# In Search of a Matrix


**CARL D. MEYER**

North Carolina State University

Raleigh, NC


Canadian Mathematical Society

University of Saskatchewan

Saskatoon, Saskatchewan, June 2-4, 2001

# Outline

- Background & History
- Vector Space Approach
- Frequency Weighting
- Compression & Noise Reduction
- Examples
- LSI with SVD
- LSI with URV & Centroids
- References

# Background

**Goal:** Identify documents that best match users query

**Measures**

- Recall $= \frac{\#relevant\ docs\ retrieved}{\#docs\ in\ collection}$ (max # useful docs)
- Precision $= \frac{\#relevant\ docs\ retrieved}{\#docs\ retrieved}$ (min # useless docs)
- Do it $FAST!$

**Methods**

- Combinatorial
- Statistical
- Hashing
- Pattern matching
- Vector Space

# SMART

(**S**ystem for the **M**echanical **A**nalysis and **R**etrieval of **T**ext)

## 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*)

## Index Each Document (by human or by computer)

- Count $f_{ij}$ = # times term $i$ appears in document $j$

## Unweighted Term–Document Matrix

$$\begin{array}{c c} & \begin{array}{cccc} \textsc{Doc } 1 & \textsc{Doc } 2 & \cdots & \textsc{Doc } n \end{array} \\ \begin{array}{c} \textsc{Term } 1 \\ \textsc{Term } 2 \\ \vdots \\ \textsc{Term } m \end{array} & \begin{pmatrix} f_{\mathbf{11}} & f_{\mathbf{12}} & \cdots & f_{\mathbf{1}n} \\ f_{\mathbf{21}} & f_{\mathbf{22}} & \cdots & f_{\mathbf{2}n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m\mathbf{1}} & f_{m\mathbf{2}} & \cdots & f_{mn} \end{pmatrix} \end{array} = \mathbf{A}_{m\times n}$$

## Features

- $\mathbf{A} \geq \mathbf{0}$
- **A** can be really big! — Terms = $O(\mathbf{10^6})$ Docs = $O(\mathbf{10^7})$
- **A** is sparse — but otherwise unstructured
- **A** contains a lot of uncertainty (noise)

# Example

(M. W. Berry & M. Browne, 1999, SIAM)

| Terms | | Documents | |
|---|---|---|---|
| T1: | Bab(y, ies, y's) | D1: | Infant and toddler first aid |
| T2: | Child(ren's) | D2: | Babies and children's room for your home |
| T3: | Guide | D3: | Child safety at home |
| T4: | Health | D4: | Your baby's health and safety: from infant to toddler |
| T5: | Home | D5: | Baby proofing basics |
| T6: | Infant | D6: | Your guide to easy rust proofing |
| T7: | Proofing | D7: | Beanie babies collector's guide |
| T8: | Safety | | |
| T9: | Toddler | | |

$$
\mathbf{A} = \begin{array}{c} \\ T1 \\ T2 \\ T3 \\ T4 \\ T5 \\ T6 \\ T7 \\ T8 \\ T9 \end{array}
\begin{array}{c} \begin{array}{ccccccc} D1 & D2 & D3 & D4 & D5 & D6 & D7 \end{array} \\
\begin{pmatrix}
0 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0
\end{pmatrix}_{9\times 7} \end{array}
$$

# Example

(M. W. Berry & M. Browne, 1999, SIAM)

| Terms | Documents |
|---|---|
| T1: Bab(y, ies, y's) | D1: Infant and toddler first aid |
| T2: Child(ren's) | D2: Babies and children's room for your home |
| T3: Guide | D3: Child safety at home |
| T4: Health | D4: Your baby's health and safety: from infant to toddler |
| T5: Home | D5: Baby proofing basics |
| T6: Infant | D6: Your guide to easy rust proofing |
| T7: Proofing | D7: Beanie babies collector's guide |
| T8: Safety | |
| T9: Toddler | |

$$
\mathbf{A} = \begin{array}{c} \\ \mathrm{T1} \\ \mathrm{T2} \\ \mathrm{T3} \\ \mathrm{T4} \\ \mathrm{T5} \\ \mathrm{T6} \\ \mathrm{T7} \\ \mathrm{T8} \\ \mathrm{T9} \end{array}
\begin{array}{c}
\begin{array}{ccccccc} \mathrm{D1} & \mathrm{D2} & \mathrm{D3} & \mathrm{D4} & \mathrm{D5} & \mathrm{D6} & \mathrm{D7} \end{array} \\
\begin{pmatrix}
0 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0
\end{pmatrix}_{9 \times 7}
\end{array}
$$

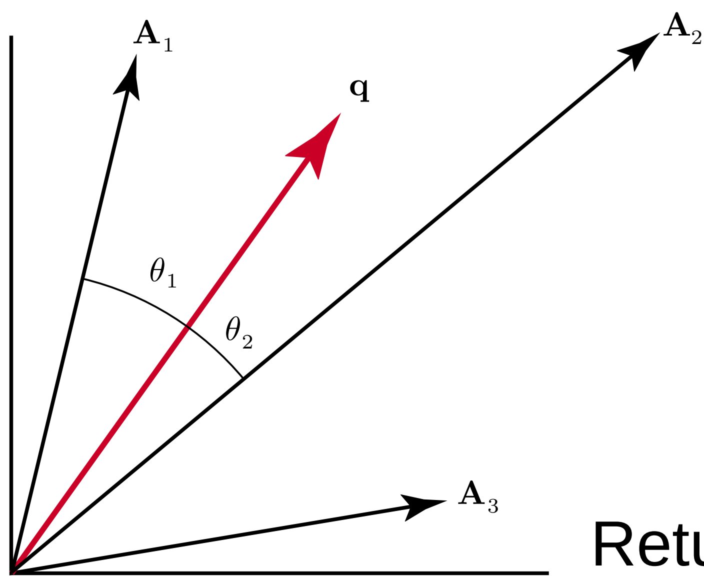
# Query Matching

## Query Vector

- $\mathbf{q}^T = (q_1, q_2, \ldots, q_m)$ 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 $\mathbf{q}$ to each column $\mathbf{A}_k$?




$\|\mathbf{q} - \mathbf{A}_1\| < \|\mathbf{q} - \mathbf{A}_2\|$ but $\theta_2 < \theta_1$

Use $\delta_i = \cos\theta_i = \dfrac{\mathbf{q}^T \mathbf{A}_i}{\|\mathbf{q}\| \, \|\mathbf{A}_i\|}$

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 $\mathbf{1}$ but twice in Doc $\mathbf{2}$, and if $\|\mathbf{A_1}\| \approx \|\mathbf{A_2}\|$, then a query containing only $bank$ produces $\delta_\mathbf{2} \approx \mathbf{2}\delta_\mathbf{1}$ (i.e., Doc $\mathbf{2}$ is rated twice as relevant as Doc $\mathbf{1}$).

## To Compensate

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

## Query Weights

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

## To Compensate

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

# Noise in A

## Ambiguity in Vocabulary

- e.g., A $bank$ could be
  - A financial institution
  - A river side
  - A shot in the game of pool

## 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

# Practical Problems

## Simple in Theory

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

## Difficult in Practice

- Must be able to do it $FAST!$

  $\Longrightarrow$ Somehow compress the data

- Must account for variations due to ambiguity in language and variations in writing and indexing styles.

  $\Longrightarrow$ Somehow reduce "noise" or "uncertainty" in **A**

# Basis Games

## Consider Vector of Data

$\mathbf{v} = \sum_{i=1}^{n} \alpha_i \mathbf{e}_i$ with respect to o.n. basis $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n\}$

$\Longrightarrow \alpha_i = \langle \mathbf{e}_i | \mathbf{v} \rangle$ = amount of $\mathbf{v}$ in direction of $\mathbf{e}_i$

## Compression

- Select new o.n. basis $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n\}$ so that fewer vectors are needed to represent $\mathbf{v} = \sum_{i=1}^{\mathbf{r}} \beta_i \mathbf{u}_i$ $\quad (\mathbf{r} < n)$
- Data is compressed from $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$ into $\{\beta_1, \beta_2, \cdots, \beta_{\mathbf{r}}\}$

## Even More Compression

- Eliminate data lying in insignificant directions

  Arrange: $|\beta_1| \geq |\beta_2| \geq \cdots \geq |\beta_{\mathbf{k}}| \geq \epsilon > |\beta_{\mathbf{k}+1}| \cdots \geq |\beta_r|$

  Approximate: $\mathbf{v} \approx \widetilde{\mathbf{v}} = \sum_{i=1}^{\mathbf{k}} \beta_i \mathbf{u}_i$ $\quad (\mathbf{k} < \mathbf{r} < n)$
- Data now compressed from $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$ to $\{\beta_1, \beta_2, \cdots, \beta_{\mathbf{k}}\}$

# Added Benefit

## Noise Reduction

- Assume noise (or uncertainty) is nondirectional

  $\Longrightarrow$ As much noise in one direction as in any other direction

  $\Longrightarrow$ $\mathbf{v} = \sum_{i=1}^{r} \beta_i \mathbf{u}_i = \sum_{i=1}^{r} s_i \mathbf{u}_i + \sum_{i=1}^{r} \epsilon \mathbf{u}_i$ = (signal) + (noise)

- Suppose $|s_1| \geq |s_2| \geq \cdots \geq |s_k| \geq \epsilon > |s_{k+1}| \cdots \geq |s_r|$

  Drop $\beta_{k+1}, \beta_{k+2}, \cdots, \beta_r$, and use $\mathbf{v} \approx \tilde{\mathbf{v}} = \sum_{i=1}^{k} \beta_i \mathbf{u}_i$

  $\Longrightarrow$ Only a small proportion of the signal is lost

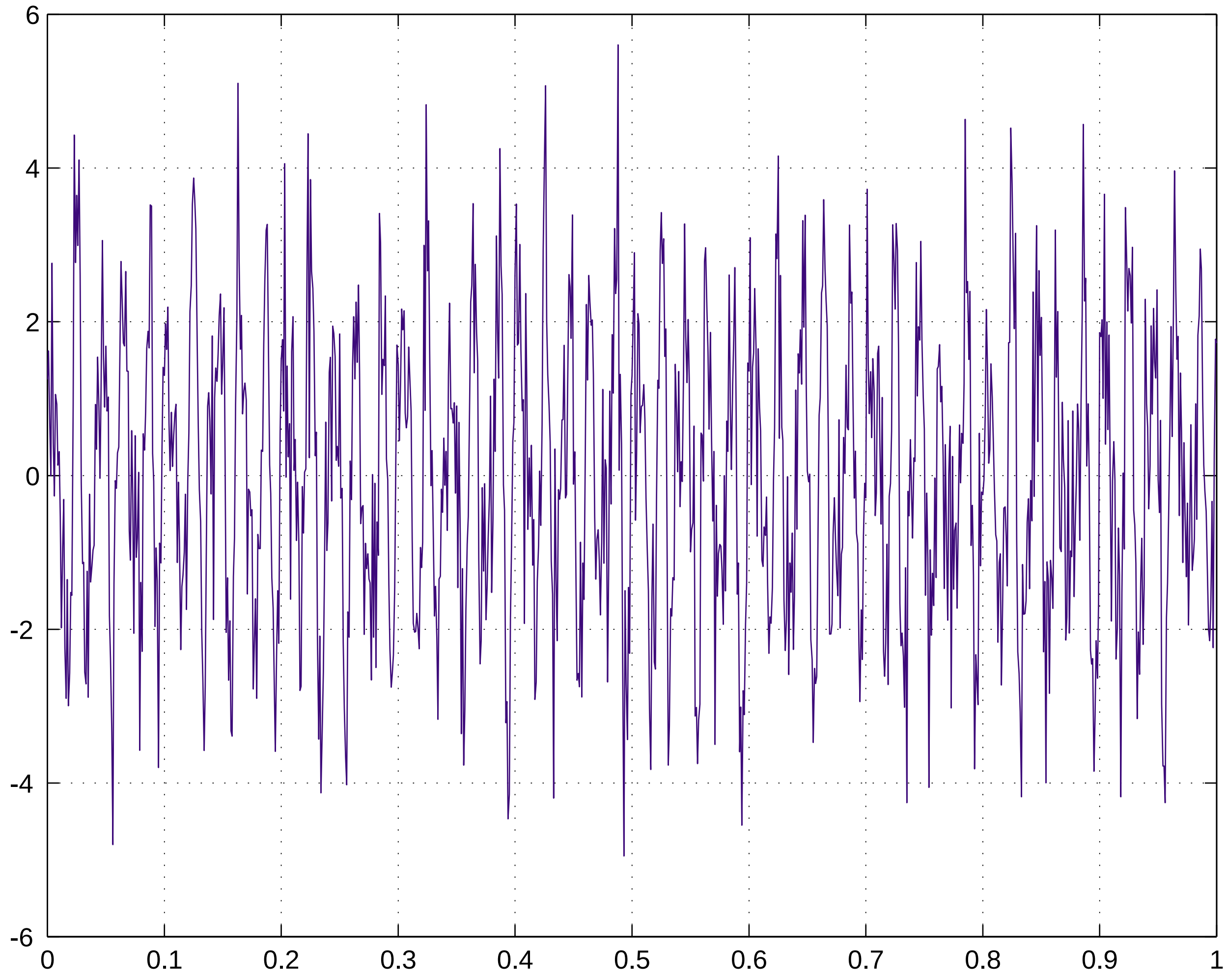
  $\Longrightarrow$ A larger proportion of the noise is lost

# Example

(Matrix Analysis and Applied Linear Algebra, SIAM, 2000)

## Sample Audio Signal 512 times for 1 sec.

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




## Goal: Compress and Reduce Noise

# Find A Better Basis

## Oscillatory $\Longrightarrow$ Cosines & Sines

$$\mathbf{x}(t) \sim \sum_{f=1}^{\infty} \alpha_f \cos 2\pi f t + \beta_f \sin 2\pi f t \qquad (\text{if } \mathbf{x}(t) \text{ was a continuous function})$$

## Discrete Time, Cosine, and Sine Vectors

$$\mathbf{t} = \begin{bmatrix} {}^{0}/_{n} \\ {}^{1}/_{n} \\ {}^{2}/_{n} \\ \vdots \\ {}^{n-1}/_{n} \end{bmatrix} \quad \cos 2\pi f \mathbf{t} = \begin{bmatrix} \cos\left(2\pi f \cdot \frac{0}{n}\right) \\ \cos\left(2\pi f \cdot \frac{1}{n}\right) \\ \cos\left(2\pi f \cdot \frac{2}{n}\right) \\ \vdots \\ \cos\left(2\pi f \cdot \frac{n-1}{n}\right) \end{bmatrix} \quad \sin 2\pi f \mathbf{t} = \begin{bmatrix} \sin\left(2\pi f \cdot \frac{0}{n}\right) \\ \sin\left(2\pi f \cdot \frac{1}{n}\right) \\ \sin\left(2\pi f \cdot \frac{2}{n}\right) \\ \vdots \\ \sin\left(2\pi f \cdot \frac{n-1}{n}\right) \end{bmatrix}$$

## Discrete Exponential Vectors

$$e^{i2\pi f \mathbf{t}} = \cos 2\pi f \mathbf{t} + i \sin 2\pi f \mathbf{t} \qquad e^{-i2\pi f \mathbf{t}} = \cos 2\pi f \mathbf{t} - i \sin 2\pi f \mathbf{t}$$

# Discrete Fourier Transform

$$\omega = e^{2\pi i/n} \qquad \mathbf{W} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{n-2} & \cdots & \omega \end{bmatrix}_{n\times n} \qquad \mathbf{W}_f = \frac{1}{2}e^{2\pi i f \mathbf{t}}$$

## Identities

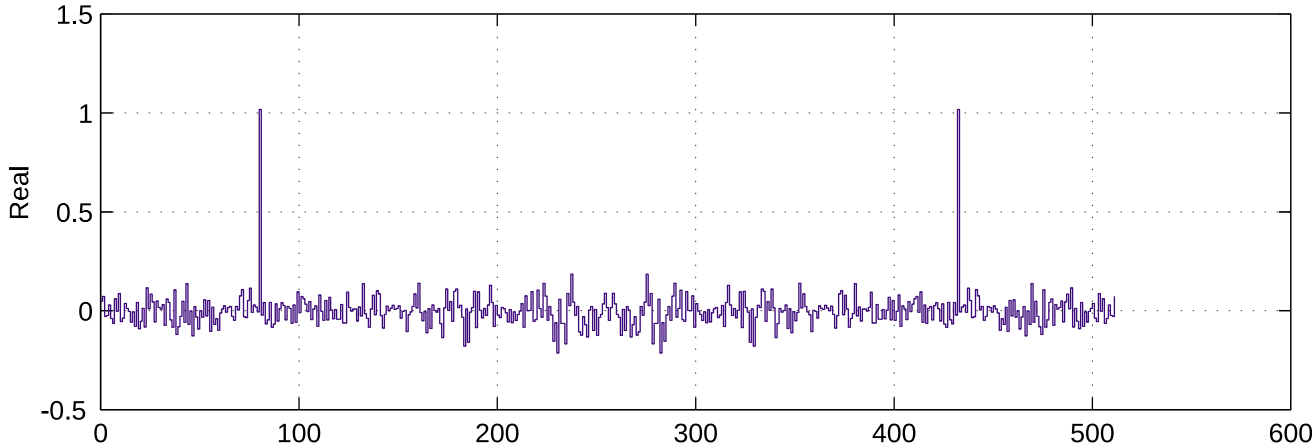
- $\cos 2\pi f\mathbf{t} = \mathbf{W}_f + \mathbf{W}_{n-f}$ $\quad (0 < f < n)$
- $\sin 2\pi f\mathbf{t} = -i(\mathbf{W}_f - \mathbf{W}_{n-f})$

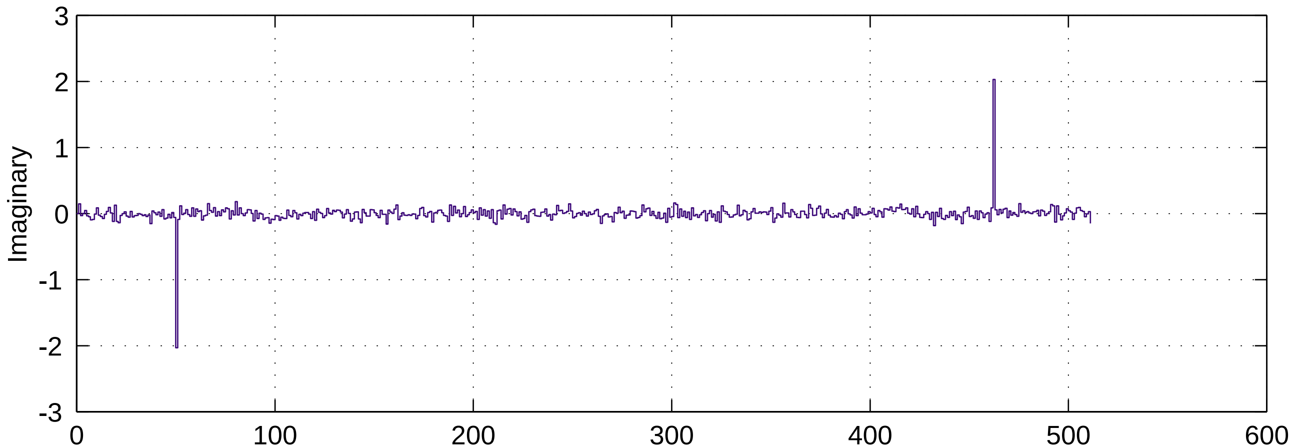
**New Basis** $= \{\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_{n-1}\}$ (Find coordinates of $\mathbf{x}$ wrt $\mathbf{W}_i$'s)

$$\mathbf{x} = \mathbf{W}\mathbf{y} \implies \mathbf{y} = \mathbf{W}^{-1}\mathbf{x} = \frac{2}{n}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \xi & \xi^2 & \cdots & \xi^{n-1} \\ 1 & \xi^2 & \xi^4 & \cdots & \xi^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi^{n-1} & \xi^{n-2} & \cdots & \xi \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

$$\xi = e^{-2\pi i/n} = \overline{\omega}$$

# The New Coordinates

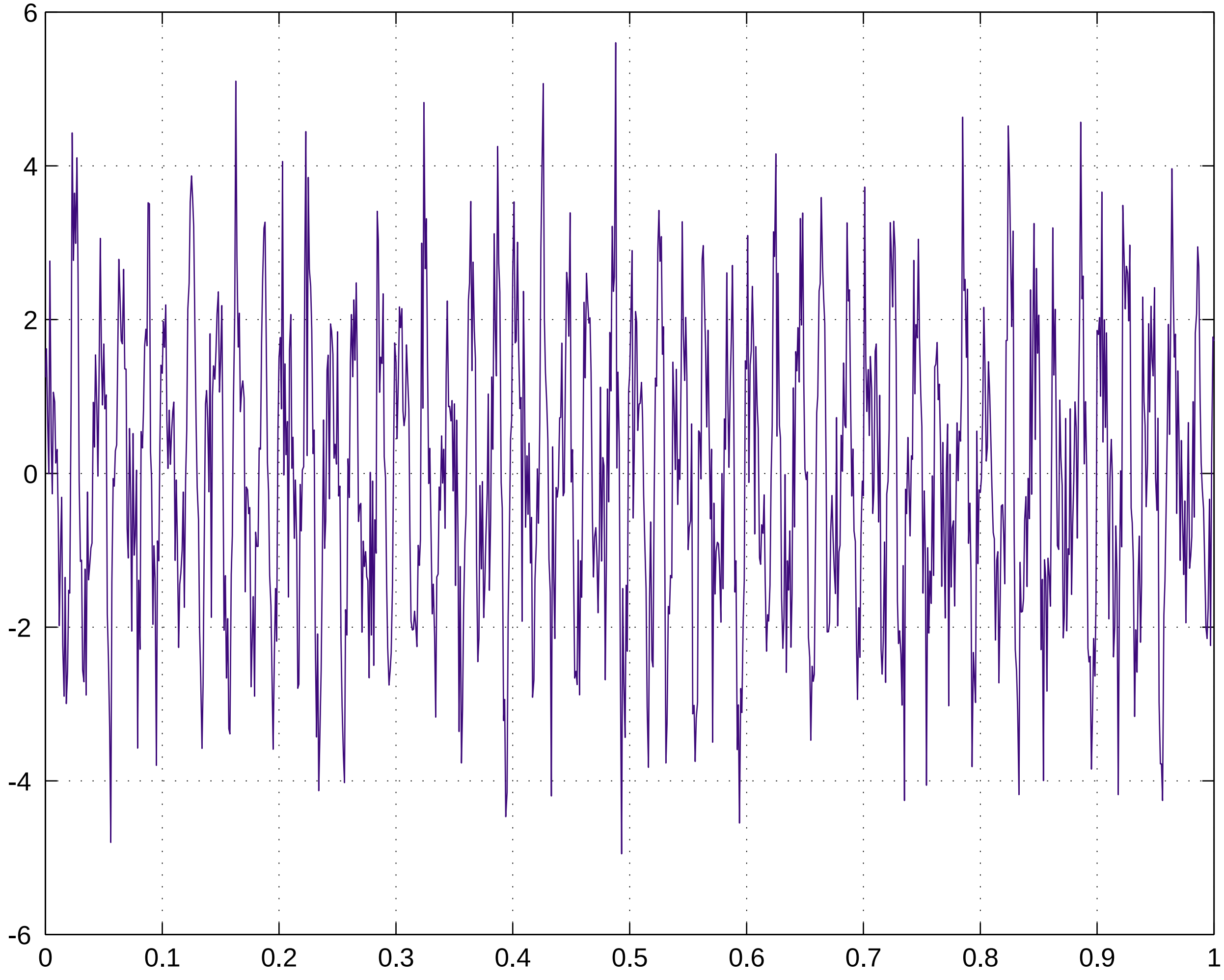


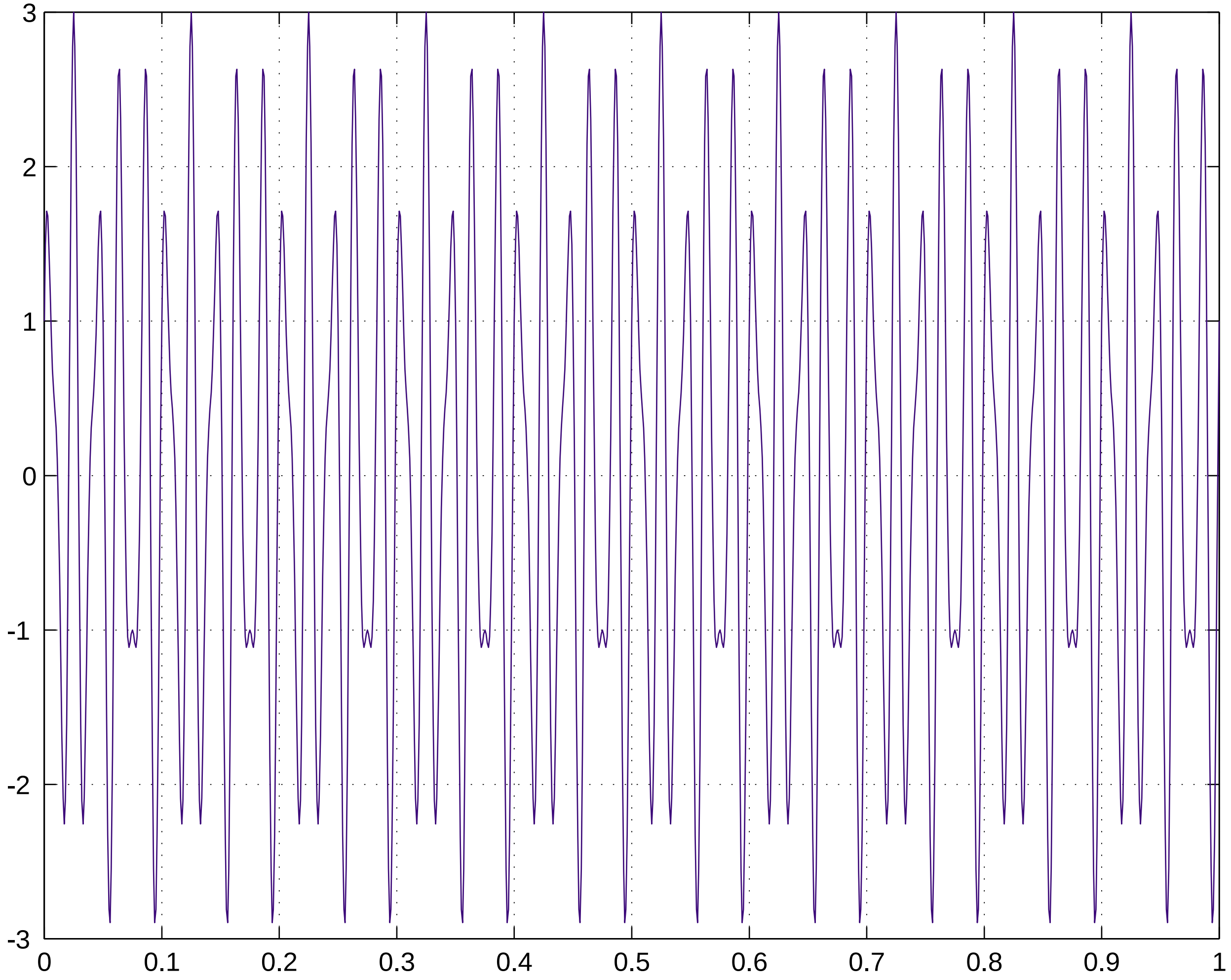

$$\mathbf{y} = (\mathbf{e}_{80} + \mathbf{e}_{432}) - 2\mathrm{i}(\mathbf{e}_{50} - \mathbf{e}_{462}) + \varepsilon$$
$$= (\mathbf{e}_{80} + \mathbf{e}_{n-80}) - 2\mathrm{i}(\mathbf{e}_{50} - \mathbf{e}_{n-50}) + \varepsilon \qquad (n = 512)$$

$$\mathbf{x} = \mathbf{W}\mathbf{y} = (\mathbf{W}_{80} + \mathbf{W}_{n-80}) - 2\mathrm{i}(\mathbf{W}_{50} - \mathbf{W}_{n-50}) + \varepsilon$$
$$= \cos 2\pi 80\mathbf{t} + 2\sin 2\pi 50\mathbf{t} + \varepsilon$$

# Original



# Cleaned & Compressed




$$\cos 2\pi 80t + 2 \sin 2\pi 50t$$

# The DFT Game

## Matrix–Vector Product

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

## Simple in Theory but Difficult in Practice

- Must do it *FAST!*

## Need For Speed $\Longrightarrow$ Matrix Factorizations $\Longrightarrow$ FFT

- $\mathbf{F}_n = \mathbf{B}_n(\mathbf{I}_2 \otimes \mathbf{F}_{n/2})\mathbf{P}_n \qquad \mathbf{B}_n = \begin{bmatrix} \mathbf{I}_{n/2} & \mathbf{D}_{n/2} \\ \mathbf{I}_{n/2} & -\mathbf{D}_{n/2} \end{bmatrix} \qquad \mathbf{D}_{n/2} = \begin{bmatrix} 1 & & & \\ & \xi & & \\ & & \xi^2 & \\ & & & \ddots \end{bmatrix}$
- FFT changes $n^2$ flop requirement into $(n/2)\log_2 n$

"The most valuable numerical algorithm in our lifetime."

—G. Strang, Bulletin of the AMS, April, 1993.

# Things Have Changed

- “For engineers and social and physical scientists, linear algebra now fills a place that is often more important than calculus.”
- “It is partly [due to] the move from analog to digital; functions are replaced by vectors.”
- “My generation of students, and certainly my teachers, did not see this change coming.”

—Gilbert Strang

FROM THE AMERICAN SCIENTIST

APRIL, 1994.

# Back To IR

## Almost the Same Problem

- Evaluate $\mathbf{q}^T\mathbf{A}$ fast

## Data is Now the Term-Doc Matrix in Standard Coordinates

- $\mathbf{A} = \sum_{i,j} \langle \mathbf{E}_{ij} | \mathbf{A} \rangle \, \mathbf{E}_{ij}$ $\qquad \mathbf{E}_{ij} = \mathbf{e}_i \mathbf{e}_j^T$ $\qquad \langle \mathbf{E}_{ij} | \mathbf{A} \rangle = a_{ij}$
- $\langle \mathbf{X} | \mathbf{Y} \rangle = trace\left(\mathbf{X}^{\mathsf{T}}\mathbf{Y}\right)$ $\qquad \|\mathbf{X}\| = \langle \mathbf{X} | \mathbf{X} \rangle^{1/2}$ (Frobenius Norm)

## Seek New o.n. Basis That Squeezes & Cleans

- $\mathbf{A} = \sum_{i=1}^{r} \langle \mathbf{Z}_i | \mathbf{A} \rangle \, \mathbf{Z}_i$

## Think Matrix Factorizations $\Longrightarrow$ SVD $\Longrightarrow$ $\mathbf{A} = \mathbf{UDV}^T$

- $\mathbf{A} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \sum_{i=1}^{r} \sigma_i \mathbf{Z}_i, \qquad \mathbf{Z}_i = \mathbf{u}_i \mathbf{v}_i^T, \qquad \langle \mathbf{Z}_i | \mathbf{Z}_j \rangle = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$
- $\sigma_i = \langle \mathbf{Z}_i | \mathbf{A} \rangle$ = the amount of $\mathbf{A}$ in direction of $\mathbf{Z}_i$

# Same As Before

## Assume Nondirectional Noise or Uncertainty

- Drop small $\sigma_i$'s and use $\mathbf{A} \approx \widetilde{\mathbf{A}} = \sum_{i=1}^{k} \sigma_i \mathbf{Z}_i$
- Lose only small part of relevance
- Lose larger proportion of noise or uncertainty

## Be Liberal in Dropping $\sigma_i$'s

- Numerical accuracy not important — 2 or 3 significant digits

## New Query Matching Strategy

- Normalize
  - $\mathbf{q} \leftarrow \mathbf{q}/\|\mathbf{q}\|$
  - $\widetilde{\mathbf{A}} \leftarrow \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \mathbf{D} = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \widetilde{\mathbf{v}}_i^T$
- Compare query to each document
  - $(\delta_1, \delta_2, \ldots, \delta_n) = \mathbf{q}^T \widetilde{\mathbf{A}} = \sum_{i=1}^{k} \sigma_i (\mathbf{q}^T \mathbf{u}_i) \widetilde{\mathbf{v}}_i^T$

# Pros & Cons

## Advantages

- Compression
  - **A** is replaced by only a few sing values and sing vectors
  - They are determined & normalized only once
- $SPEED!$
  - Each query requires only a few inner products

$$\mathbf{q}^T\widetilde{\mathbf{A}}_{m\times n} = \sum_{i=1}^{k} \sigma_i(\mathbf{q}^T\mathbf{u}_i)\widetilde{\mathbf{v}}_i^T$$

- Latent semantic associations are made
  - Relevant docs not found by direct matching show up

## Disadvantages

- Adding & deleting docs (updating & downdating SVD) difficult
- Determining optimal $k$ is not easy (empirical tuning required)

# Variations

## Projected Query

- First project the query onto the document space

$$\widetilde{\mathbf{q}} = \mathbf{P}_{R(\mathbf{A})}\mathbf{q} = \sum_{j=1}^{r} \mathbf{u}_j \mathbf{u}_j^T \mathbf{q}$$

- Or, better yet, use truncated projection

$$\widetilde{\mathbf{q}} = \mathbf{P}_{R\left(\widetilde{\mathbf{A}}\right)}\mathbf{q} = \sum_{j=1}^{k} \mathbf{u}_j \mathbf{u}_j^T \mathbf{q}$$

- Notice
  - $\widetilde{\mathbf{q}}^T \widetilde{\mathbf{A}} = \sum_{i=1}^{k} \sigma_i (\widetilde{\mathbf{q}}^T \mathbf{u}_i) \widetilde{\mathbf{v}}_i^T = \sum_{i=1}^{k} \sigma_i \left( (\sum_{j=1}^{k} \mathbf{q}^T \mathbf{u}_j^T \mathbf{u}_j) \mathbf{u}_i \right) \widetilde{\mathbf{v}}_i^T = \mathbf{q}^T \widetilde{\mathbf{A}}$
  - $\|\widetilde{\mathbf{q}}\| \leq \|\mathbf{q}\|$
  - $\cos \widetilde{\theta}_i \geq \cos \theta_i$ (more documents are deemed relevant)

# Other Factorizations

## DFT — FFT

- No compression — no oscillatory components

## Haar Transform $\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ $\quad \mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix}$

- $\mathbf{H}_n = (\mathbf{I}_2 \otimes \mathbf{H}_{n/2})\mathbf{P}_n \begin{bmatrix} \mathbf{H}_{n/2} & \\ & \mathbf{I}_{n/2} \end{bmatrix} \quad \Rightarrow \quad \mathbf{H}_n\mathbf{x}$ is $Fast$ (if $n = 2^p$)
- Factor $\mathbf{A} = \mathbf{H}_m\mathbf{B}\mathbf{H}_n^T = \sum_{i,j} \beta_{ij}\mathbf{h}_i\mathbf{h}_j^T$ ( $\mathbf{h}$'s only use $-1$, $0$, or $1$)
  - More than a few $\beta_{ij}$'s may be needed
  - Needs padding if $m$ or $n$ not a power of 2

## Semidiscrete Decomposition (T. Kolda and D. O'Leary, 1998)

- Approximate $\mathbf{A} \approx \sum_{i=1}^{k} \alpha_i\mathbf{x}_i\mathbf{y}_j$ $\quad \mathbf{x}_i$ and $\mathbf{y}_j$ only use $-1$, $0$, or $1$

## Other Wavelet Transforms?

# Using Centroids

## Document Clusters

- Assume $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \cdots\ \mathbf{a}_n]$ (normalized) represents a cluster
- Mean: $\mathbf{m} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{a}_i = \frac{\mathbf{Ae}}{n}$ where $\mathbf{e}^T = (1, 1, \ldots, 1)$
- Centroid vector: $\mathbf{c} = \mathbf{m}/\|\mathbf{m}\| = \mathbf{Ae}/\|\mathbf{Ae}\|$

## Facts

- $$\min_{\mathbf{p}\geq 0, \|\mathbf{p}\|=1} \sum_{i=1}^{n} \cos\theta(\mathbf{a}_i, \mathbf{p}) = \sum_{i=1}^{n} \cos\theta(\mathbf{a}_i, \mathbf{c}).$$
- i.e., $\mathbf{c}$ is the closest vector to the cluster
- $\mathbf{c}^T\mathbf{A} \approx \mathbf{e}^T \qquad (\mathbf{c}^T\mathbf{a}_i = \cos\theta_i \approx 1)$

## Reflectors With Specified Unit Column — Say $\mathbf{c}$

- $\mathbf{L} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^T/(\mathbf{u}^T\mathbf{u}) = [\mathbf{c} \mid \mathbf{X}]$ where $\mathbf{u} = \mathbf{c} \pm \mathbf{e}_1$
- $\mathbf{L} = \mathbf{L}^T = \mathbf{L}^{-1}$

# Orthogonal Factorization

## Specify $\mathbf{c}$ and $\mathbf{e}/\sqrt{n}$ as First Columns

- $\mathbf{L} = [\mathbf{c} \mid \mathbf{X}]$ and $\mathbf{R} = [\mathbf{e}/\sqrt{n} \mid \mathbf{Y}]$

$$\mathbf{LAR} = \mathbf{L}^T\mathbf{AR} = \begin{bmatrix} \mathbf{c}^T\mathbf{Ae}/\sqrt{n} & \mathbf{c}^T\mathbf{AY} \\ \mathbf{X}^T\mathbf{Ae}/\sqrt{n} & \mathbf{X}^T\mathbf{AY} \end{bmatrix} \approx \begin{bmatrix} \mathbf{e}^T\mathbf{e}/\sqrt{n} & \mathbf{e}^T\mathbf{Y} \\ \mathbf{X}^T\mathbf{c}\|\mathbf{Ae}\|/\sqrt{n} & \mathbf{X}^T\mathbf{AY} \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{n} & 0 \\ 0 & \mathbf{X}^T\mathbf{AY} \end{bmatrix} = \begin{bmatrix} \sqrt{n} & 0 \\ 0 & \mathbf{B} \end{bmatrix}$$

## A New Factorization

- $\mathbf{A} = \mathbf{LDR} = \mathbf{LDR}^T = \mathbf{ce}^T + \sum_{i,j \neq 1} \beta_{ij}\mathbf{x}_i\mathbf{y}_j^T$

## Truncate To Compress & Clean

## Break Entire Collection Into Clusters

# To Learn More

## Books

- Matrix Analysis and Applied Linear Algebra, C. D. Meyer, SIAM, 2000.
- Understanding Search Engines; Mathematical Modeling and Text Retrieval, M. W. Berry and M. Browne, SIAM, 1999.
- Introduction to Modern Information Retrieval, G. Salton and M. McGill, McGraw–Hill, 1983.

## Papers


- M. W. Berry, Z. Drmac, and E. R. Jessup, Matrices, vector spaces, and information retrieval, SIAM Rev., 41(1999), pp.335-362.
- M. W. Berry, S. T. Dumas, and G. W. O'Brien, Using linear algebra for intelligent information retrieval, U. Tenn. Comp. Sci. Report CS-94-270, Dec, 1994.
- I. S. Dhillon and D. S. Modha, Concept decompositions for large sparse text data using clustering, IBM Research Report RJ 10147 (95022), July 8, 1999–declassified on March 13, 2000, to appear in Machine Learning.


## URLs

- http://www.cs.utk.edu/~ lsi/
- http://lsi.research.telcordia.com/
- http://lsa.colorado.edu/
- http://www.searchenginewatch.com/resources/index.html