Information Retrieval

and

Web Search

Amy Langville

Carl Meyer

Department of Mathematics
North Carolina State University
Raleigh, NC

MAA–Meredith 3/11/2005

Outline

Part 1: Traditional IR

- Vector Space Model (1960s and 1970s)
- Latent Semantic Indexing (1990s)
- Other VSM decompositions (1990s)
- Nonnegative Matrix Factorization (2000)

Part 2: Web IR

-

Vector Space Model (1960s and 1970s)

Gerard Salton's Information Retrieval System

SMART: System for the Mechanical Analysis and Retrieval of Text

(Salton's Magical Automatic Retriever of Text)

- turn n textual documents into n document vectors $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n$
- create term-by-document matrix $\mathbf{A}_{m \times n} = [\, \mathbf{d}_1 | \mathbf{d}_2 | \cdots | \mathbf{d}_n \,]$
- to retrieve info., create query vector $\mathbf{q}$, which is a pseudo-doc

Vector Space Model (1960s and 1970s)

Gerard Salton's Information Retrieval System

SMART: System for the Mechanical Analysis and Retrieval of Text

(Salton's Magical Automatic Retriever of Text)

- turn n textual documents into n document vectors $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n$
- create term-by-document matrix $\mathbf{A}_{m \times n} = [\, \mathbf{d}_1 | \mathbf{d}_2 | \cdots | \mathbf{d}_n \,]$
- to retrieve info., create query vector $\mathbf{q}$, which is a pseudo-doc

GOAL: find doc. $\mathbf{d}_i$ closest to $\mathbf{q}$

— angular cosine measure used: $\delta_i = cos\,\theta_i = \mathbf{q}^T \mathbf{d}_i / (\|\mathbf{q}\|_2 \|\mathbf{d}_i\|_2)$

Example from Berry's book

Terms	Documents
T1: Bab(y,ies,y's)	D1: Infant & Toddler First Aid
T2: Child(ren's)	D2: Babies & Children's Room (For Your Home)
T3: Guide	D3: Child Safety at Home
T4: Health	D4: Your Baby's Health & Safety : From Infant to Toddler
T5: Home	D5: Baby Proofing Basics
T6: Infant	D6: Your Guide to Easy Rust Proofing
T7: Guide	D7: Beanie Babies Collector's Guide
T8: Safety	
T9: Toddler	

Example from Berry's book

Terms

T1: Bab(y,ies,y's)
T2: Child(ren's)
T3: Guide
T4: Health
T5: Home
T6: Infant
T7: Guide
T8: Safety
T9: Toddler

Documents

D1: Infant & Toddler First Aid
D2: Babies & Children's Room (For Your Home)
D3: Child Safety at Home
D4: Your Baby's Health & Safety : From Infant to Toddler
D5: Baby Proofing Basics
D6: Your Guide to Easy Rust Proofing
D7: Beanie Babies Collector's Guide

$$\mathbf{A} = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \end{array} \begin{array}{c} \begin{array}{ccccccc} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{\delta} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \\ \delta_7 \end{bmatrix} = \begin{bmatrix} 0 \\ .5774 \\ 0 \\ .8944 \\ .7071 \\ 0 \\ .7071 \end{bmatrix}$$

VSM Performance

Measuring Performance

- Precision = $\left[\frac{\#\ \text{REL. DOCS RETRIEVED}}{\#\ \text{DOCS RETRIEVED}}\right]$ EX: $3/10$
- Recall = $\left[\frac{\#\ \text{REL. DOCS RETRIEVED}}{\#\ \text{REL. DOCS}}\right]$ EX: $3/7$
- Time
  - normalize cols of **A** and **q** to speed cosine computation
  - now relevancy vector $\delta = \mathbf{q}^T\mathbf{A}$ (just 1 V-M mult. at query time)

VSM Performance

Measuring Performance

- Precision = $\left[\frac{\#\ \text{REL. DOCS RETRIEVED}}{\#\ \text{DOCS RETRIEVED}}\right]$
- Recall = $\left[\frac{\#\ \text{REL. DOCS RETRIEVED}}{\#\ \text{REL. DOCS}}\right]$
- Time
  - normalize cols of **A** and **q** to speed cosine computation
  - now relevancy vector $\boldsymbol{\delta} = \mathbf{q}^T\mathbf{A}$ (just 1 V-M mult. at query time)

Enhancing Performance

- angle cutoff value: $\delta_i \geq .7$ vs $\delta_i \geq .8$
- weighting elements of **A**: tf-idf, b-idf, etc.
- stemming, stoplisting, etc.

(Resource: Text to Matrix Generator `http://scgroup.hpclab.ceid.upatras.gr/scgroup/Projects/TMG/`)

(Resource: Porter Stemmer Demo `http://snowball.tartarus.org/demo.php`)

(Resource: VSM Demo `http://kt2.exp.sis.pitt.edu:8080/VectorModel/main.html`)

Strengths and Weaknesses of VSM

Strengths

- $\mathbf{A}$ is sparse
- $\mathbf{q}^T\mathbf{A}$ is fast and can be done in parallel
- relevance feedback: $\tilde{\mathbf{q}} = \delta_1\mathbf{d}_1 + \delta_3\mathbf{d}_3 + \delta_7\mathbf{d}_7$

Weaknesses

- synonyms and polysems—noise in $\mathbf{A}$
- decent performance
- basis vectors are standard basis vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m$, which are orthogonal $\Rightarrow$ independence of terms

VSM Resources

- Gerard Salton. Automatic information organization and retrieval. McGraw-Hill, 1968.
- Gerard Salton and Michael J. McGill. Introduction to modern information retrieval. McGraw-Hill, 1983.
- Gerard Salton. Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley, 1989.
- Michael W. Berry and Murray Browne. Understanding search engines: mathematical modeling and text retrieval. SIAM, 1999.
- Amy N. Langville. The Linear Algebra behind Search Engines. JOMA. `http://mac04-204ha.math.ncsu.edu/ langville/JOMA/JOMAIntro.html`, 2005.
- Michael W. Berry. LSI Website. `http://www.cs.utk.edu/ lsi/`

Latent Semantic Indexing (1990s)

Susan Dumais's improvement to VSM = LSI

Idea: use low-rank approximation to **A** to filter out noise

- Great Idea! 2 patents for Bell/Telcordia
  - Computer information retrieval using latent semantic structure. U.S. Patent No. 4,839,853, June 13, 1989.
  - Computerized cross-language document retrieval using latent semantic indexing. U.S. Patent No. 5,301,109, April 5, 1994.

  (Resource: USPTO `http://patft.uspto.gov/netahtml/srchnum.htm`)

SVD

$\mathbf{A}_{m\times n}$: rank r term-by-document matrix

- SVD: $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\,\mathbf{V}^T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$
- LSI: use $\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ in place of $\mathbf{A}$
- Why?
  - reduce storage when $k << r$
  - filter out uncertainty, so that performance on text mining tasks (e.g., query processing and clustering) improves

What's Really Happening?

Change of Basis

using truncated SVD $\mathbf{A}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T$

- Original Basis: docs represented in Term Space using Standard Basis $S = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\}$
- New Basis: docs represented in smaller Latent Semantic Space using Basis $B = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ (k<<min(m,n))

$$\overset{doc_1}{\underset{\substack{nonneg.\\entries}}{}}\begin{pmatrix} \vdots \\ \mathbf{A}_{*1} \\ \vdots \end{pmatrix}_{m\times 1} \approx \begin{bmatrix} \vdots \\ \mathbf{u}_1 \\ \vdots \end{bmatrix} \sigma_1 v_{11} + \begin{bmatrix} \vdots \\ \mathbf{u}_2 \\ \vdots \end{bmatrix} \sigma_2 v_{12} + \cdots + \begin{bmatrix} \vdots \\ \mathbf{u}_k \\ \vdots \end{bmatrix} \sigma_k v_{1k}$$

What's Really Happening?

Change of Basis

using truncated SVD $\mathbf{A}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T$

- Original Basis: docs represented in Term Space using Standard Basis $S = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\}$
- New Basis: docs represented in smaller Latent Semantic Space using Basis $B = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ (k<<min(m,n))

$$\underset{\substack{nonneg.\\entries}}{} \overset{doc_1}{\begin{pmatrix} \vdots \\ \mathbf{A}_{*1} \\ \vdots \end{pmatrix}}_{m\times 1} \approx \begin{bmatrix} \vdots \\ \mathbf{u}_1 \\ \vdots \end{bmatrix} \sigma_1 v_{11} + \begin{bmatrix} \vdots \\ \mathbf{u}_2 \\ \vdots \end{bmatrix} \sigma_2 v_{12} + \cdots + \begin{bmatrix} \vdots \\ \mathbf{u}_k \\ \vdots \end{bmatrix} \sigma_k v_{1k}$$

- still use angular cosine measure

$$\delta_i = cos\,\theta_i = \mathbf{q}^T \mathbf{d}_i / (\|\mathbf{q}\|_2 \|\mathbf{d}_i\|_2) = \mathbf{q}^T \mathbf{A}_k \mathbf{e}_i / (\|\mathbf{q}\|_2 \|\mathbf{A}_k \mathbf{e}_i\|_2)$$

$$= \mathbf{q}^T \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T \mathbf{e}_i / (\|\mathbf{q}\|_2 \|\boldsymbol{\Sigma}_k \mathbf{V}_k^T \mathbf{e}_i\|_2)$$

Properties of SVD

- basis vectors $\mathbf{u}_i$ are orthogonal
- u_{ij}, v_{ij} are mixed in sign

$$\underset{nonneg}{\mathbf{A}_k} = \underset{mixed}{\mathbf{U}_k} \; \underset{nonneg}{\Sigma_k} \; \underset{mixed}{\mathbf{V}_k^T}$$

- **U**, **V** are dense
- *uniqueness*—while there are many SVD algorithms, they all create the same (truncated) factorization
- of all rank-k approximations, $\mathbf{A}_k$ is optimal (in Frobenius norm)

$$\|\mathbf{A} - \mathbf{A}_k\|_F = \min_{rank(\mathbf{B}) \le k} \|\mathbf{A} - \mathbf{B}\|_F$$

A =

0	0.5774	0	0.4472	0.7071	0	0.7071
0	0.5774	0.5774	0	0	0	0
0	0	0	0	0	0.7071	0.7071
0	0	0	0.4472	0	0	0
0	0.5774	0.5774	0	0	0	0
0.7071	0	0	0.4472	0	0	0
0	0	0	0	0.7071	0.7071	0
0	0	0.5774	0.4472	0	0	0
0.7071	0	0	0.4472	0	0	0

A4 =

-0.0018	0.5958	-0.0148	0.4523	0.6974	0.0102	0.6974
-0.0723	0.4938	0.6254	0.0743	0.0121	-0.0133	0.0121
0.0002	-0.0067	0.0052	-0.0013	0.3569	0.7036	0.3569
0.1968	0.0512	0.0064	0.2179	0.0532	-0.0540	0.0532
-0.0723	0.4938	0.6254	0.0743	0.0121	-0.0133	0.0121
0.6315	-0.0598	0.0288	0.5291	-0.0008	0.0002	-0.0008
0.0002	-0.0067	0.0052	-0.0013	0.3569	0.7036	0.3569
0.2151	0.2483	0.4347	0.2262	-0.0359	0.0394	-0.0359
0.6315	-0.0598	0.0288	0.5291	-0.0008	0.0002	-0.0008

A5 =

-0.0018	0.5958	-0.0148	0.4523	0.6974	0.0102	0.6974
-0.0723	0.4938	0.6254	0.0743	0.0121	-0.0133	0.0121
0.0002	-0.0067	0.0052	-0.0013	0.0033	0.7036	0.7105
0.1968	0.0512	0.0064	0.2179	0.0532	-0.0540	0.0532
-0.0723	0.4938	0.6254	0.0743	0.0121	-0.0133	0.0121
0.6315	-0.0598	0.0288	0.5291	-0.0008	0.0002	-0.0008
0.0002	-0.0067	0.0052	-0.0013	0.7105	0.7036	0.0033
0.2151	0.2483	0.4347	0.2262	-0.0359	0.0394	-0.0359
0.6315	-0.0598	0.0288	0.5291	-0.0008	0.0002	-0.0008

A6 =

-0.0069	0.5915	-0.0126	0.4577	0.6975	0.0100	0.6975
0.0075	0.5619	0.5911	-0.0114	0.0105	-0.0109	0.0105
0.0024	-0.0048	0.0043	-0.0036	0.0033	0.7037	0.7104
0.0402	-0.0824	0.0736	0.3861	0.0563	-0.0586	0.0563
0.0075	0.5619	0.5911	-0.0114	0.0105	-0.0109	0.0105
0.7055	0.0033	-0.0030	0.4497	-0.0023	0.0024	-0.0023
0.0024	-0.0048	0.0043	-0.0036	0.7104	0.7037	0.0033
-0.0223	0.0457	0.5366	0.4811	-0.0312	0.0325	-0.0312
0.7055	0.0033	-0.0030	0.4497	-0.0023	0.0024	-0.0023

A7 =

-0.0000	0.5774	-0.0000	0.4472	0.7071	0.0000	0.7071

-0.0000	0.5774	0.5774	-0.0000	-0.0000	-0.0000	0.0000
-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	0.7071	0.7071
-0.0000	-0.0000	-0.0000	0.4472	-0.0000	0.0000	-0.0000
-0.0000	0.5774	0.5774	0.0000	-0.0000	-0.0000	0.0000
0.7071	0.0000	-0.0000	0.4472	0.0000	0.0000	0.0000
-0.0000	0.0000	-0.0000	-0.0000	0.7071	0.7071	0.0000
-0.0000	0.0000	0.5774	0.4472	-0.0000	-0.0000	0.0000
0.7071	0.0000	-0.0000	0.4472	0.0000	0.0000	0.0000

LSI Demos

- Telcordia LSI Demo: `http://lsi.research.telcordia.com/lsi-bin/lsiQuery`
- Netlib LSI Demo: `http://www.netlib.org/cgi-bin/lsiBook`

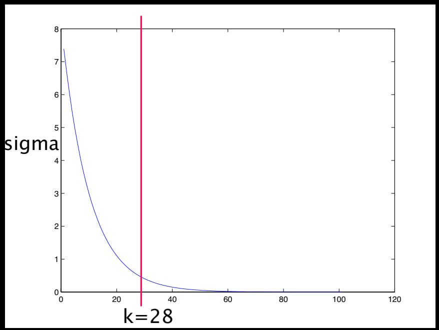
Strengths and Weaknesses of LSI

Strengths

- using $\mathbf{A}_k$ in place of $\mathbf{A}$ gives improved performance
- dimension reduction considers only essential components of term-by-document matrix, filters out noise
- best rank-k approximation

Weaknesses

- storage—$\mathbf{U}_k$ and $\mathbf{V}_k$ are usually completely dense
- interpretation of basis vectors $\mathbf{u}_i$ is impossible due to mixed signs
- good truncation point k is hard to determine
- orthogonality restriction



LSI Resources

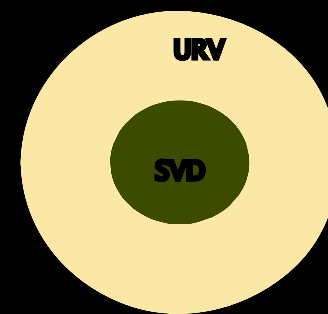
- Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien. Using Linear Algebra for Intelligent Information Retrieval. SIAM Review 37(4):573-595, 1995).
- Michael W. Berry, Z. Drmac, and Elizabeth R. Jessup. Matrices, Vector Spaces, and Information Retrieval. SIAM Review 41(2):335-362, 1999.
- Michael W. Berry and Murray Browne. Understanding search engines: mathematical modeling and text retrieval. SIAM, 1999.
- Amy N. Langville. The Linear Algebra behind Search Engines. JOMA. `http://mac04-204ha.math.ncsu.edu/ langville/JOMA/JOMAIntro.html`, 2005.
- Michael W. Berry. LSI Website. `http://www.cs.utk.edu/ lsi/`
- SVDPACK and SVDLIBC. Software for singular value decomposition.

  links at: `http://www.cs.utk.edu/ lsi/`

Other Low-Rank Approximations

- **QR** decomposition (see Berry et al. 1999 SIREV or Berry/Browne book)

- any **URV**^{T} factorization — Boeing's Donut Patent

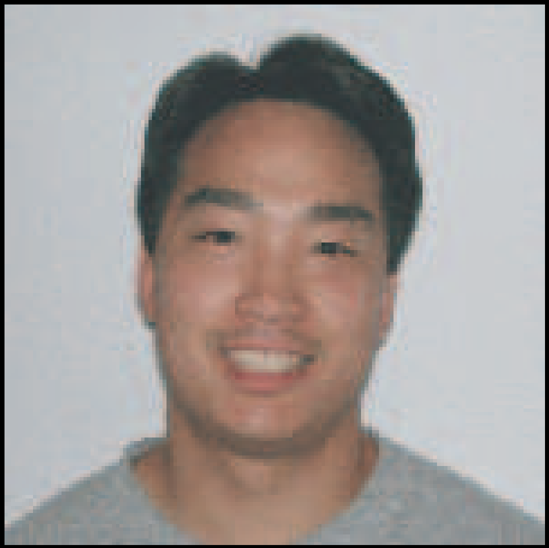


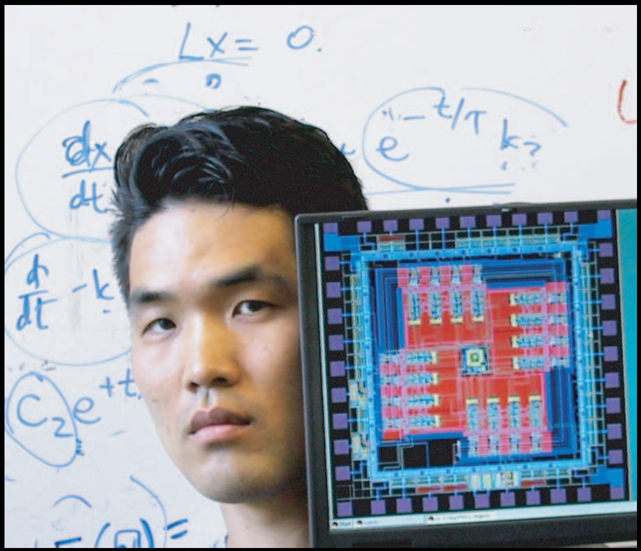
- Semidiscrete decomposition (SDD)

$$\mathbf{A}_k = \mathbf{X}_k \mathbf{D}_k \mathbf{Y}_k^T,$$ where $\mathbf{D}_k$ is diagonal, and elements of $\mathbf{X}_k, \mathbf{Y}_k \in \{-1, 0, 1\}$.

— Resource: Kolda/O'Leary C and Matlab Code `http://www.cs.umd.edu/ oleary/SDDPACK/`

Nonnegative Matrix Factorization (2000)





Daniel Lee and Sebastian Seung's Nonnegative Matrix Factorization

Idea: use low-rank approximation with nonnegative factors to improve LSI

$$\underset{nonneg}{\mathbf{A}_k} = \underset{mixed}{\mathbf{U}_k} \underset{nonneg}{\Sigma_k} \underset{mixed}{\mathbf{V}_k^T}$$

$$\underset{nonneg}{\mathbf{A}_k} = \underset{nonneg}{\mathbf{W}_k} \underset{nonneg}{\mathbf{H}_k}$$

Better Basis for Text Mining

Change of Basis

using NMF $\mathbf{A}_k = \mathbf{W}_k\mathbf{H}_k$, where $\mathbf{W}_k, \ \mathbf{H}_k \geq \mathbf{0}$

- Use of NMF: replace $\mathbf{A}$ with $\mathbf{A}_k = \mathbf{W}_k\mathbf{H}_k$ $\quad(\mathbf{W}_k = [\,\mathbf{w}_1|\mathbf{w}_2|\ldots|\mathbf{w}_k\,])$
- New Basis: docs represented in smaller Topic Space using Basis $B = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$ $\quad(k << \min(m,n))$

$$\underset{\textit{entries}}{\textit{nonneg.}}\overset{doc_1}{\begin{pmatrix} \vdots \\ \mathbf{A}_{*1} \\ \vdots \end{pmatrix}}_{m\times 1} \approx \begin{bmatrix} \vdots \\ \mathbf{w}_1 \\ \vdots \end{bmatrix} h_{11} + \begin{bmatrix} \vdots \\ \mathbf{w}_2 \\ \vdots \end{bmatrix} h_{21} + \cdots + \begin{bmatrix} \vdots \\ \mathbf{w}_k \\ \vdots \end{bmatrix} h_{k1}$$

Properties of NMF

- basis vectors $\mathbf{w}_i$ are not $\perp \Rightarrow$ can have overlap of topics
- can restrict **W**, **H** to be sparse
- $\mathbf{W}_k$, $\mathbf{H}_k \geq 0 \Rightarrow$ immediate interpretation (additive parts-based rep.)

  EX: large w_{ij}'s $\Rightarrow$ basis vector $\mathbf{w}_i$ is mostly about terms j

  EX: h_{i1} how much doc_1 is pointing in the "direction" of topic vector $\mathbf{w}_i$

$$\mathbf{A}_k\mathbf{e}_1 = \mathbf{W}_k\mathbf{H}_{*1} = \begin{bmatrix} \vdots \\ \mathbf{w}_1 \\ \vdots \end{bmatrix} h_{11} + \begin{bmatrix} \vdots \\ \mathbf{w}_2 \\ \vdots \end{bmatrix} h_{21} + \cdots + \begin{bmatrix} \vdots \\ \mathbf{w}_k \\ \vdots \end{bmatrix} h_{k1}$$

- NMF is algorithm-dependent: **W**, **H** not unique

NMF Literature

Papers report NMF is

- $\cong$ LSI for query processing

NMF Literature

Papers report NMF is

- $\cong$ LSI for query processing
- $\cong$ LSI for document clustering

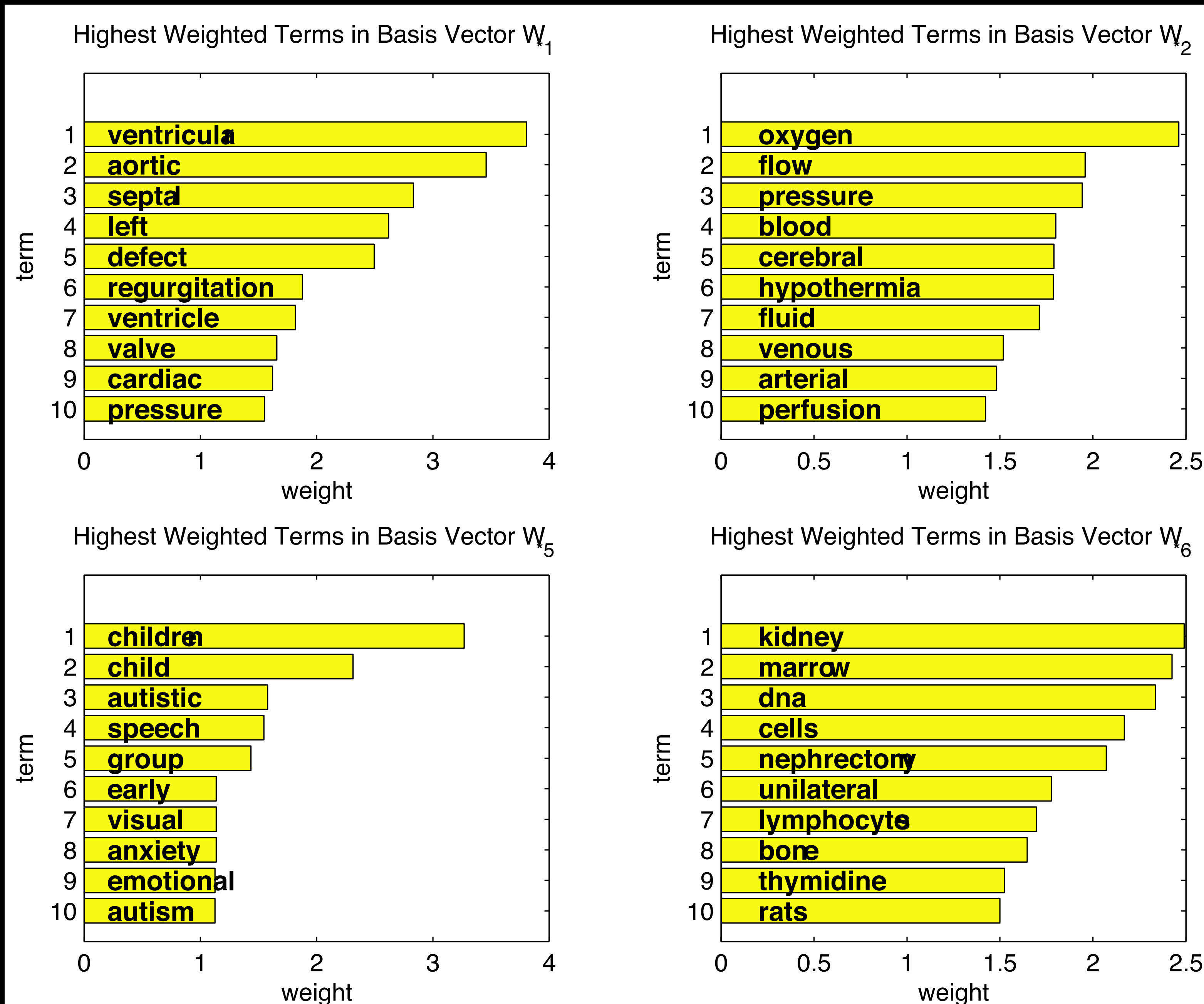
NMF Literature

Papers report NMF is

- $\cong$ LSI for query processing
- $\cong$ LSI for document clustering
- $>$ LSI for interpretation of elements of factorization

Interpretation of Basis Vectors

MED dataset ($k = 10$)



Interpretation of Basis Vectors

MED dataset ($k = 10$)

$$\textbf{doc}_5 \approx \overset{\mathbf{w}_9}{\begin{pmatrix} \text{fatty} \\ \text{glucose} \\ \text{acids} \\ \text{ffa} \\ \text{insulin} \\ \vdots \end{pmatrix}} .1646 + \overset{\mathbf{w}_6}{\begin{pmatrix} \text{kidney} \\ \text{marrow} \\ \text{dna} \\ \text{cells} \\ \text{nephr.} \\ \vdots \end{pmatrix}} .0103 + \overset{\mathbf{w}_7}{\begin{pmatrix} \text{hormone} \\ \text{growth} \\ \text{hgh} \\ \text{pituitary} \\ \text{mg} \\ \vdots \end{pmatrix}} .0045 + \cdots$$

NMF Literature

Papers report NMF is

- $\cong$ LSI for query processing
- $\cong$ LSI for document clustering
- $>$ LSI for interpretation of elements of factorization
- $>$ LSI potentially in terms of storage (sparse implementations)

NMF Literature

Papers report NMF is

- $\cong$ LSI for query processing
- $\cong$ LSI for document clustering
- $>$ LSI for interpretation of elements of factorization
- $>$ LSI potentially in terms of storage (sparse implementations)
- — NLP requires $O(kmn)$ computation per iteration, $\approx$ 10-15 iterations enough for convergence to local min

Computation of NMF

(Lee and Seung 2000)

Mean squared error objective function

$$\min \|\mathbf{A} - \mathbf{W}\mathbf{H}\|^2 \quad s.t. \quad \mathbf{W}, \mathbf{H} \geq 0$$

W = abs(randn(m,k));

H = abs(randn(k,n));

for i = 1 : maxiter

$\mathbf{H} = \mathbf{H}$.* $(\mathbf{W}^T\mathbf{A})$ ./ $(\mathbf{W}^T\mathbf{W}\mathbf{H} + 10^{-9})$;

$\mathbf{W} = \mathbf{W}$.* $(\mathbf{A}\mathbf{H}^T)$ ./ $(\mathbf{W}\mathbf{H}\mathbf{H}^T + 10^{-9})$;

end

Many parameters affect performance (k, obj. function, sparsity constraints, algorithm, etc.).

— NMF is not unique!

Strengths and Weaknesses of NMF

Strengths

- Great Interpretability
- Performance for query processing/clustering comparable to LSI
- Sparsity of factorization allows for significant storage savings
- Scalability good as k, m, n increase
- possibly faster computation time than SVD

Weaknesses

- Factorization is not unique $\Rightarrow$ dependency on algorithm and parameters
- Unable to reduce the size of the basis without recomputing the NMF

NMF Resources

- Daniel D. Lee and H. Sebastian Seung. Learning the Parts of Objects by Non-negative Matrix Factorization. Nature, 401:788, 1999.
- Farial Shahnaz, Michael Berry, Paul Pauca, and Robert Plemmons. Document Clustering using Nonnegative Matrix Factorization. Journal on Information Processing and Management, submitted 2004.
- Patrik O. Hoyer. NMF papers and Matlab code. `http://www.cs.helsinki.fi/u/phoyer/`
- Simon John Shepherd. nnmf() executable C file. `http://www.simonshepherd.supanet.com/nnmf.htm`