Fast Newton-type Methods for Nonnegative Matrix and Tensor Approximation

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 $A \approx BC$ 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:

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

- NNMA objective function is not simultaneously convex in B & C
- But is individually convex in B & 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 solve the problem w.r.t C, Obtain C^{t+1}.
3. Fix C^{t+1} and solve the problem w.r.t B, Obtain B^{t+1}.
4. Let $t \leftarrow t+1$, & repeat Steps 2 and 3 until convergence criteria are satisfied.

Nonnegative Tensor Approximation

Problem Setting

- For brevity, consider 3-mode tensors only
- Least squares objective function

$$\mathscr{A}, \mathscr{T} \in \mathbb{R}_+^{\ell \times m \times n}$$

$$\|\mathscr{A} - \mathscr{T}\|_F^2 = \sum_{i=1}^{l} \sum_{j=1}^{m} \sum_{k=1}^{n} \left([\mathscr{A}]_{ijk} - [\mathscr{T}]_{ijk}\right)^2.$$

- Given a nonnegative tensor $\mathscr{A} \in \mathbb{R}^{\ell \times m \times n}$, find a nonnegative approximation $\mathscr{T} \in \mathbb{R}^{\ell \times m \times n}$ which consists of nonnegative *components*
- Tensor decomposition :
  "PARAFAC" or "Tucker"

Nonnegative PARAFAC Decomposition

- PARAFAC or Outer Product Decomposition:

$$
\begin{aligned}
\text{minimize} \quad & \|\mathscr{A} - \mathscr{T}\|_F^2 \\
\text{subject to} \quad & \mathscr{T} = \sum_{i=1}^{k} p^i \otimes q^i \otimes r^i, \\
\text{where} \quad & \mathscr{A},\ \mathscr{T} \in \mathbb{R}^{\ell \times m \times n}, \\
& P = [p^i] \in \mathbb{R}^{\ell \times k},\ Q = [q^i] \in \mathbb{R}^{m \times k},\ R = [r^i] \in \mathbb{R}^{n \times k}, \\
& P,\ Q,\ R \geq 0.
\end{aligned}
$$

Nonnegative Tucker Decomposition

- Tucker decomposition of tensors,

$$
\begin{array}{rl}
\text{minimize} & \|\mathscr{A} - \mathscr{T}\|_{\mathrm{F}}^2 \\
\text{subject to} & \mathscr{T} = \big(P, Q, R\big) \cdot \mathscr{Z}, \\
\text{where} & \mathscr{A},\ \mathscr{T} \in \mathbb{R}^{\ell\times m\times n},\ \mathscr{Z} \in \mathbb{R}^{p\times q\times r}, \\
 & P \in \mathbb{R}^{\ell\times p},\ Q \in \mathbb{R}^{m\times q},\ R \in \mathbb{R}^{n\times r}, \\
 & \mathscr{Z},\ P,\ Q,\ R \geq 0.
\end{array}
$$

- Basic Idea: build a matrix approximation problem
- For example, for matrix factor P,
  - Fix Q and R
  - Form $Z \in \mathbb{R}^{k \times mn}$ where i-th row corresponds to vectorized $q^i \otimes r^i$
  - Form $A \in \mathbb{R}^{\ell \times mn}$ where i-th row corresponds to vectorized $\mathscr{A}(i,:,:)$
  - Now the problem is

$$\underset{P \geq 0}{\text{minimize}} \quad \|A - PZ\|_{\mathrm{F}}^2.$$

Nonnegative Tucker Decomposition

Algorithm - Update Matrix Factors by Reducing to NNMA

- Basic Idea: build a matrix approximation problem
- For example, for matrix factor P,
  - Fix $\mathscr{Z}$, Q and R
  - Form $Z \in \mathbb{R}^{p \times mn}$ by flattening the tensor $\big(Q, R\big) \cdot \mathscr{Z}$ along mode-1
  - Computing $\mathscr{T} = \big(P, Q, R\big) \cdot \mathscr{Z}$ is equivalent to PZ
  - Flatten the tensor $\mathscr{A}$ similarly, obtain a matrix $A \in \mathbb{R}^{\ell \times mn}$
  - Now the problem is

$$\underset{P,Z \geq 0}{\text{minimize}} \quad \|A - PZ\|_{\mathrm{F}}^2.$$

- 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
- For example, fix B and find a solution x — i-th column of C reduces a NNLS problem:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x) = \frac{1}{2}\|Bx - a_i\|_2^2, \\ \text{subject to} & x \geq 0. \end{array}$$

Existing NNMA Algorithms

Exact Methods

Basic Framework for *Exact* Methods

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

$$C^{t+1} = \operatorname*{argmin}_C \mathscr{F}(B^t, C),$$

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

$$B^{t+1} = \operatorname*{argmin}_B \mathscr{F}(B, C^{t+1}),$$

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

Exact Methods

- Based on NNLS algorithms:
  - Active set procedure [Lawson & Hanson, 1974]
  - FNNLS [Bro & Jong, 1997]
  - Interior-point gradient method
- Projected gradient method [Lin, 2005].

Existing NNMA Algorithms

Inexact Methods

Basic Framework for *Inexact* Methods

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.

Inexact Methods

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

- Active Set based methods

> NOT suitable for large-scale problems

- Gradient Descent based methods

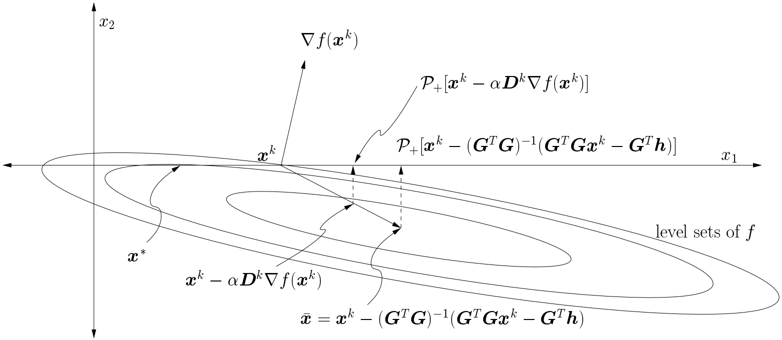
> May suffer from slow convergence — known as zigzagging

- Newton-type methods

> Naïve combination with projection does NOT guarantee convergence

Previous Attempts at Newton-type Methods for NNMA

Difficulties



- Naïve Combination of projection step and non-diagonal gradient scaling does not guarantee convergence
- 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 reduces to an **equality-constrained** problem
  - Equivalently one can solve an **unconstrained** sub-problem over inactive variables

Projection :

- The projection step identifies active variables at each iteration

Gradient :

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

- Combine Projection with non-diagonal gradient scaling
- At each iteration, partition variables into two disjoint set, Fixed and Free variables
- Optimize the objective function over Free variables
- Convergence to a stationary point of $\mathscr{F}$ is guaranteed
- Any positive definite gradient scaling scheme is allowed, i.e., the inverse of full Hessian, an approximated Hessian by BFGS, conjugate gradient, etc

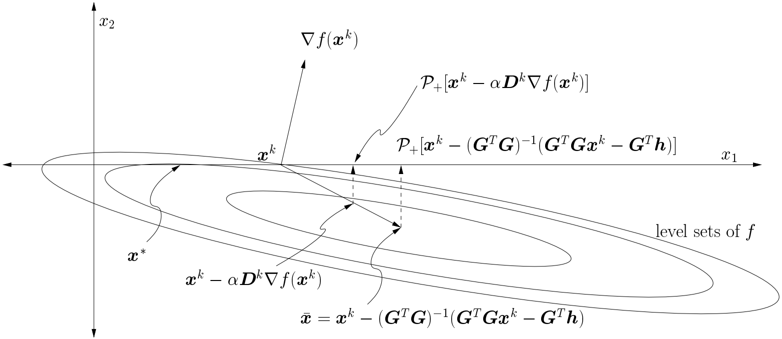
Divide variables into *Free* variables and *Fixed* variables.

- *Fixed* Set: Indices listing 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

Newton-type Methods



Fast Newton-type Nonnegative Matrix Approximation

FNMA^{E} & FNMA^{I}– an *exact* and *Inexact* Method

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.
4. Update C^{old} as

$$U \leftarrow \mathscr{Z}_+\left[\nabla_C \mathscr{F}(B; C^{\text{old}})\right]; \quad \textit{//Remove gradient info. from fixed vars}$$

$$U \leftarrow \mathscr{Z}_+\left[DU\right]; \quad \textit{//Fix fixed vars}$$

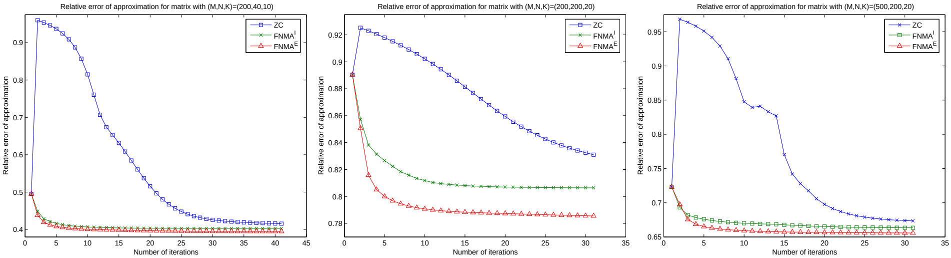
$$C^{\text{new}} \leftarrow \mathscr{P}_+\left[C^{\text{old}} - U \cdot \text{diag}(\alpha)\right] \quad \textit{//Enforce feasibility}$$

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

FNMA^{I}: To speed up computation,

- Step-size α is parameterized
- Inverse Hessian is used for non-diagonal gradient scaling

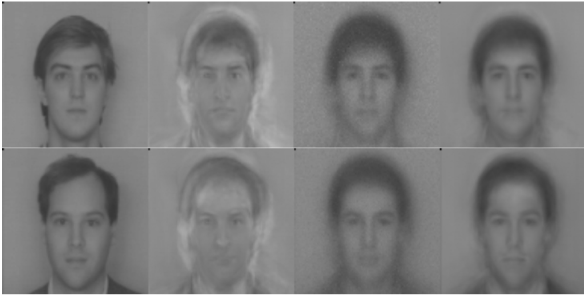
Experiments
Comparisons against ZC



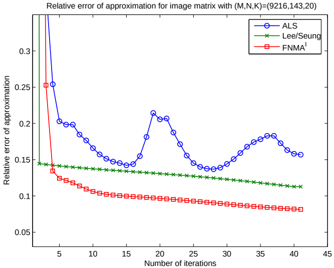
(a) Dense (b) Sparse (c) Sparse

- 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
Application to Image Processing



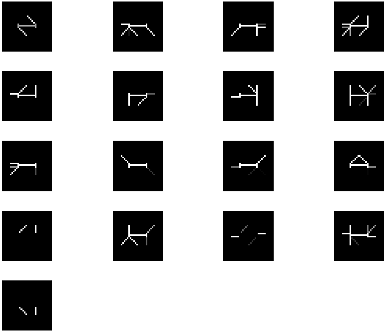
Original ALS LS FNMA^{I}



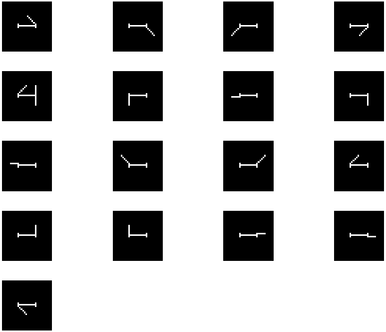
- Image reconstruction as obtained by the ALS, LS, and FNMA^{I} procedures
- Reconstruction was computed from a rank-20 approximation
- ALS leads to a non-monotonic change in the objective function value

Experiments

Application to Image Processing - Swimmer dataset - rank 13



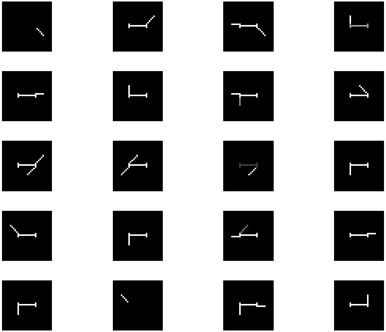
Lee & Seung's rank 17



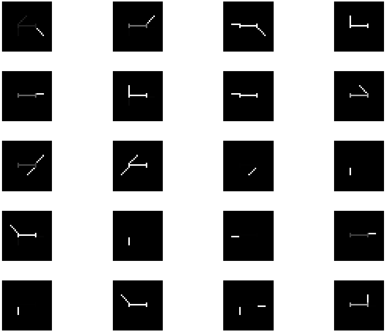
FNMA^{E} rank 17

Experiments

Application to Image Processing - Swimmer dataset - rank 20



Lee & Seung's rank 20

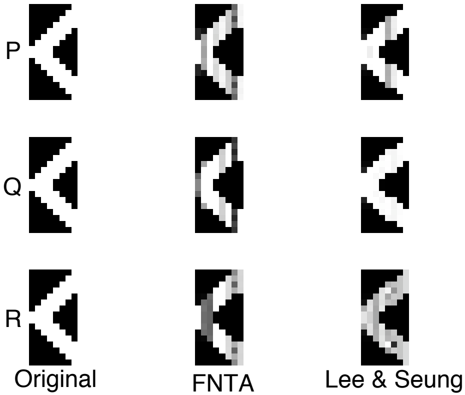


FNMA^E rank 20

	Lee & Seung's	FNMA^E	
Rank 13	140.53	47.06	Elapsed CPU Time
	4.49×10^7	2.01×10^7	Objective Function Value
Rank 17	182.24	62.29	Elapsed CPU Time
	2.41×10^7	$\mathbf{6.85 \times 10^{-4}}$	Objective Value
Rank 20	156.18	41.93	Elapsed CPU Time
	5.61×10^5	4.71×10^3	Objective Function Value

Experiments

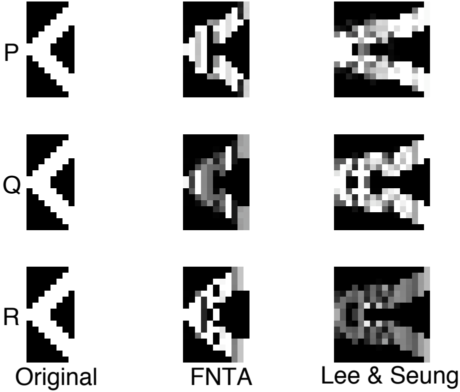
Comparison against Lee & Seung-type Algorithms - PARAFAC/ $k = 8$



- Nonnegative PARAFAC decomposition with $k = 8$
- Original $P, Q, R \in \mathbb{R}^{16\times 8}$ and rank 8
- Final rank is 8 for both FNTA and Lee & Seung
- FNTA gives smaller reconstruction error

Experiments

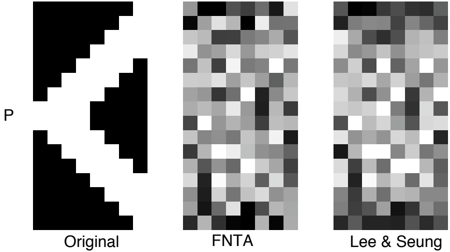
Comparison against Lee & Seung-type Algorithms - PARAFAC/ $k = 16$



- Nonnegative PARAFAC decomposition with $k = 16$
- Original $P, Q, R \in \mathbb{R}^{16 \times 8}$ and rank 8
- Final ranks are 11 for FNTA and 16 for Lee & Seung
- FNTA produces sparser and low-rank solution

Experiments

Comparison against Lee & Seung-type Algorithms - Tucker



- Nonnegative Tucker decomposition with $[p\ q\ r] = [8\ 8\ 8]$
- Original P, Q, R as before
- Original core tensor has 1 for all entries
- FNTA gives smaller reconstruction error
- Both methods fit the original tensor very well (error < 1e-4)
- Unlike PARAFAC, Both are unable to discover factors

Simultaneous Update of Factors

- Instead of alternating optimization between B and C,
- Update B and C jointly
- For example, after computing $\bar{B}$ and $\bar{C}$ s.t.

$$\bar{B} = \operatorname*{argmin}_{B \geq 0} \|A - BC_k\|_F^2, \quad \bar{C} = \operatorname*{argmin}_{C \geq 0} \|A - B_k C\|_F^2.$$

- Solve two-dimensional bound-constrained optimization,

$$\min_{0 \leq (\beta, \gamma) \leq 1} \|A - \big(B_k + \beta(\bar{B} - B_k)\big)\big(C_k + \gamma(\bar{C} - C_k)\big)\|_F^2.$$

Summary

- Nonnegative matrix and tensor approximation problems
- Non-diagonal gradient scaling can give faster convergence
- Algorithmic framework based on partitioning of variables
  - an *exact* & probably convergent method (more accurate)
  - an *inexact* method analogous to ALS (faster)
  - extensions to NNTA
- In progress...
  - More general distortion functions, e.g., Bregman divergences
  - Publicly available software toolbox
- A MATLAB Implementation of FNMA^E is now available at www.cs.utexas.edu/users/dmkim/Source/software/nnma/index.html

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.
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.

Amnon Shashua and Tamir Hazan.
Non-negative Tensor Factorization with Applications to Statistics and Computer Vision.
In *International Conference on Machine Learning*, pages 792–799, 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.