ALS Algorithms

for the

Nonnegative Matrix Factorization

in

Text Mining

Amy Langville

Carl Meyer

SAS NMF Day 6/9/2005

Outline

- Nonnegative Matrix Factorization replaces LSI
- Alternating Least Squares Algorithm
- Multiplicative Update Algorithms
- Our ALS Algorithms: ACLS and AHCLS

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 $k << r$ (but, not true in practice, since even though $\mathbf{A}$ is sparse, $\mathbf{u}_i$'s, $\mathbf{v}_i$'s are dense)
  - 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))

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

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)

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

- sequential buildup of essential components of **A**

  $\Rightarrow$ computing $\mathbf{A}_{100}$ means you also have $\mathbf{A}_k$ for $k < 100$

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

$$\mathop{\begin{pmatrix} \vdots \\ \mathbf{A}_{*1} \\ \vdots \end{pmatrix}}^{doc_1}_{m\times 1} \approx \begin{bmatrix} \vdots \\ \mathbf{w}_1 \\ \vdots \end{bmatrix} h_{11} + \begin{bmatrix} \vdots \\ \mathbf{w}_2 \\ \vdots \end{bmatrix} h_{21} + \cdots + \begin{bmatrix} \vdots \\ \mathbf{w}_k \\ \vdots \end{bmatrix} h_{k1}$$

nonneg. entries

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

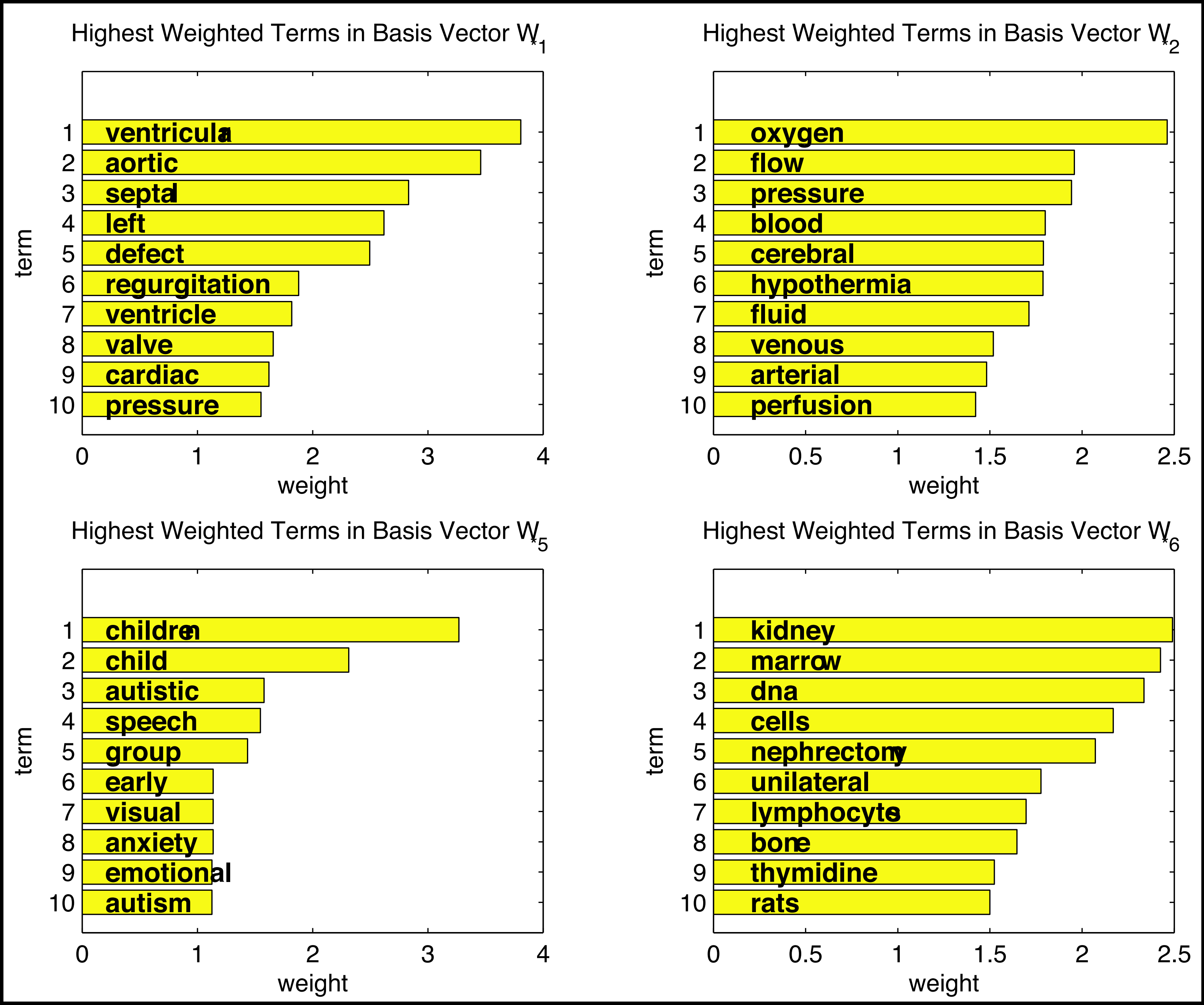
  EX: h_{i1} how much doc_1 is pointing in the "direction" of topic vector $\mathbf{w}_i$

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

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

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

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

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)
- — most NLP algorithms require $O(kmn)$ computation per iteration

Computation of NMF

(Lee and Seung 2000)

Mean squared error objective function

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

Nonlinear Optimization Problem

- convex in **W** or **H**, but not both $\Rightarrow$ tough to get global min
- huge # unknowns: mk for **W** and kn for **H**

  (EX: $\mathbf{A}_{70K \times 1K}$ and k=10 topics $\Rightarrow$ 800K unknowns)
- above objective is one of many possible
- convergence to local min only guaranteed for some algorithms

NMF Algorithms

- Alternating Least Squares
  - Paatero 1994
- Multiplicative update rules
  - Lee-Seung 2000
  - Hoyer 2002
- Gradient Descent
  - Hoyer 2004
  - Berry-Plemmons 2004

PMF Algorithm: Paatero & Tapper 1994

MEAN SQUARED ERROR—ALTERNATING LEAST SQUARES

$$\min \|\mathbf{A} - \mathbf{W}\mathbf{H}\|_F^2$$

$$s.t. \quad \mathbf{W}, \mathbf{H} \geq 0$$

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

for i = 1 : maxiter

LS for j = 1 : $\#docs$, solve

$$\min_{\mathbf{H}_{*j}} \|\mathbf{A}_{*j} - \mathbf{W}\mathbf{H}_{*j}\|_2^2$$

$$\text{s.t.} \quad \mathbf{H}_{*j} \geq 0$$

LS for j = 1 : $\#terms$, solve

$$\min_{\mathbf{W}_{j*}} \|\mathbf{A}_{j*} - \mathbf{W}_{j*}\mathbf{H}\|_2^2$$

$$\text{s.t.} \quad \mathbf{W}_{j*} \geq 0$$

end

ALS Algorithm

$\mathbf{W}$ = abs(randn(m,k));

for i = 1 : maxiter

LS solve matrix equation $\mathbf{W}^T\mathbf{W}\mathbf{H} = \mathbf{W}^T\mathbf{A}$ for $\mathbf{H}$

NONNEG $\mathbf{H} = \mathbf{H}. * (\mathbf{H} >= 0)$

LS solve matrix equation $\mathbf{H}\mathbf{H}^T\mathbf{W}^T = \mathbf{H}\mathbf{A}^T$ for $\mathbf{W}$

NONNEG $\mathbf{W} = \mathbf{W}. * (\mathbf{W} >= 0)$

end

ALS Summary

Pros

- \+ fast
- \+ works well in practice
- \+ speedy convergence
- \+ only need to initialize $\mathbf{W}^{(0)}$
- \+ 0 elements not *locked*

Cons

- – no sparsity of **W** and **H** incorporated into mathematical setup
- – ad hoc nonnegativity: negative elements are set to 0
- – ad hoc sparsity: negative elements are set to 0
- – no convergence theory

Alternating LP

Alternating Least Squares (one column at a time)

$$\min_{\mathbf{H}_{*j}} \|\mathbf{A}_{*j} - \mathbf{W}\mathbf{H}_{*j}\|_2^2$$
$$\text{s.t. } \mathbf{H}_{*j} \geq 0$$

"Linear L1 minimization can be solved by LP"—Warren Sarle, SAS

Alternating Linear Programming

$$\min_{\mathbf{H}_{*j}} \|\mathbf{A}_{*j} - \mathbf{W}\mathbf{H}_{*j}\|_1{}^2$$
$$\text{s.t. } \mathbf{H}_{*j} \geq 0$$

becomes

$$\min_{\mathbf{H}_{*j},\mathbf{r}} \quad \mathbf{r}^T\mathbf{e}$$
$$\text{s.t. } -r_i \leq \mathbf{A}_{ij} - \mathbf{W}\mathbf{H}_{*j} \leq r_i, \quad i = 1, \ldots, m$$
$$\mathbf{H}_{*j} \geq 0$$

Alternating LP

Considering entire matrix **H** at once...

Alternating Least Squares

solve matrix equation $\mathbf{W}^T\mathbf{W}\mathbf{H} = \mathbf{W}^T\mathbf{A}$ for **H**

($\mathbf{W}^T\mathbf{W}$ is small $k \times k$ matrix.)

Alternating Linear Programming

$$\begin{aligned} \min_{\mathbf{H},\mathbf{R}} \quad & \mathbf{e}^T\mathbf{R}\mathbf{e} \\ \text{s.t.} \quad & -\mathbf{R} \leq \mathbf{A} - \mathbf{W}\mathbf{H} \leq \mathbf{R} \\ & \mathbf{H}, \mathbf{R} \geq 0 \end{aligned}$$

(**H** is $k \times n$ and **R** is $m \times n$.)

— ALP has mn more variables than ALS

— not easy to add in sparsity rewards

\+ no ad-hoc enforcement of nonnegativity

NMF Algorithm: Lee and Seung 2000

Mean Squared Error objective function

$$\min \|\mathbf{A} - \mathbf{WH}\|_F^2$$

$$s.t. \quad \mathbf{W}, \mathbf{H} \geq 0$$

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

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

for i = 1 : maxiter

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

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

end

(proof of convergence to local min based on E-M convergence proof)

(objective function tails off after 50-100 iterations)

NMF Algorithm: Lee and Seung 2000

Divergence objective function

$$\min \sum_{i,j} (\mathbf{A}_{ij} \log \frac{\mathbf{A}_{ij}}{[\mathbf{WH}]_{ij}} - \mathbf{A}_{ij} + [\mathbf{WH}]_{ij})$$

$$s.t. \quad \mathbf{W}, \mathbf{H} \geq 0$$

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

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

for i = 1 : maxiter

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

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

end

(proof of convergence to local min based on E-M convergence proof)

(objective function tails off after 50-100 iterations)

Multiplicative Update Summary

Pros

+ convergence theory: guaranteed to converge to local min, but possibly poor local min
+ good initialization $\mathbf{W}^{(0)}, \mathbf{H}^{(0)}$ speeds convergence and gets to better local min

Cons

– good initialization $\mathbf{W}^{(0)}, \mathbf{H}^{(0)}$ speeds convergence and gets to better local min
– slow: many M-M multiplications at each iteration
– hundreds/thousands of iterations until convergence
– no sparsity of **W** and **H** incorporated into mathematical setup
– 0 elements *locked*

Multiplicative Update and Locking

During iterations of mult. update algorithms, once an element in **W** *or* **H** *becomes 0, it can never become positive.*

- Implications for **W**: In order to improve objective function, algorithm can only take terms out, not add terms, to topic vectors.
- Very inflexible: once algorithm starts down a path for a topic vector, it must continue in that vein.
- ALS-type algorithms do not *lock* elements, greater flexibility allows them to escape from path heading towards poor local min

Sparsity Measures

- Berry et al. $\|\mathbf{x}\|_2^2$

- Hoyer $spar(\mathbf{x}_{n\times 1}) = \frac{\sqrt{n}-\|\mathbf{x}\|_1/\|\mathbf{x}\|_2}{\sqrt{n}-1}$

- Diversity measure $E^{(p)}(\mathbf{x}) = \sum_{i=1}^{n} |x_i|^p,\ 0 \le p \le 1$

  $E^{(p)}(\mathbf{x}) = -\sum_{i=1}^{n} |x_i|^p,\ p < 0$

  Rao and Kreutz-Delgado: algorithms for minimizing $E^{(p)}(\mathbf{x})$ s.t. $\mathbf{Ax} = \mathbf{b}$, but expensive iterative procedure

- Ideal $nnz(\mathbf{x})$ not continuous, NP-hard to use this in optim.

NMF Algorithm: Berry et al. 2004

GRADIENT DESCENT–CONSTRAINED LEAST SQUARES

W = abs(randn(m,k)); (scale cols of **W** to unit norm)

H = zeros(k,n);

for i = 1 : maxiter

CLS for j = 1 : $\#docs$, solve

$$\min_{\mathbf{H}_{*j}} \|\mathbf{A}_{*j} - \mathbf{W}\mathbf{H}_{*j}\|_2^2 + \lambda\|\mathbf{H}_{*j}\|_2^2$$

$$\text{s.t. } \mathbf{H}_{*j} \geq \mathbf{0}$$

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

end

NMF Algorithm: Berry et al. 2004

GRADIENT DESCENT–CONSTRAINED LEAST SQUARES

W = abs(randn(m,k)); (scale cols of **W** to unit norm)

H = zeros(k,n);

for i = 1 : maxiter

CLS for j = 1 : $\#docs$, solve

$$\min_{\mathbf{H}_{*j}} \|\mathbf{A}_{*j} - \mathbf{W}\mathbf{H}_{*j}\|_2^2 + \lambda \|\mathbf{H}_{*j}\|_2^2$$

$$\text{s.t. } \mathbf{H}_{*j} \geq \mathbf{0}$$

solve for **H**: $(\mathbf{W}^T\mathbf{W} + \lambda\,\mathbf{I})\,\mathbf{H} = \mathbf{W}^T\mathbf{A}$; (small matrix solve)

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

end

(objective function tails off after 15-30 iterations)

Berry et al. 2004 Summary

Pros

+ fast: less work per iteration than most other NMF algorithms
+ fast: small # of iterations until convergence
+ sparsity parameter for **H**

Cons

– 0 elements in **W** are *locked*
– no sparsity parameter for **W**
– ad hoc nonnegativity: negative elements in **H** are set to 0, could run lsqnonneg or snnls instead
– no convergence theory

Alternating Constrained Least Squares

If the very fast ALS works well in practice and the only NMF algorithms guaranteeing convergence to local min are slow multiplicative update rules, why not use ALS?

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

for i = 1 : maxiter

CLS for j = 1 : $\#docs$, solve

$$\min_{\mathbf{H}_{*j}} \|\mathbf{A}_{*j} - \mathbf{W}\mathbf{H}_{*j}\|_2^2 + \lambda_H \|\mathbf{H}_{*j}\|_2^2$$

$$\text{s.t. } \mathbf{H}_{*j} \geq 0$$

CLS for j = 1 : $\#terms$, solve

$$\min_{\mathbf{W}_{j*}} \|\mathbf{A}_{j*} - \mathbf{W}_{j*}\mathbf{H}\|_2^2 + \lambda_W \|\mathbf{W}_{j*}\|_2^2$$

$$\text{s.t. } \mathbf{W}_{j*} \geq 0$$

end

Alternating Constrained Least Squares

If the very fast ALS works well in practice and the only NMF algorithms guaranteeing convergence to local min are slow multiplicative update rules, why not use ALS?

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

for i = 1 : maxiter

CLS solve for **H**: $(\mathbf{W}^T\mathbf{W} + \lambda_H \mathbf{I})\,\mathbf{H} = \mathbf{W}^T\mathbf{A}$

NONNEG $\mathbf{H} = \mathbf{H}.*(\mathbf{H} >= 0)$

CLS solve for **W**: $(\mathbf{H}\mathbf{H}^T + \lambda_W \mathbf{I})\,\mathbf{W}^T = \mathbf{H}\mathbf{A}^T$

NONNEG $\mathbf{W} = \mathbf{W}.*(\mathbf{W} >= 0)$

end

ACLS Summary

Pros

+ fast: 6.6 sec vs. 9.8 sec (gd-cls)
+ works well in practice
+ speedy convergence
+ only need to initialize $\mathbf{W}^{(0)}$
+ 0 elements not *locked*
+ allows for sparsity in both **W** and **H**

Cons

– ad hoc nonnegativity: after LS, negative elements set to 0, could run lsqnonneg or snnls instead (doesn't improve accuracy much)
– no convergence theory

ACLS + spar(x)

Is there a better way to measure sparsity and still maintain speed of ACLS?

$$\text{spar}(\mathbf{x}_{n\times 1}) = \frac{\sqrt{n} - \|\mathbf{x}\|_1/\|\mathbf{x}\|_2}{\sqrt{n}-1} \quad \Leftrightarrow \quad ((1-\text{spar}(\mathbf{x}))\sqrt{n} + \text{spar}(\mathbf{x}))\|\mathbf{x}\|_2 - \|\mathbf{x}\|_1 = 0$$

$$(\text{spar}(\mathbf{W}_{j*}) = \alpha_W \text{ and } \text{spar}(\mathbf{H}_{*j}) = \alpha_H)$$

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

for i = 1 : maxiter

CLS for j = 1 : $\#docs$, solve

$$\min_{\mathbf{H}_{*j}} \|\mathbf{A}_{*j} - \mathbf{W}\mathbf{H}_{*j}\|_2^2 + \lambda_H(((1-\alpha_H)\sqrt{k} + \alpha_H)\|\mathbf{H}_{*j}\|_2^2 - \|\mathbf{H}_{*j}\|_1^2)$$

$$\text{s.t. } \mathbf{H}_{*j} \geq 0$$

CLS for j = 1 : $\#terms$, solve

$$\min_{\mathbf{W}_{j*}} \|\mathbf{A}_{j*} - \mathbf{W}_{j*}\mathbf{H}\|_2^2 + \lambda_W(((1-\alpha_W)\sqrt{k} + \alpha_W)\|\mathbf{W}_{j*}\|_2^2 - \|\mathbf{W}_{j*}\|_1^2)$$

$$\text{s.t. } \mathbf{W}_{j*} \geq 0$$

end

AHCLS

(spar($\mathbf{W}_{j*}$)=α_W and spar($\mathbf{H}_{*j}$)=α_H)

$\mathbf{W}$ = abs(randn(m,k));

for i = 1 : maxiter

$$\beta_H = ((1-\alpha_H)\sqrt{k} + \alpha_H)^2$$

CLS solve for **H**: $(\mathbf{W}^T\mathbf{W} + \lambda_H\beta_H\,\mathbf{I} - \lambda_H\mathbf{E})\,\mathbf{H} = \mathbf{W}^T\mathbf{A}$

NONNEG $\mathbf{H} = \mathbf{H}.*(\mathbf{H} >= 0)$

$$\beta_W = ((1-\alpha_W)\sqrt{k} + \alpha_W)^2$$

CLS solve for **W**: $(\mathbf{H}\mathbf{H}^T + \lambda_W\beta_W\,\mathbf{I} - \lambda_W\mathbf{E})\,\mathbf{W}^T = \mathbf{H}\mathbf{A}^T$

NONNEG $\mathbf{W} = \mathbf{W}.*(\mathbf{W} >= 0)$

end

AHCLS Summary

Pros

- \+ fast: 6.8 sec vs. 9.8 sec (gd-cls)
- \+ works well in practice
- \+ speedy convergence
- \+ only need to initialize $\mathbf{W}^{(0)}$
- \+ 0 elements not *locked*
- \+ allows for *more explicit* sparsity in both **W** and **H**

Cons

- – ad hoc nonnegativity: after LS, negative elements set to 0, could run lsqnonneg or snnls instead (doesn't improve accuracy much)
- – no convergence theory

Initialization of W

- Random initialization: done by most NMF algorithms
- Centroid initialization: shown by Wilds to converge to better local min., but expensive
- SVD-centroid initialization: run kmeans to cluster rows of $\mathbf{V}_{n \times k}$ from SVD and form cheap centroid decomposition. $\mathbf{W}^{(0)}$=Centroid vectors $\Rightarrow$ shown to converge to better local min.
- Random Acol initialization: works better than Random init., not as good as SVD-Centroid initialization. Very inexpensive.

EX: $(k{=}3)$ $\mathbf{W}^{(0)} = [\, \sum_{i \in \{1,4,10,12\}} \mathbf{A}_{*i} \,|\, \sum_{i \in \{2,3,9,11\}} \mathbf{A}_{*i} \,|\, \sum_{i \in \{5,6,7,8\}} \mathbf{A}_{*i} \,]$

Remaining Work

- Other Sparsity Measures
- Nonnegativity Enforcement
  — add negativity penalty to ALS objective
  ex: min error + density + negativity, where negativity=$\sum e^{-x_i}$
- Basis-constrained problem: user with dataset knowledge sets some basis vectors (cols of **W**), NMF algorithm must converge to solution that contains these vectors.
- Duality theory