (but dense)
  - They are determined & normalized only once
Pros & Cons

Advantages

- Compression
  - A replaced with a few sing values & vectors (but dense)
  - They are determined & normalized only once
- SPEED!
Pros & Cons

Advantages

• Compression
  — A replaced with a few sing values & vectors (but dense)
  — They are determined & normalized only once

• SPEED!
  — Each query requires only a few inner products

\[ q^T \tilde{A}_{m \times n} = \sum_{i=1}^{k} \sigma_i (q^T u_i) \tilde{v}_i^T \]
Pros & Cons

Advantages

- Compression
  - $A$ replaced with a few sing values & vectors (but dense)
  - They are determined & normalized only once
- *SPEED!*
  - Each query requires only a few inner products
    \[
    q^T \tilde{A}_{m \times n} = \sum_{i=1}^{k} \sigma_i (q^T u_i) \tilde{v}_i^T
    \]
- Latent semantic associations are made
  - Relevant docs not found by direct matching show up
Pros & Cons

Advantages

- **Compression**
  - $A$ replaced with a few sing values & vectors (but dense)
  - They are determined & normalized only once

- **SPEED!**
  - Each query requires only a few inner products

\[
q^T \tilde{A}_{m \times n} = \sum_{i=1}^{k} \sigma_i (q^T u_i) \tilde{v}_i^T
\]

- Latent semantic associations are made
  - Relevant docs not found by direct matching show up
  - *Latent Semantic Indexing* (LSI)
Pros & Cons

Advantages

- Compression
  - \( A \) replaced with a few singular values & vectors (but dense)
  - They are determined & normalized only once
- \textit{SPEED!}
  - Each query requires only a few inner products
    \[
    q^T \tilde{A}_{m \times n} = \sum_{i=1}^{k} \sigma_i (q^T u_i) \tilde{v}_i^T
    \]
- Latent semantic associations are made
  - Relevant docs not found by direct matching show up
  - \textit{Latent Semantic Indexing} (LSI)

Disadvantages
Pros & Cons

Advantages

• Compression
  — $\mathbf{A}$ replaced with a few singular values & vectors (but dense)
  — They are determined & normalized only once

• SPEED!
  — Each query requires only a few inner products
    \[
    \mathbf{q}^T \tilde{\mathbf{A}}_{m \times n} = \sum_{i=1}^{k} \sigma_i (\mathbf{q}^T \mathbf{u}_i) \mathbf{v}_i^T
    \]

• Latent semantic associations are made
  — Relevant docs not found by direct matching show up
  — *Latent Semantic Indexing* (LSI)

Disadvantages

• Adding & deleting docs requires updating & downdating SVD
Pros & Cons

Advantages

• Compression
  — A replaced with a few sing values & vectors (but dense)
  — They are determined & normalized only once

• SPEED!
  — Each query requires only a few inner products
    \[ q^T \tilde{A}_{m \times n} = \sum_{i=1}^{k} \sigma_i (q^T u_i) \tilde{v}_i^T \]

• Latent semantic associations are made
  — Relevant docs not found by direct matching show up
  — *Latent Semantic Indexing* (LSI)

Disadvantages

• Adding & deleting docs requires updating & downdating SVD
• Determining optimal \( k \) is not easy (empirical tuning required)
Other Fourier Expansions ??
Other Fourier Expansions ??

Truncated URV Factorizations
Other Fourier Expansions ??

Truncated URV Factorizations

DFT — FFT
Other Fourier Expansions ??

Truncated URV Factorizations

DFT — FFT

• No compression — no oscillatory components
Other Fourier Expansions??

Truncated URV Factorizations

DFT — FFT

- No compression — no oscillatory components

Haar Transform

\[
H_2 = \begin{bmatrix}
1 & 1 \\
1 & -1
\end{bmatrix}
\]

\[
H_4 = \begin{bmatrix}
1 & 1 & 1 & 0 \\
1 & 1 & -1 & 0 \\
1 & -1 & 0 & 1 \\
1 & -1 & 0 & -1
\end{bmatrix}
\]
Other Fourier Expansions ??

Truncated URV Factorizations

DFT — FFT

- No compression — no oscillatory components

Haar Transform

\[ H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \]

\[ H_4 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \]

- \[ H_n = (I_2 \otimes H_{n/2}) P_n \begin{bmatrix} H_{n/2} \\ I_{n/2} \end{bmatrix} \Rightarrow H_n x \text{ is Fast!} \quad \text{(if } n=2^p) \]
Other Fourier Expansions ??

Truncated URV Factorizations

DFT — FFT

- No compression — no oscillatory components

Haar Transform

\[ H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad H_4 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \]

- \[ H_n = (I_2 \otimes H_{n/2}) P_n \begin{bmatrix} H_{n/2} \\ I_{n/2} \end{bmatrix} \Rightarrow H_n x \text{ is Fast!} \quad \text{(if } n=2^p) \]

- Factor \[ A = H_m B H_n^T = \sum_{i,j} \beta_{ij} h_i h_j^T \quad \text{(h's only use -1, 0, 1)} \]
Other Fourier Expansions ??

Truncated URV Factorizations

DFT — FFT

- No compression — no oscillatory components

Haar Transform

\[
H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad H_4 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix}
\]

- \( H_n = (I_2 \otimes H_{n/2})P_n \left( \begin{array}{c} H_{n/2} \\ I_{n/2} \end{array} \right) \Rightarrow H_n x \text{ is Fast!} \quad \text{(if } n=2^p) \)

- Factor \( A = H_mBH_n^T = \sum_{i,j} \beta_{ij} h_i h_j^T \) \quad (h's only use -1, 0, 1)

- More than a few \( \beta_{ij} \)'s may be needed
Other Fourier Expansions ??

Truncated URV Factorizations

DFT — FFT

• No compression — no oscillatory components

Haar Transform

\[ H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad H_4 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \]

• \( H_n = (I_2 \otimes H_{n/2}) P_n \begin{bmatrix} H_{n/2} \\ I_{n/2} \end{bmatrix} \Rightarrow H_n x \text{ is Fast!} \quad \text{(if } n=2^p)\)

• Factor \( A = H_mBH_n^T = \sum_{i,j} \beta_{ij} h_i h_j^T \)
  
  — More than a few \( \beta_{ij} \)'s may be needed
  
  — Needs padding if \( m \) or \( n \) not a power of 2
Other Fourier Expansions ??

Truncated URV Factorizations

DFT — FFT

- No compression — no oscillatory components

Haar Transform

\[ H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \quad H_4 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \]

- \( H_n = (I_2 \otimes H_{n/2})P_n \left[ \begin{array}{c} H_{n/2} \\ I_{n/2} \end{array} \right] \Rightarrow H_n x \text{ is Fast! (if } n=2^p) \)

- Factor \( A = H_m B H_n^T = \sum_{i,j} \beta_{ij} h_i h_j^T \) (h's only use -1, 0, 1)

- More than a few \( \beta_{ij} \)'s may be needed
- Needs padding if \( m \) or \( n \) not a power of 2

Semidiscrete Decomposition

- Approximate \( A \approx \sum_{i=1}^k \alpha_i x_i y_j \) \( x_i \) and \( y_j \) only use -1, 0, or 1 (T. Kolda and D. O’Leary, 1998)
Other Fourier Expansions ??

Truncated URV Factorizations

DFT — FFT

- No compression — no oscillatory components

Haar Transform

\[ H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad H_4 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \]

- \( H_n = (I_2 \otimes H_{n/2}) P_n \begin{bmatrix} H_{n/2} \\ I_{n/2} \end{bmatrix} \Rightarrow H_n x \) is Fast! (if \( n=2^p \))

- Factor \( A = H_m B H_n^T = \sum_{i,j} \beta_{ij} h_i h_j^T \) (h’s only use -1, 0, 1)
  - More than a few \( \beta_{ij} \)’s may be needed
  - Needs padding if \( m \) or \( n \) not a power of 2

Semidiscrete Decomposition

- Approximate \( A \approx \sum_{i=1}^k \alpha_i x_i y_j \) (T. Kolda and D. O’Leary, 1998)
  - \( x_i \) and \( y_j \) only use -1, 0, or 1

Other Wavelet Transforms?
Link Analysis (Think Web)

How To Take Advantage of Link Structure?
How To Take Advantage of Link Structure?

Indexing and Ranking

- Still must index key terms on each page
Link Analysis (Think Web)

How To Take Advantage of Link Structure?

Indexing and Ranking

- Still must index key terms on each page
  - Robots crawl the web — software does indexing
How To Take Advantage of Link Structure?

Indexing and Ranking

- Still must index key terms on each page
  - Robots crawl the web — software does indexing

- Inverted file structure
  - $\text{Term}_1 \rightarrow P_i, P_j, \ldots$
How To Take Advantage of Link Structure?

Indexing and Ranking

- Still must index key terms on each page
  - Robots crawl the web — software does indexing

- Inverted file structure
  - $\text{Term}_1 \rightarrow P_i, P_j, \ldots$
  - $\text{Term}_2 \rightarrow P_k, P_l, \ldots$
  - \vdots
Link Analysis (Think Web)

How To Take Advantage of Link Structure?

Indexing and Ranking

- Still must index key terms on each page
  - Robots crawl the web — software does indexing

- Inverted file structure
  - $\text{Term}_1 \rightarrow P_i, P_j, \ldots$
  - $\text{Term}_2 \rightarrow P_k, P_l, \ldots$
  - $\vdots$

- Attach an importance rating to $P_i, P_j, P_k, P_l, \ldots$
Link Analysis (Think Web)

How To Take Advantage of Link Structure?

Indexing and Ranking

- Still must index key terms on each page
  - Robots crawl the web — software does indexing

- Inverted file structure
  - $\text{Term}_1 \rightarrow P_i, P_j, \ldots$
  - $\text{Term}_2 \rightarrow P_k, P_l, \ldots$
    
  :  

- Attach an importance rating to $P_i, P_j, P_k, P_l, \ldots$

- Direct query matching
  - $Q = \text{Term}_1, \text{Term}_2, \ldots$ produces $P_i, P_j, P_k, P_l, \ldots$
Link Analysis (Think Web)

How To Take Advantage of Link Structure?

Indexing and Ranking

• Still must index key terms on each page
  — Robots crawl the web — software does indexing

• Inverted file structure
  — $\text{Term}_1 \rightarrow P_i, P_j, \ldots$
  — $\text{Term}_2 \rightarrow P_k, P_l, \ldots$
  \vdots

• Attach an importance rating to $P_i, P_j, P_k, P_l, \ldots$

• Direct query matching
  — $Q = \text{Term}_1, \text{Term}_2, \ldots$ produces $P_i, P_j, P_k, P_l, \ldots$

• Return $P_i, P_j, P_k, P_l, \ldots$ to user in order of importance
How To Measure “Importance”
How To Measure “Importance”

Hubs & Authorities

• Good hub pages point to good authority pages
• Good authorities are pointed to by good hubs
How To Measure “Importance”

Hubs & Authorities

- Good hub pages point to good authority pages
- Good authorities are pointed to by good hubs

HITS Algorithm

- For each query a “neighborhood graph” $N$ is built
How To Measure “Importance”

Hubs & Authorities

- Good hub pages point to good authority pages
- Good authorities are pointed to by good hubs

HITS Algorithm

- For each query a “neighborhood graph” $N$ is built
- Hub and authority scores for nodes in $N$ are computed
  - Eigenvector computation
How To Measure “Importance”

Hubs & Authorities

• Good hub pages point to good authority pages
• Good authorities are pointed to by good hubs

HITS Algorithm

• For each query a “neighborhood graph” $N$ is built
• Hub and authority scores for nodes in $N$ are computed
  — Eigenvector computation
• Works, but requires new graph for each query
How To Measure “Importance”

Hubs & Authorities (Jon Kleinberg 1998)

- Good hub pages point to good authority pages
- Good authorities are pointed to by good hubs

HITS Algorithm

- For each query a “neighborhood graph” $N$ is built
- Hub and authority scores for nodes in $N$ are computed
  - Eigenvector computation
- Works, but requires new graph for each query
- Similar ideas in TEOMA.com
Google’s Idea

PageRank

(Sergey Brin & Lawrence Page 1998)
Google’s Idea

PageRank

(Sergey Brin & Lawrence Page 1998)

- Your page $P$ has some rank $r(P)$
Google’s Idea

PageRank
(Sergey Brin & Lawrence Page 1998)

- Your page $P$ has some rank $r(P)$
- Adjust $r(P)$ higher or lower depending on ranks of pages that point to $P$. 
Google’s Idea

PageRank

(Sergey Brin & Lawrence Page 1998)

- Your page $P$ has some rank $r(P)$

- Adjust $r(P)$ higher or lower depending on ranks of pages that point to $P$

- Importance is not number of in-links or out-links
Google’s Idea

PageRank

(Sergey Brin & Lawrence Page 1998)

• Your page $P$ has some rank $r(P)$

• Adjust $r(P)$ higher or lower depending on ranks of pages that point to $P$

• Importance is not number of in-links or out-links
  — One link to $P$ from Yahoo! is important
  — Many links to $P$ from me is not
Google’s Idea

PageRank

(Sergey Brin & Lawrence Page 1998)

- Your page $P$ has some rank $r(P)$

- Adjust $r(P)$ higher or lower depending on ranks of pages that point to $P$

- Importance is not number of in-links or out-links
  - One link to $P$ from Yahoo! is important
  - Many links to $P$ from me is not

- But if Yahoo! points to many places, the value of the link to $P$ is diluted
PageRank

The Definition

\[ r(P) = \sum_{P \in \mathcal{B}_P} \frac{r(P)}{|P|} \]

- \( \mathcal{B}_P = \{ \text{all pages pointing to } P \} \)
- \(|P| = \text{number of out links from } P \)
PageRank

The Definition

- \( r(P) = \sum_{P \in B_P} \frac{r(P)}{|P|} \)
- \( B_P = \{ \text{all pages pointing to } P \} \)
- \(|P| = \text{number of out links from } P\)

Successive Refinement

- Start with \( r_0(P_i) = 1/n \) for all pages \( P_1, P_2, \ldots, P_n \)
- Iteratively refine rankings for each page
PageRank

The Definition

- \( r(P) = \sum_{P \in B_P} \frac{r(P)}{|P|} \)

- \( B_P = \{ \text{all pages pointing to } P \} \)

- \( |P| = \text{number of out links from } P \)

Successive Refinement

- Start with \( r_0(P_i) = \frac{1}{n} \) for all pages \( P_1, P_2, \ldots, P_n \)

- Iteratively refine rankings for each page

- \( r_1(P_i) = \sum_{P \in B_{P_i}} \frac{r_0(P)}{|P|} \)
PageRank

The Definition

- \( r(P) = \sum_{P \in B_P} \frac{r(P)}{|P|} \)
- \( B_P = \{ \text{all pages pointing to } P \} \)
- \( |P| = \text{number of out links from } P \)

Successive Refinement

- Start with \( r_0(P_i) = \frac{1}{n} \) for all pages \( P_1, P_2, \ldots, P_n \)
- Iteratively refine rankings for each page

- \( r_1(P_i) = \sum_{P \in B_{P_i}} \frac{r_0(P)}{|P|} \)
- \( r_2(P_i) = \sum_{P \in B_{P_i}} \frac{r_1(P)}{|P|} \)
PageRank

The Definition

- \( r(P) = \sum_{P \in \mathcal{B}_P} \frac{r(P)}{|P|} \)

- \( \mathcal{B}_P = \{ \text{all pages pointing to } P \} \)

- \( |P| = \text{number of out links from } P \)

Successive Refinement

- Start with \( r_0(P_i) = \frac{1}{n} \) for all pages \( P_1, P_2, \ldots, P_n \)

- Iteratively refine rankings for each page

- \( r_1(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_0(P)}{|P|} \)

- \( r_2(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_1(P)}{|P|} \)

- \( \vdots \)

- \( r_{j+1}(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_j(P)}{|P|} \)
In Matrix Notation

After Step $j$

- $\pi_j^T = [r_j(P_1), r_j(P_2), \cdots, r_j(P_n)]$
In Matrix Notation

After Step $j$

- $\pi_j^T = [r_j(P_1), r_j(P_2), \cdots, r_j(P_n)]$

- $\pi_{j+1}^T = \pi_j^T P$ where $p_{ij} = \begin{cases} 1/|P_i| & \text{if } i \to j \\ 0 & \text{otherwise} \end{cases}$
In Matrix Notation

After Step $j$

- $\pi_j^T = [r_j(P_1), r_j(P_2), \cdots, r_j(P_n)]$

- $\pi_{j+1}^T = \pi_j^T P$ where $p_{ij} = \begin{cases} \frac{1}{|P_i|} & \text{if } i \rightarrow j \\ 0 & \text{otherwise} \end{cases}$

- PageRank $= \lim_{j \to \infty} \pi_j^T = \pi^T$ (provided limit exists)
In Matrix Notation

After Step \( j \)

- \( \pi_j^T = [r_j(P_1), r_j(P_2), \cdots, r_j(P_n)] \)

- \( \pi_{j+1}^T = \pi_j^T P \) where \( p_{ij} = \begin{cases} 1/|P_i| & \text{if } i \rightarrow j \\ 0 & \text{otherwise} \end{cases} \)

- PageRank = \( \lim_{j \to \infty} \pi_j^T = \pi^T \) (provided limit exists)

It’s A Markov Chain

- \( P = [p_{ij}] \) is a stochastic matrix (row sums = 1)
In Matrix Notation

After Step $j$

- $\pi^T_j = [r_j(P_1), r_j(P_2), \cdots, r_j(P_n)]$

- $\pi^T_{j+1} = \pi^T_j P$ where $p_{ij} = \begin{cases} 1/|P_i| & \text{if } i \to j \\ 0 & \text{otherwise} \end{cases}$

- PageRank $= \lim_{j \to \infty} \pi^T_j = \pi^T$ (provided limit exists)

It’s A Markov Chain

- $P = [p_{ij}]$ is a stochastic matrix (row sums $= 1$)

- Each $\pi^T_j$ (and $\pi^T$) is a probability vector $\left( \sum_i r_j(P_i) = 1 \right)$
In Matrix Notation

After Step $j$

- $\pi_j^T = [r_j(P_1), r_j(P_2), \cdots, r_j(P_n)]$

- $\pi_{j+1}^T = \pi_j^T P$ where $p_{ij} = \begin{cases} 1/|P_i| & \text{if } i \to j \\ 0 & \text{otherwise} \end{cases}$

- PageRank $= \lim_{j \to \infty} \pi_j^T = \pi^T$ (provided limit exists)

It’s A Markov Chain

- $P = [p_{ij}]$ is a stochastic matrix (row sums $= 1$)

- Each $\pi_j^T$ (and $\pi^T$) is a probability vector $\left(\sum_i r_j(P_i) = 1\right)$

- $\pi_{j+1}^T = \pi_j^T P$ is random walk on the graph defined by links
Random Surfer

Web Surfer Randomly Clicks On Links

- Long-run proportion of time on page $P_i$ is $\pi_i$ (Back button not a link)
Random Surfer

Web Surfer Randomly Clicks On Links

- Long-run proportion of time on page $P_i$ is $\pi_i$

Problems

- Dead end page (nothing to click on)
Random Surfer

Web Surfer Randomly Clicks On Links

- Long-run proportion of time on page $P_i$ is $\pi_i$

Problems

- Dead end page (nothing to click on)
  - No convergence!
Random Surfer

Web Surfer Randomly Clicks On Links

- Long-run proportion of time on page $P_i$ is $\pi_i$

Problems

- Dead end page (nothing to click on)
  - No convergence!
- Could get trapped into a cycle ($P_i \rightarrow P_j \rightarrow P_i$)
Random Surfer

Web Surfer Randomly Clicks On Links

- Long-run proportion of time on page $P_i$ is $\pi_i$

Problems

- Dead end page (nothing to click on)
  - No convergence!

- Could get trapped into a cycle ($P_i \rightarrow P_j \rightarrow P_i$)
  - No convergence!
Random Surfer

Web Surfer Randomly Clicks On Links

- Long-run proportion of time on page $P_i$ is $\pi_i$

Problems

- Dead end page (nothing to click on)
  - No convergence!

- Could get trapped into a cycle $P_i \to P_j \to P_i$
  - No convergence!

Convergence

- Markov chain must be irreducible and aperiodic
Random Surfer

Web Surfer Randomly Clicks On Links

- Long-run proportion of time on page $P_i$ is $\pi_i$

Problems

- Dead end page (nothing to click on)
  - No convergence!
- Could get trapped into a cycle ($P_i \rightarrow P_j \rightarrow P_i$)
  - No convergence!

Convergence

- Markov chain must be irreducible and aperiodic

Bored Surfer Enters Random URL
Random Surfer

Web Surfer Randomly Clicks On Links

- Long-run proportion of time on page $P_i$ is $\pi_i$

Problems

- Dead end page (nothing to click on)
  - No convergence!
- Could get trapped into a cycle $(P_i \rightarrow P_j \rightarrow P_i)$
  - No convergence!

Convergence

- Markov chain must be irreducible and aperiodic

Bored Surfer Enters Random URL

- Replace $P$ by $\tilde{P} = \alpha P + (1 - \alpha)E$ where $e_{ij} = 1/n$ \[ \alpha \approx .85 \]
Random Surfer

Web Surfer Randomly Clicks On Links

- Long-run proportion of time on page $P_i$ is $\pi_i$

Problems

- Dead end page (nothing to click on)
  - No convergence!
- Could get trapped into a cycle ($P_i \rightarrow P_j \rightarrow P_i$)
  - No convergence!

Convergence

- Markov chain must be irreducible and aperiodic

Bored Surfer Enters Random URL

- Replace $P$ by $\tilde{P} = \alpha P + (1 - \alpha)E$ where $e_{ij} = 1/n$ $\alpha \approx .85$
  - Different $E$'s and $\alpha$'s allow customization & speedup
Computing $\pi^T$

World’s Largest Eigenvector Problem (C. Moler)

- Solve $\pi^T = \pi^T P$ (stationary distribution vector)
Computing $\pi^T$

World’s Largest Eigenvector Problem (C. Moler)

- Solve $\pi^T = \pi^T P$

- $\pi^T (I - P) = 0$

(stationary distribution vector)

(too big for direct solves)
Computing $\pi^T$

World’s Largest Eigenvector Problem (C. Moler)

- Solve $\pi^T = \pi^T P$ (stationary distribution vector)
- $\pi^T (I - P) = 0$ (too big for direct solves)
- Start with $\pi_0^T = e/n$ and iterate $\pi_{j+1}^T = \pi_j^T P$ (power method)
Computing $\pi^T$

World’s Largest Eigenvector Problem (C. Moler)

- Solve $\pi^T = \pi^T P$ (stationary distribution vector)
- $\pi^T (I - P) = 0$ (too big for direct solves)
- Start with $\pi_0^T = e/n$ and iterate $\pi_{j+1}^T = \pi_j^T P$ (power method)

Updating Is A Big Problem

- Link structure of web is extremely dynamic
Computing $\pi^T$

World’s Largest Eigenvector Problem  (C. Moler)

• Solve $\pi^T = \pi^T P$  
  (stationary distribution vector)

• $\pi^T (I - P) = 0$  
  (too big for direct solves)

• Start with $\pi_0^T = e/n$ and iterate $\pi_{j+1}^T = \pi_j^T P$  
  (power method)

Updating Is A Big Problem

• Link structure of web is extremely dynamic
  — Links on CNN point to different pages every day (hour)
Computing $\pi^T$

World’s Largest Eigenvector Problem (C. Moler)

- Solve $\pi^T = \pi^T P$ (stationary distribution vector)
- $\pi^T(I - P) = 0$ (too big for direct solves)
- Start with $\pi_0^T = e/n$ and iterate $\pi_{j+1}^T = \pi_j^T P$ (power method)

Updating Is A Big Problem

- Link structure of web is extremely dynamic
  - Links on CNN point to different pages every day (hour)
  - Links are added and deleted every sec (milli-sec?)
Computing $\pi^T$

World’s Largest Eigenvector Problem (C. Moler)

- Solve $\pi^T = \pi^T P$ (stationary distribution vector)
- $\pi^T (I - P) = 0$ (too big for direct solves)
- Start with $\pi_0^T = e/n$ and iterate $\pi_{j+1}^T = \pi_j^T P$ (power method)

Updating Is A Big Problem

- Link structure of web is extremely dynamic
  - Links on CNN point to different pages every day (hour)
  - Links are added and deleted every sec (milli-sec?)
- Google says every 3 to 4 weeks just start from scratch
Computing $\pi^T$

World’s Largest Eigenvector Problem (C. Moler)

- Solve $\pi^T = \pi^T P$
  (stationary distribution vector)
- $\pi^T(I - P) = 0$
  (too big for direct solves)
- Start with $\pi_0^T = e/n$ and iterate $\pi_{j+1}^T = \pi_j^T P$
  (power method)

Updating Is A Big Problem

- Link structure of web is extremely dynamic
  - Links on CNN point to different pages every day (hour)
  - Links are added and deleted every sec (milli-sec?)
- Google says every 3 to 4 weeks just start from scratch
- Old results don’t help to restart (even if size doesn’t change)
  - Cutoff phenomenon in random walks (P. Diaconis, 1996)
## Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEATURES</td>
<td>LSI</td>
<td>LINK ANALYSIS</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----</td>
<td>---------------</td>
</tr>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEATURES</td>
<td>LSI</td>
<td>LINK ANALYSIS</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----</td>
<td>---------------</td>
</tr>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEATURES</td>
<td>LSI</td>
<td>LINK ANALYSIS</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------</td>
<td>---------------</td>
</tr>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B⁻</td>
<td></td>
</tr>
</tbody>
</table>


# Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B⁻</td>
<td>A⁺</td>
</tr>
</tbody>
</table>

---

---
# Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B⁻</td>
<td>A⁺</td>
</tr>
<tr>
<td>Easy To Update</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B⁻</td>
<td>A⁺</td>
</tr>
<tr>
<td>Easy To Update</td>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>
# Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B⁻</td>
<td>A⁺</td>
</tr>
<tr>
<td>Easy To Update</td>
<td>D</td>
<td>F (↑↑?)</td>
</tr>
</tbody>
</table>
## Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B⁻</td>
<td>A⁺</td>
</tr>
<tr>
<td>Easy To Update</td>
<td>D</td>
<td>F (?↑?)</td>
</tr>
<tr>
<td>Scales Up</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B⁻</td>
<td>A⁺</td>
</tr>
<tr>
<td>Easy To Update</td>
<td>D</td>
<td>F (↑↑?)</td>
</tr>
<tr>
<td>Scales Up</td>
<td>D⁻</td>
<td></td>
</tr>
</tbody>
</table>
## Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B−</td>
<td>A+</td>
</tr>
<tr>
<td>Easy To Update</td>
<td>D</td>
<td>F (?↑?)</td>
</tr>
<tr>
<td>Scales Up</td>
<td>D−</td>
<td>A</td>
</tr>
</tbody>
</table>
## Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B⁻</td>
<td>A⁺</td>
</tr>
<tr>
<td>Easy To Update</td>
<td>D</td>
<td>F (?↑?)</td>
</tr>
<tr>
<td>Scales Up</td>
<td>D⁻</td>
<td>A</td>
</tr>
<tr>
<td>Takes Advantage of Link Structure</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B⁻</td>
<td>A⁺</td>
</tr>
<tr>
<td>Easy To Update</td>
<td>D</td>
<td>F (?↑?)</td>
</tr>
<tr>
<td>Scales Up</td>
<td>D⁻</td>
<td>A</td>
</tr>
<tr>
<td>Takes Advantage of Link Structure</td>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>
## Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B−</td>
<td>A+</td>
</tr>
<tr>
<td>Easy To Update</td>
<td>D</td>
<td>F (?↑?)</td>
</tr>
<tr>
<td>Scales Up</td>
<td>D−</td>
<td>A</td>
</tr>
<tr>
<td>Takes Advantage of Link Structure</td>
<td>F</td>
<td>A+</td>
</tr>
</tbody>
</table>
# Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B⁻</td>
<td>A⁺</td>
</tr>
<tr>
<td>Easy To Update</td>
<td>D</td>
<td>F (?↑?)</td>
</tr>
<tr>
<td>Scales Up</td>
<td>D⁻</td>
<td>A</td>
</tr>
<tr>
<td>Takes Advantage of Link Structure</td>
<td>F</td>
<td>A⁺</td>
</tr>
</tbody>
</table>

## Goals

- Do better job using link structure to reveal hidden connections
## Report Card

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>LSI</th>
<th>LINK ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveals Hidden Patterns</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Speed</td>
<td>B⁻</td>
<td>A⁺</td>
</tr>
<tr>
<td>Easy To Update</td>
<td>D</td>
<td>F (?↑?)</td>
</tr>
<tr>
<td>Scales Up</td>
<td>D⁻</td>
<td>A</td>
</tr>
<tr>
<td>Takes Advantage of Link Structure</td>
<td>F</td>
<td>A⁺</td>
</tr>
</tbody>
</table>

### Goals

- Do better job using link structure to reveal hidden connections
- Improve updating
Hybrid Approach

The Idea

- Use link structure to define measure of page (doc) contiguity
  - What’s the “distance” from $P_i$ to $P_j$?
Hybrid Approach

The Idea

- Use link structure to define measure of page (doc) contiguity
  - What’s the “distance” from $P_i$ to $P_j$?
  - Link structure $\Rightarrow \delta_{ij} \neq \delta_{ji}$
Hybrid Approach

The Idea

- Use link structure to define measure of page (doc) contiguity
  - What’s the “distance” from \( P_i \) to \( P_j \)?
  - Link structure \( \iff \delta_{ij} \neq \delta_{ji} \)

1. Compute the distance \( \delta_{ij} \) from \( P_i \) to \( P_j \) for all \( i, j \)
   - Keep only those for which \( \delta_{ij} < \tau \) (provides sparsity)
Hybrid Approach

The Idea

- Use link structure to define measure of page (doc) contiguity
  - What’s the “distance” from $P_i$ to $P_j$?
  - Link structure $\implies \delta_{ij} \neq \delta_{ji}$

1. Compute the distance $\delta_{ij}$ from $P_i$ to $P_j$ for all $i, j$
   - Keep only those for which $\delta_{ij} < \tau$ (provides sparsity)
   - File structure:
     \[
     \begin{cases}
     P_1 \rightarrow P_i, P_j, \ldots \\
     P_2 \rightarrow P_k, P_l, \ldots \\
     \vdots
     \end{cases}
     \]
Hybrid Approach

The Idea

- Use link structure to define measure of page (doc) contiguity
  - What’s the “distance” from $P_i$ to $P_j$?
  - Link structure $\implies \delta_{ij} \neq \delta_{ji}$

1. Compute the distance $\delta_{ij}$ from $P_i$ to $P_j$ for all $i, j$
   - Keep only those for which $\delta_{ij} < \tau$ (provides sparsity)
   - File structure: $\left\{ \begin{array}{c}
   P_1 \rightarrow P_i, P_j, \ldots \\
   P_2 \rightarrow P_k, P_l, \ldots \\
   \vdots
   \end{array} \right.$

2. Match query most relevant page(s) $\mathcal{P}$
   - LSI — Link analysis — You pick
Hybrid Approach

The Idea

- Use link structure to define measure of page (doc) contiguity
  - What’s the “distance” from $P_i$ to $P_j$?
  - Link structure $\implies \delta_{ij} \neq \delta_{ji}$

1. Compute the distance $\delta_{ij}$ from $P_i$ to $P_j$ for all $i, j$
   - Keep only those for which $\delta_{ij} < \tau$ (provides sparsity)
   - File structure: $\begin{cases} P_1 \rightarrow P_i, P_j, \ldots \\ P_2 \rightarrow P_k, P_l, \ldots \\ \vdots \end{cases}$

2. Match query most relevant page(s) $\mathcal{P}$
   - LSI — Link analysis — You pick

3. Return $\mathcal{P}$ together with those $\mathcal{P} \rightarrow P_i, P_j, P_k, P_l, \ldots$
What’s the “distance” from $D_i$ to $D_j$?
What’s the “distance” from $D_i$ to $D_j$?

- LSI uses $\delta_{ij} = \cos \theta_{ij} = \delta_{ji}$
Distance

What’s the “distance” from $D_i$ to $D_j$?

- LSI uses $\delta_{ij} = \cos \theta_{ij} = \delta_{ji}$

\{ Based only on term frequencies \\
No link structure \}
Distance

What’s the “distance” from $D_i$ to $D_j$?

- LSI uses $\delta_{ij} = \cos \theta_{ij} = \delta_{ji}$

Directed Link Structure $\Rightarrow$ Nonsymmetric Metric

Based only on term frequencies
No link structure