Fast Newton-type Methods for the Least Squares Nonnegative Matrix Approximation Problem

Inderjit S. Dhillon

Department of Computer Sciences
The University of Texas at Austin

Joint work with Dongmin Kim and Suvrit Sra

Outline

Nonnegative matrix approximation (NNMA) problem:

- $A = [a_1, \ldots, a_N], \quad a_i \in \mathbb{R}_+^M$, is input nonnegative matrix.
- **Goal :** Approximate A by conic combinations of *nonnegative representative vectors* $b_1, \ldots, b_K$ such that

$$a_i \approx \sum_{j=1}^{K} b_j c_{ji}, \quad c_{ji} \geq 0, \quad b_j \geq 0,$$

$$\text{i.e.} \quad A \approx BC, \quad B, C \geq 0.$$

The quality of the approximation *BC* is

- Measured using an appropriate distortion function.
- For example, the Frobenius norm distortion or the Kullback-Leibler divergence.

In this presentation, we focus on the Frobenius norm distortion, which leads to the *least squares NNMA* problem.

$$\underset{B,C\geq 0}{\text{minimize}} \quad \mathscr{F}(B;C) = \frac{1}{2}\|A - BC\|_{\mathrm{F}}^2,$$

The quality of the approximation *BC* is

- Measured using an appropriate distortion function.
- For example, the Frobenius norm distortion or the Kullback-Leibler divergence.

In this presentation, we focus on the Frobenius norm distortion, which leads to the *least squares NNMA* problem.

$$\underset{B,C\geq 0}{\text{minimize}} \quad \mathscr{F}(B;C) = \tfrac{1}{2}\|A - BC\|_{\mathrm{F}}^{2},$$

Existing NNMA Algorithms

Basic Framework

- The NNMA objective function is not simultaneously convex in B and C.
- But is individually convex in B and in C.
- Most NNMA algorithms are iterative and perform an alternating optimization.

Basic Framework for NNMA algorithms

1. Initialize B^0 and/or C^0; set $t \leftarrow 0$.
2. Fix B^t and find C^{t+1} such that

$$\mathscr{F}(B^t, C^{t+1}) \leq \mathscr{F}(B^t, C^t),$$

3. Fix C^{t+1} and find B^{t+1} such that

$$\mathscr{F}(B^{t+1}, C^{t+1}) \leq \mathscr{F}(B^t, C^{t+1}),$$

4. Let $t \leftarrow t+1$, & repeat Steps 2 and 3 until convergence criteria are satisfied.

- The NNMA objective function is not simultaneously convex in B and C.
- But is individually convex in B and in C.
- Most NNMA algorithms are iterative and perform an alternating optimization.

Basic Framework for NNMA algorithms

1. Initialize B^0 and/or C^0; set $t \leftarrow 0$.
2. Fix B^t and find C^{t+1} such that

$$\mathscr{F}(B^t, C^{t+1}) \leq \mathscr{F}(B^t, C^t),$$

3. Fix C^{t+1} and find B^{t+1} such that

$$\mathscr{F}(B^{t+1}, C^{t+1}) \leq \mathscr{F}(B^t, C^{t+1}),$$

4. Let $t \leftarrow t+1$, & repeat Steps 2 and 3 until convergence criteria are satisfied.

- The Frobenius norm is the sum of Euclidean norms over columns.
- Optimization over B (or C) boils down to a series of *nonnegative least squares (NNLS)* problems.

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) = \tfrac{1}{2}\|Gx - h\|_2^2, \\ \text{subject to} \quad & x \geq 0. \end{aligned}$$

- *Exact* NNMA methods find a global optimum of this subproblem.
- *Inexact* NNMA methods roughly approximate it.

Exact Methods

- Based on NNLS algorithms:
  - Active set procedure [Lawson and Hanson(1974)]
  - FNNLS [Bro and Jong(1997)]
  - Interior-point gradient method [Merritt and Zhang(2005)]
- Projected gradient method [Lin(2005)].

Inexact Methods

- Multiplicative method [Lee and Seung(1999)].
- Alternating Least Squares (ALS) algorithm.
- "Projected Quasi-Newton" method [Zdunek and Cichocki(2006)].

Existing NNMA Algorithms
Examples

Exact Methods

- Based on NNLS algorithms:
  - Active set procedure [Lawson and Hanson(1974)]
  - FNNLS [Bro and Jong(1997)]
  - Interior-point gradient method [Merritt and Zhang(2005)]
- Projected gradient method [Lin(2005)].

Inexact Methods

- Multiplicative method [Lee and Seung(1999)].
- Alternating Least Squares (ALS) algorithm.
- “Projected Quasi-Newton” method [Zdunek and Cichocki(2006)].

Consider Lee & Seung's update rule.

$$[C]_{ij} \leftarrow [C]_{ij} \frac{[B^T A]_{ij}}{[B^T BC]_{ij}} \implies [C]_{ij} \leftarrow [C]_{ij} + \alpha_{ij} \left[[B^T A]_{ij} - [B^T BC]_{ij} \right],$$

where $\alpha_{ij} = \frac{[C]_{ij}}{[B^T BC]_{ij}}$.

- This is a **gradient descent update** with a special choice of step-size, α_{ij}.
- It can also be viewed as a **special case of projected gradient** method:

$$[C]_{ij} \leftarrow \mathscr{P}_+ \left[[C]_{ij} + \alpha_{ij} \left[[B^T A]_{ij} - [B^T BC]_{ij} \right] \right],$$

where $\mathscr{P}_+$ is the orthogonal projection onto the nonnegative orthant.

Consider Lee & Seung's update rule.

$$[C]_{ij} \leftarrow [C]_{ij} \frac{[B^T A]_{ij}}{[B^T BC]_{ij}} \implies [C]_{ij} \leftarrow [C]_{ij} + \alpha_{ij} \left[[B^T A]_{ij} - [B^T BC]_{ij} \right],$$

where $\alpha_{ij} = \frac{[C]_{ij}}{[B^T BC]_{ij}}$.

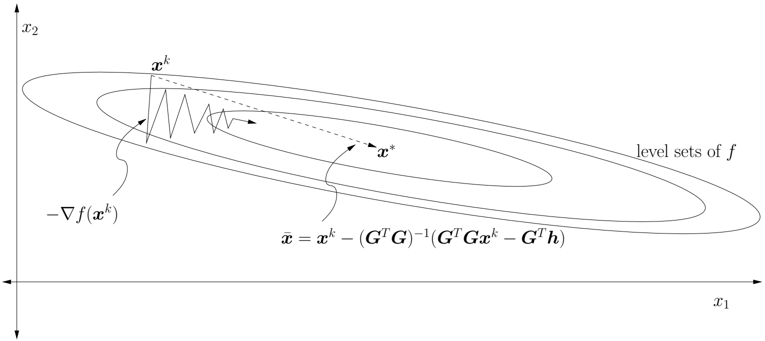
- This is a **gradient descent update** with a special choice of step-size, α_{ij}.
- It can also be viewed as a **special case of projected gradient** method:

$$[C]_{ij} \leftarrow \mathscr{P}_+ \left[[C]_{ij} + \alpha_{ij} \left[[B^T A]_{ij} - [B^T BC]_{ij} \right] \right],$$

where $\mathscr{P}_+$ is the orthogonal projection onto the nonnegative orthant.

Motivation for Newton-type Methods

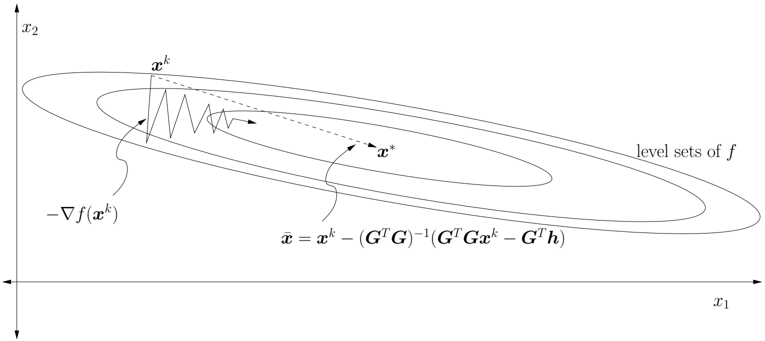
Fast Convergence



- Example of zigzagging phenomenon in gradient descent.
- Inner ellipses correspond to a smaller objective value of $f(x) = \|Gx - h\|_2^2$.

One iteration of the Newton-method gives the global optimum.

Motivation for Newton-type Methods
Fast Convergence



- Example of zigzagging phenomenon in gradient descent.
- Inner ellipses correspond to a smaller objective value of $f(x) = \|Gx - h\|_2^2$.

One iteration of the Newton-method gives the global optimum.

- Use Newton-type method for fast convergence.
- How can we handle the constraints?
  Combine with simplicity of projected gradient method, i.e.,

Combine orthogonal projection with Newton-type method!

The key in Newton-type method is to use a non-diagonal gradient scaling matrix H.

$$[C]_{ij} \leftarrow \mathscr{P}_{+}\left[[C]_{ij} + \alpha_{ij} H \left[[B^{T}A]_{ij} - [B^{T}BC]_{ij}\right]\right],$$

- Use Newton-type method for fast convergence.
- How can we handle the constraints?
  Combine with simplicity of projected gradient method, i.e.,

Combine orthogonal projection with Newton-type method!

The key in Newton-type method is to use a non-diagonal gradient scaling matrix H.

$$[C]_{ij} \leftarrow \mathscr{P}_{+}\left[[C]_{ij} + \alpha_{ij} H\left[[B^T A]_{ij} - [B^T BC]_{ij}\right]\right],$$

Handling Nonnegativity Constraints

Combining Projection with Newton-type Method

- Use Newton-type method for fast convergence.
- How can we handle the constraints?
  Combine with simplicity of projected gradient method, i.e.,

Combine orthogonal projection with Newton-type method!

The key in Newton-type method is to use a non-diagonal gradient scaling matrix H.

$$[C]_{ij} \leftarrow \mathscr{P}_+ \left[[C]_{ij} + \alpha_{ij} H \left[[B^T A]_{ij} - [B^T BC]_{ij} \right] \right],$$

- Use Newton-type method for fast convergence.
- How can we handle the constraints?
  Combine with simplicity of projected gradient method, i.e.,

Combine orthogonal projection with Newton-type method!

The key in Newton-type method is to use a non-diagonal gradient scaling matrix H.

$$[C]_{ij} \leftarrow \mathscr{P}_{+}\left[[C]_{ij} + \alpha_{ij} H \left[[B^T A]_{ij} - [B^T BC]_{ij}\right]\right],$$

- Use Newton-type method for fast convergence.
- How can we handle the constraints?
  Combine with simplicity of projected gradient method, i.e.,

Combine orthogonal projection with Newton-type method!

The key in Newton-type method is to use a non-diagonal gradient scaling matrix H.

$$[C]_{ij} \leftarrow \mathscr{P}_{+}\left[[C]_{ij} + \alpha_{ij} H\left[[B^T A]_{ij} - [B^T BC]_{ij}\right]\right],$$

Previous Attempts at Newton-type Methods for NNMA

Alternating Least Squares (ALS) and Zdunek & Cichocki's (ZC) Methods

- Consider ALS update for NNLS subproblem, $\min_{x \geq 0} = \frac{1}{2}\|Gx - h\|_2^2$.

$$x = \mathscr{P}_+[(G^TG)^{-1}G^Th], \text{ or equivalently,}$$

$$x = \mathscr{P}_+[x - (G^TG)^{-1}(G^TGx - G^Th)].$$

- where step-size $\alpha = 1$ and non-diagonal gradient scaling $H = (G^TG)^{-1}$.

- The ZC update is

$$x^{\text{new}} = \mathscr{P}_+[x^{\text{old}} - \alpha H(G^TGx^{\text{old}} - G^Th)],$$

- where $\alpha > 0$ and H is a **positive definite** matrix that approximates the inverse Hessian.

Previous Attempts at Newton-type Methods for NNMA

Alternating Least Squares (ALS) and Zdunek & Cichocki's (ZC) Methods

- Consider ALS update for NNLS subproblem, $\min_{x \geq 0} = \frac{1}{2}\|Gx - h\|_2^2$.

$$x = \mathscr{P}_+[(G^T G)^{-1} G^T h], \text{ or equivalently,}$$

$$x = \mathscr{P}_+[x - (G^T G)^{-1}(G^T Gx - G^T h)].$$

- where step-size $\alpha = 1$ and non-diagonal gradient scaling $H = (G^T G)^{-1}$.

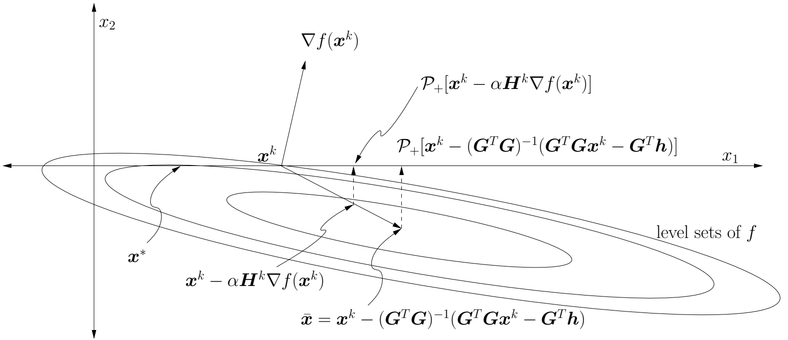
- The ZC update is

$$x^{\text{new}} = \mathscr{P}_+[x^{\text{old}} - \alpha H(G^T Gx^{\text{old}} - G^T h)],$$

- where $\alpha > 0$ and H is a **positive definite** matrix that approximates the inverse Hessian.

Previous Attempts at Newton-type Methods for NNMA

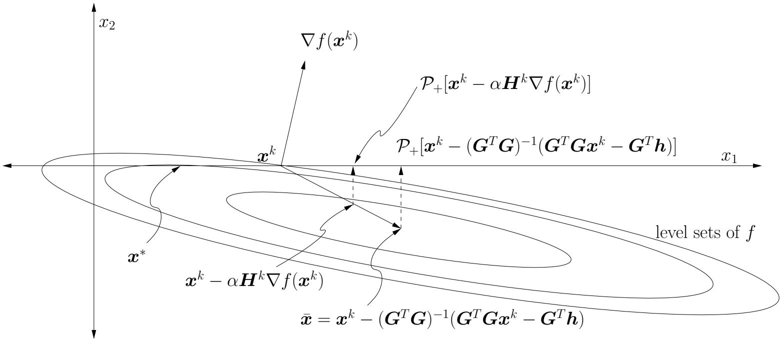
Difficulties



- Naïve Combination of projection step and non-diagonal gradient scaling does not guarantee convergence of the resulting algorithm.
- An iteration may actually lead to an increase of objective.

Previous Attempts at Newton-type Methods for NNMA

Difficulties



- Naïve Combination of projection step and non-diagonal gradient scaling does not guarantee convergence of the resulting algorithm.
- An iteration may actually lead to an increase of objective.

The active set :

- If active variables at the final solution are known in advance,
- Original problem can be solved as an **equality-constrained** problem.
- Equivalently one can solve an **unconstrained** sub-problem over inactive variables.

Projection :

- The projection step identifies the active variables at the current iteration.

Gradient :

- The gradient information gives a guideline to determine which variables *will not* be optimized at the next iteration.

The active set :

- If active variables at the final solution are known in advance,
- Original problem can be solved as an **equality-constrained** problem.
- Equivalently one can solve an **unconstrained** sub-problem over inactive variables.

Projection :

- The projection step identifies the active variables at the current iteration.

Gradient :

- The gradient information gives a guideline to determine which variables *will not* be optimized at the next iteration.

The active set :

- If active variables at the final solution are known in advance,
- Original problem can be solved as an **equality-constrained** problem.
- Equivalently one can solve an **unconstrained** sub-problem over inactive variables.

Projection :

- The projection step identifies the active variables at the current iteration.

Gradient :

- The gradient information gives a guideline to determine which variables *will not* be optimized at the next iteration.

New Newton-type Methods

Fixed Set

Divide variables into *Free* variables and *Fixed* variables.

- *Fixed* Set: Indices listing the entries of x^k that are held *fixed*.
- Definition: a set of indices

$$I^k = \left\{ i \middle| x_i^k = 0, [\nabla f(x^k)]_i > 0 \right\}.$$

- A **subset** of active variables at iteration k.
- Contains active variables that satisfy the KKT conditions.

Divide variables into *Free* variables and *Fixed* variables.

- *Fixed* Set: Indices listing the entries of x^k that are held *fixed*.
- Definition: a set of indices

$$I^k = \left\{ i \middle| x_i^k = 0,\, [\nabla f(x^k)]_i > 0 \right\}.$$

- A **subset** of active variables at iteration k.
- Contains active variables that satisfy the KKT conditions.

Divide variables into *Free* variables and *Fixed* variables.

- *Fixed* Set: Indices listing the entries of x^k that are held *fixed*.
- Definition: a set of indices

$$I^k = \left\{ i \middle| x_i^k = 0, \, [\nabla f(x^k)]_i > 0 \right\}.$$

- A **subset** of active variables at iteration k.
- Contains active variables that satisfy the KKT conditions.

What happens when $x_j^k = 0$, but $[\nabla f(x^k)]_j \leq 0$?

- Further optimization is possible.
- Could become $x_j^{k+1} > 0$ and $[\nabla f(x^{k+1})]_j = 0$.
- Thus, such an x_j^k is NOT designated a *fixed* variable.

Solve the problem over *Free* variables only.

What happens when $x_j^k = 0$, but $[\nabla f(x^k)]_j \leq 0$?

- Further optimization is possible.
- Could become $x_j^{k+1} > 0$ and $[\nabla f(x^{k+1})]_j = 0$.
- Thus, such an x_j^k is NOT designated a *fixed* variable.

Solve the problem over *Free* variables only.

What happens when $x_j^k = 0$, but $[\nabla f(x^k)]_j \leq 0$?

- Further optimization is possible.
- Could become $x_j^{k+1} > 0$ and $[\nabla f(x^{k+1})]_j = 0$.
- Thus, such an x_j^k is NOT designated a *fixed* variable.

Solve the problem over *Free* variables only.

What happens when $x_j^k = 0$, but $[\nabla f(x^k)]_j \leq 0$?

- Further optimization is possible.
- Could become $x_j^{k+1} > 0$ and $[\nabla f(x^{k+1})]_j = 0$.
- Thus, such an x_j^k is NOT designated a *fixed* variable.

Solve the problem over *Free* variables only.

What happens when $x_j^k = 0$, but $[\nabla f(x^k)]_j \leq 0$?

- Further optimization is possible.
- Could become $x_j^{k+1} > 0$ and $[\nabla f(x^{k+1})]_j = 0$.
- Thus, such an x_j^k is NOT designated a *fixed* variable.

Solve the problem over *Free* variables only.

- Non-diagonal gradient scaling to improve convergence rate.
- Let H^k be the current approximation to the Hessian.
- BFGS update adds a rank-two correction to H^k to obtain

$$H^{k+1} = H^k - \frac{H^k u u^T H^k}{u^T H^k u} + \frac{w w^T}{u^T w},$$

where w and u are defined as

$$w = \nabla f(x^{k+1}) - \nabla f(x^k), \quad \text{and} \quad u = x^{k+1} - x^k.$$

- Let D^k denote the inverse of H^k.
- Apply the Sherman-Morrison-Woodbury formula to get:

$$D^{k+1} = D^k + \left(1 + \frac{w^T D^k w}{u^T w}\right) \frac{u u^T}{u^T w} - \frac{(D^k w u^T + u w^T D^k)}{u^T w}.$$

- Non-diagonal gradient scaling to improve convergence rate.
- Let H^k be the current approximation to the Hessian.
- BFGS update adds a rank-two correction to H^k to obtain

$$H^{k+1} = H^k - \frac{H^k u u^T H^k}{u^T H^k u} + \frac{w w^T}{u^T w},$$

where w and u are defined as

$$w = \nabla f(x^{k+1}) - \nabla f(x^k), \quad \text{and} \quad u = x^{k+1} - x^k.$$

- Let D^k denote the inverse of H^k.
- Apply the Sherman-Morrison-Woodbury formula to get:

$$D^{k+1} = D^k + \left(1 + \frac{w^T D^k w}{u^T w}\right) \frac{u u^T}{u^T w} - \frac{(D^k w u^T + u w^T D^k)}{u^T w}.$$

- Non-diagonal gradient scaling to improve convergence rate.
- Let H^k be the current approximation to the Hessian.
- BFGS update adds a rank-two correction to H^k to obtain

$$H^{k+1} = H^k - \frac{H^k uu^T H^k}{u^T H^k u} + \frac{ww^T}{u^T w},$$

where w and u are defined as

$$w = \nabla f(x^{k+1}) - \nabla f(x^k), \quad \text{and} \quad u = x^{k+1} - x^k.$$

- Let D^k denote the inverse of H^k.
- Apply the Sherman-Morrison-Woodbury formula to get:

$$D^{k+1} = D^k + \left(1 + \frac{w^T D^k w}{u^T w}\right) \frac{uu^T}{u^T w} - \frac{(D^k wu^T + uw^T D^k)}{u^T w}.$$

For the given problem,

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) = \frac{1}{2}\|Gx - h\|^2, \\ \text{subject to} \quad & x \geq 0. \end{aligned}$$

The gradient is

$$\nabla f(x) = G^T G x - G^T h.$$

The BFGS update reduces to

$$D^{k+1} \leftarrow D^k + \left(1 + \frac{u^T G^T G D^k G^T G u}{u^T G^T G u}\right) \frac{u u^T}{u^T G^T G u} - \frac{(D^k G^T G u u^T + u u^T G^T G D^k)}{u^T G^T G u}.$$

Define some quantities,

- Gradient matrices:

$$\nabla_C \mathscr{F}(B;C) = B^T BC - B^T A, \quad \text{and}$$
$$\nabla_B \mathscr{F}(B;C) = BCC^T - AC^T.$$

- Fixed set (corresponding to B):

$$I_+ = \left\{(i,j) \middle| B_{ij} = 0, [\nabla_B \mathscr{F}(B;C)]_{ij} > 0\right\}.$$

- Zero-out operator:

$$\left[\mathscr{Z}_+[X]\right]_{ij} = \begin{cases} X_{ij}, & (i,j) \notin I_+, \\ 0, & \text{otherwise.} \end{cases}$$

A subprocedure to update C in FNMA^{E}

1. Compute the gradient matrix $\nabla_C \mathscr{F}(B; C^{old})$.
2. Compute fixed set I_+ for C^{old}.
3. Compute the step length vector α using line-search.
4. Update C^{old} as

$$
\begin{aligned}
U &\leftarrow \mathscr{Z}_+\left[\nabla_C \mathscr{F}(B; C^{\text{old}})\right];\\
U &\leftarrow \mathscr{Z}_+\left[DU\right];\\
C^{\text{new}} &\leftarrow \mathscr{P}_+\left[C^{\text{old}} - U \cdot \text{diag}(\alpha)\right].
\end{aligned}
$$

5. $C^{\text{old}} \leftarrow C^{\text{new}}$.
6. Update D if necessary.

FNMA$^{\mathrm{E}}$

Input: $A \in \mathbb{R}_+^{M\times N}$, K such that $1 \le K \le \min\{M, N\}$

Output: $B \in \mathbb{R}_+^{M\times K}$, $C \in \mathbb{R}_+^{K\times N}$

1. Initialize B^0, C^0, $t = 0$.

repeat

  2. $B \leftarrow B^t$; $C^{\mathrm{old}} \leftarrow C^t$.

  repeat

    3. The subprocedure to update C.

  until C^{old} converges

  4. $C^{t+1} \leftarrow C^{\mathrm{old}}$; $C \leftarrow C^{t+1}$; $B^{\mathrm{old}} \leftarrow B^t$.

  repeat

    5. The subprocedure to update B.

  until B^{old} converges

  6. $B^{t+1} \leftarrow B^{\mathrm{old}}$; $t \leftarrow t+1$.

until Stopping criteria are met

Theorem (Convergence of FNMA$^{\mathrm{E}}$)

If B^t and C^t retain full-rank, then the sequence $\{B^t, C^t\}$ generated by Algorithm FNMA^E converges to a stationary point of the least squares NNMA problem.

Sketch of proof:

- Show that unique solution is obtained at each alternating step.
- Show that the sequence $\{B^t, C^t\}$ has a limit point.
- Invoke proof of the two-block Gauss-Seidel method.

FNMA^{I}: an *inexact* Method

Update Rule

A subprocedure to update C in FNMA^{I}

1. Compute the gradient matrix $\nabla_C \mathscr{F}(B; C^{old})$.
2. Compute fixed set I_+ for C^{old}.
3. Update C^{old} as

$$U \leftarrow \mathscr{Z}_+\left[\nabla_C \mathscr{F}(B; C^{\mathrm{old}})\right];$$
$$U \leftarrow \mathscr{Z}_+\left[(B^T B)^{-1} U\right];$$
$$C^{\mathrm{new}} \leftarrow \mathscr{P}_+\left[C^{\mathrm{old}} - \alpha U)\right].$$

4. $C^{\mathrm{old}} \leftarrow C^{\mathrm{new}}$.

To speed up computation:

- Step-size α is parameterized.
- Inverse Hessian is used for non-diagonal gradient scaling.
- Note the analogy between FNMA^{I}and ALS.

Theorem (Monotonicity of FNMA$^{\mathrm{I}}$)

If B^t and C^t retain full-rank, then FNMA^I decreases its objective function monotonically for sufficiently small α.

Sketch of proof:

- Since B^t and C^t retain full-rank, their Hessians are positive definite, hence satisfy condition for descent in the proof of FNMA$^{\mathrm{E}}$.
- Show that for sufficiently small α, the algorithm decreases the objective function value for each subproblem.

Extensions

For Regularizers in the Objective Function

Regularized version of the NNMA problem,

$$\underset{B,C\geq 0}{\text{minimize}} \quad \frac{1}{2}\|A-BC\|_F^2+\lambda\|B\|_F^2+\mu\|C\|_F^2, \qquad \lambda,\mu>0.$$

- The gradient and Hessian get redefined. For example,

The gradient

$$\nabla_C \mathscr{F}(B;C)=(B^TB+\lambda I)C-B^TA,$$

and the Hessian

$$\nabla_C^2 \mathscr{F}(B;C)=(B^TB+\lambda I).$$

- Use these updated values in the algorithms FNMA^E and FNMA^I
- Regularization ensures the Hessian remains positive-definite.
- All convergence results for FNMA^E & FNMA^I carry over without any additional work.

Extensions

For Regularizers in the Objective Function

Regularized version of the NNMA problem,

$$\underset{B,C\geq 0}{\text{minimize}} \quad \frac{1}{2}\|A-BC\|_F^2+\lambda\|B\|_F^2+\mu\|C\|_F^2, \qquad \lambda,\mu>0.$$

- The gradient and Hessian get redefined. For example,

The gradient

$$\nabla_C \mathscr{F}(B;C)=(B^TB+\lambda I)C-B^TA,$$

and the Hessian

$$\nabla_C^2 \mathscr{F}(B;C)=(B^TB+\lambda I).$$

- Use these updated values in the algorithms FNMA^{E} and FNMA^{I}
- Regularization ensures the Hessian remains positive-definite.
- All convergence results for FNMA^{E} & FNMA^{I} carry over without any additional work.

Extensions

For Regularizers in the Objective Function

Regularized version of the NNMA problem,

$$\underset{B,C\geq 0}{\text{minimize}} \quad \frac{1}{2}\|A-BC\|_F^2+\lambda\|B\|_F^2+\mu\|C\|_F^2, \qquad \lambda,\mu>0.$$

- The gradient and Hessian get redefined. For example,

The gradient

$$\nabla_C \mathscr{F}(B;C)=(B^TB+\lambda I)C-B^TA,$$

and the Hessian

$$\nabla_C^2 \mathscr{F}(B;C)=(B^TB+\lambda I).$$

- Use these updated values in the algorithms FNMA^E and FNMA^I
- Regularization ensures the Hessian remains positive-definite.
- All convergence results for FNMA^E & FNMA^I carry over without any additional work.

NNMA problem with box-constraints,

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|A - BC\|_F^2, \\ \text{subject to} & P \le B \le Q, \quad R \le C \le S, \end{array}$$

where inequalities are component-wise.

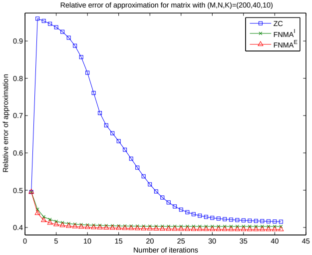
- Replace the $\mathscr{P}_+[\cdot]$ projection by $\mathscr{P}_\Omega[\cdot]$, where

$$[\mathscr{P}_\Omega[x]]_i = \begin{cases} p_i & : \quad x_i \le p_i \\ x_i & : \quad p_i < x_i < q_i \\ q_i & : \quad q_i \le x_i \end{cases}$$

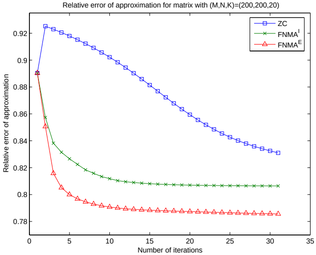
- Fixed set for B is redefined as

$$I_\Omega = \left\{ (i,j) \middle| \left(B_{ij} = P_{ij}, [\nabla_B \mathscr{F}(B;C)]_{ij} > 0 \right), \text{ or } \left(B_{ij} = Q_{ij}, [\nabla_B \mathscr{F}(B;C)]_{ij} < 0 \right) \right\}.$$

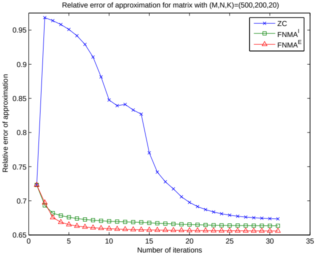
Experiments
Comparisons against ZC



(a) Dense



(b) Sparse

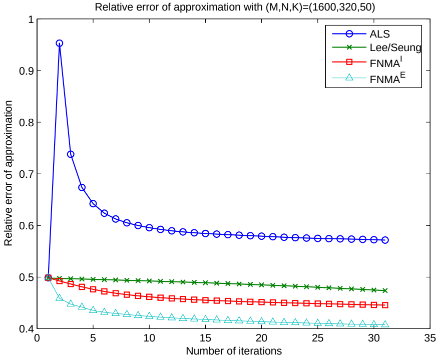


(c) Sparse

- Relative approximation error against iteration count for ZC, FNMA^{I} & FNMA^{E}.
- Relative errors achieved by both FNMA^{I} and FNMA^{E} are lower than ZC.
- Note that ZC does not decrease the errors monotonically.

Experiments

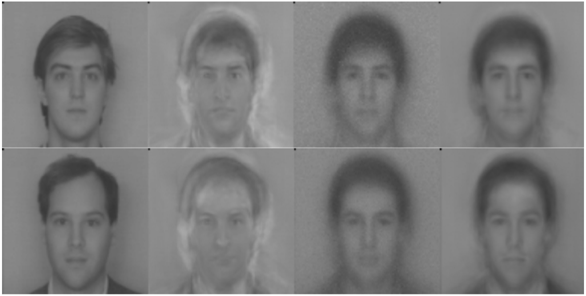
Comparisons against Lee & Seung's and ALS



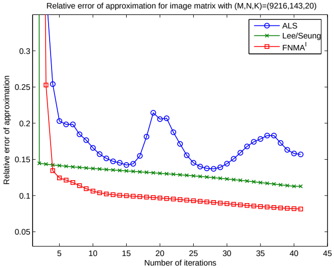
- Relative error values against iteration count for a random dense matrix of size 1600×320 for a rank 50 approximation.
- All methods other than ALS show a monotonic decrease when initialized with one step of LS.

Experiments

Application to Image Processing



Original ALS LS FNMA^I



- Image reconstruction as obtained by the ALS, LS, and FNMA^I procedures.
- Reconstruction was computed from a rank-20 approximation
- ALS leads to a non-monotonic change in the objective function value.

Summary

- Non-diagonal gradient scaling scheme can alleviate slow convergence of the gradient descent based methods.
- Naïve combination of projection and non-diagonal gradient scaling has theoretical deficiencies.
- We provide an algorithmic framework based on partitioning of variables
  - an *exact* & probably convergent method (more accurate)
  - an *inexact* method analogous to ALS (faster).
- In progress...
  - Other optimization techniques such as L-BFGS, conjugate gradient, trust region, etc.
  - More general distortion functions, e.g., Bregman divergences.
  - Exploit sparsity of problem.
  - Develop publicly available software toolbox.

References

R. Bro and S. D. Jong.
A Fast Non-negativity-constrained Least Squares Algorithm.
Journal of Chemometrics, 11(5):393–401, 1997.

D. Kim, S. Sra, and I. S. Dhillon.
Fast Newton-type Methods for the Least Squares Nonnegative Matrix Approximation Problem.
To appear in Proceedings of SIAM Conference on Data Mining, 2007.

C. L. Lawson and R. J. Hanson.
Solving Least Squares Problems.
Prentice–Hall, 1974.

D. D. Lee and H. S. Seung.
Learning The Parts of Objects by Nonnegative Matrix Factorization.
Nature, 401:788–791, 1999.

C. Lin.
Projected Gradient Methods for Non-negative Matrix Factorization.
Technical Report ISSTECH-95-013, National Taiwan University, 2005.

M. Merritt and Y. Zhang.
Interior-Point Gradient Method for Large-Scale Totally Nonnegative Least Squares Problems.
Journal of Optimization Theory and Applications, 126(1):191–202, 2005.

R. Zdunek and A. Cichocki.
Non-Negative Matrix Factorization with Quasi-Newton Optimization.
In *Eighth International Conference on Artificial Intelligence and Soft Computing, ICAISC*, pages 870–879, 2006.