Non-negative Matrix Factorization via Alternating Updates

Yingyu Liang, Princeton University

Joint work with
Yuanzhi Li    Andrej Risteski

Simons workshop, Berkeley, Nov 2016
Non-convex Problems in ML

Dictionary learning

Matrix completion

Deep learning
Non-convex Problems in ML

NP-hard to solve in the worst case

Matrix completion

Deep learning
Non-convex Problems in ML

• In practice, often solved by “local improvement”
  • Gradient descent and variants
Non-convex Problems in ML

- In practice, often solved by “local improvement”
  - Gradient descent and variants
  - Alternating update

Matrix factorization: Given $Y$, find $A, X$ s.t. $Y = AX$

for $t = 1, 2, \ldots$ do
  Fix $A$, update $X$
  Fix $X$, update $A$
Non-convex Problems in ML

- In practice, often solved by “local improvement”
  - Gradient descent and variants
  - Alternating update

When and why do such simple algos work for the hard problems?
Goal: provable guarantees of simple algos under natural assumptions
Non-convex Problems in ML

• In practice, often solved by “local improvement”
  • Gradient descent and variants
  • Alternating update

When and why do such simple algos work for the hard problems?
Goal: provable guarantees of simple algos under natural assumptions

This work: alternating update for Non-negative Matrix Factorization
Non-negative Matrix Factorization (NMF)

• Given: matrix $Y \in \mathbb{R}^{d \times m}$
• Find: non-negative matrices $A \in \mathbb{R}^{d \times k}, X \in \mathbb{R}^{k \times m}$ s.t. $Y = AX$
# NMF in ML Applications

Basic tool in machine learning

- **Topic modeling** [Blei-Ng-Jordan03, Arora-Ge-Kannan-Moitra12, Arora-Ge-Moitra12,…]

<table>
<thead>
<tr>
<th></th>
<th>Doc1</th>
<th>Doc2</th>
<th>Doc3</th>
<th>……</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>……</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rain</td>
<td>0.04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>soccer</td>
<td>0.08</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Weather</th>
<th>Sport</th>
<th>Science</th>
<th>……</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>……</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rain</td>
<td></td>
<td>0.2</td>
<td></td>
<td>0.8</td>
</tr>
<tr>
<td>soccer</td>
<td></td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>physics</td>
<td></td>
<td></td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

```
NMF in ML Applications

Basic tool in machine learning

- **Topic modeling** [Blei-Ng-Jordan03, Arora-Ge-Kannan-Moitra12, Arora-Ge-Moitra12,…]
- **Computer vision** [Lee-Seung97, Lee-Seung99, Buchsbaum-Bloch02,…]
NMF in ML Applications

Basic tool in machine learning

• Topic modeling [Blei-Ng-Jordan03, Arora-Ge-Kannan-Moitra12, Arora-Ge-Moitra12,…]

• Computer vision [Lee-Seung97, Lee-Seung99, Buchsbaum-Bloch02,…]

• Many others: network analysis, information retrieval, ……
Worst Case v.s. Practical Heuristic

**Worst case analysis** [Arora-Ge-Kannan-Moitra12]
- Upper bound: $O \left( (dm)^{k^2 2^k} \right)$
- Lower bound: no $(dm)^{o(k)}$ algo, assuming ETH

**Alternating updates**: typical heuristic, suggested by Lee-Seung

Set a good initialization for $A$ (often by experts)

for $t = 1, 2, \ldots$ do
    Decode: compute $X$ from the current $A$
    Update: modify $A$ based on $X$
Analyzing Non-convex Problems

- Generative model: the input data is generated from (a distribution defined by) a ground-truth solution
- Warm start: good initialization not far away from the ground-truth
Beyond Worst Case for NMF

- Separability-based assumptions [Arora-Ge-Kannan-Moitra12]
  - Motivated by topic modeling: each column of $A$ (topic) has an anchor word

\[
\begin{array}{ccc}
\text{Weather} & \text{Sport} & \text{Science} \\
\text{the} & 0.05 & 0.05 & 0.05 \\
\text{…} & & & \\
\text{rain} & 0.2 & 0 & 0 \\
\text{soccer} & & 0.3 & \\
\text{physics} & & & 0.1 \\
\end{array}
\]
Beyond Worst Case for NMF

• Separability-based assumptions [Arora-Ge-Kannan-Moitra12]
  • Motivated by topic modeling: each column of $A$ (topic) has an anchor word
  • Lots of subsequent work [Arora-Ge-Moitra12, Arora-Ge-Halpern-Mimno-Moitra-Sontag-Wu-Zhu12, Ge-Zhou15, Bhattacharyya-Goyal-Kannan-Pani16,…]

• Variational inference [Awasthi-Risteski15]
  • Alternating update method on objective $\text{KL-divergence}(Y, AX)$
  • Requires relatively strong assumptions on $A$, and/or a warm start depending on its dynamic range (not realistic)
Outline

• Introduction
• Our model, algorithm and main results
• Analysis of the algorithm
Generative Model

Each column of $\mathbf{Y}$ is i.i.d.

$$y = A^* x^*$$
Generative Model

Each column of $Y$ is i.i.d. example from

$$y = A^* x^*$$

• (A1): columns of $A^*$ are linearly independent
Generative Model

Each column of $Y$ is i.i.d. example from

$$y = A^* x^*$$

• (A1): columns of $A^*$ are linearly independent
• (A2): $x_i^*$’s are independent random variables

$$x_i^* = \begin{cases} 
1 & \text{with probability } s/k \\
0 & \text{otherwise}
\end{cases}$$

where $s$ is a parameter
Our Algorithm

Parameters: $\alpha, \eta_1, \eta_2$

Initialize $A$

for $t = 1, 2, \ldots$ do

Decode: $x \leftarrow \phi_\alpha(A^\dagger y)$ for each example $y$

Known as Rectified Linear Units (ReLU) in Deep Learning
Our Algorithm

Parameters: \( \alpha, \eta_1, \eta_2 \)

Initialize \( A \)

for \( t = 1, 2, \ldots \) do

Decode: \( x \leftarrow \phi_\alpha(A^\dagger y) \) for each example \( y \)

Update: \( A \leftarrow (1 - \eta_1)A + \eta_2 \mathbb{E}[(y - y')(x - x')^\top] \)

\( y, y' \) are two independent examples, and \( x, x' \) are their decodings
Our Algorithm

Parameters: $\alpha, \eta_1, \eta_2$

Initialize $A$
for $t = 1, 2, \ldots$ do
  Decode: $x \leftarrow \phi_\alpha(A^\dagger y)$ for each example $y$
  Update: $A \leftarrow (1 - \eta_1)A + \eta_2\mathbb{E}[(y - y')(x - x')^\top]$ 

Sanity check: if $A = A^*$, then
$\phi_\alpha(A^\dagger y) = \phi_\alpha(A^\dagger A^* x^*) = \phi_\alpha(x^*) = x^*$
Our Algorithm

Parameters: $\alpha, \eta_1, \eta_2$

Initialize $A$

\[ \text{for } t = 1, 2, \ldots \text{ do } \]

Decode: $x \leftarrow \phi_\alpha(A^\dagger y)$ for each example $y$

Update: $A \leftarrow (1 - \eta_1)A + \eta_2 \mathbb{E}[(y - y')(x - x')^\top]$

Sanity check: if $A = A^*$, then

\[
\mathbb{E}[(y - y')(x - x')^\top] = A^* \mathbb{E}[(x^* - (x')^*)(x^* - (x')^*)^\top] \\ \propto A^*
\]
Warm Start

• (A3): warm start $A$ with error $\ell \leq 1/10$

\[
A = A^*(\Sigma + E) \text{ where }
\Sigma \text{ is diagonal with } \Sigma_{i,i} \geq 1 - \ell,
E \text{ is off-diagonal with } \|E\|_1, \|E\|_\infty \leq \ell
\]
Warm Start

• (A3): warm start $A$ with error $\ell \leq 1/10$

$$A = A^*(\Sigma + E)$$

where

$\Sigma$ is diagonal with $\Sigma_{i,i} \geq 1 - \ell$,

$E$ is off-diagonal with $\|E\|_1, \|E\|_\infty \leq \ell$

$$A_i = \sum_{i,i} A_i^* + \sum_{j \neq i} E_{j,i} A_j^*$$

Aligned with truth  Not too much
Main Result

After $O(\log(1/\epsilon))$ iterations, $\sum_{i,i} \geq 1 - \ell$ and $\|E\|_1, \|E\|_\infty \leq \epsilon.$
Analysis: Overview

\[ A = A^* (\Sigma + E) \]

Show the algo will

1. Maintain \( \Sigma_{i,i} \geq 1 - \ell \) and
2. Decrease the potential function

\[ \|E_+\| + \beta \|E_-\| \]

where \( \beta > 1 \) is a constant,

\( E_+ (E_-) \) is the positive (negative) part of \( E \)
Analysis: Effect of ReLU

Decode: \( x \leftarrow \phi_\alpha(A^\dagger y) \)
Update: \( A \leftarrow (1 - \eta_1)A + \eta_2 \Delta, \Delta = \mathbb{E}[(y - y')(x - x')^\top] \)

\[ x = \phi_\alpha(A^\dagger A^* x^*) = \phi_\alpha((I + E)^{-1}x^*) \approx \phi_\alpha(x^* - Ex^*) \]
Analysis: Effect of ReLU

Decode: $x \leftarrow \phi_\alpha(A^\dagger y)$
Update: $A \leftarrow (1 - \eta_1)A + \eta_2 \Delta$, $\Delta = \mathbb{E}[(y - y')(x - x')^\top]$}

- When $A = A^*(I + E)$, how is $E$ changed by the update?
  
  $x = \phi_\alpha(A^\dagger A^*x^*) \approx \phi_\alpha(x^* - Ex^*)$

\[ [x^* - Ex^*]_1 \quad [x^* - Ex^*]_2 \quad \ldots \]
Analysis: Effect of ReLU

Decode: \( x \leftarrow \phi_\alpha(A^\dagger y) \)

Update: \( A \leftarrow (1 - \eta_1)A + \eta_2 \Delta, \Delta = \mathbb{E}[(y - y')(x - x')^\top] \)

• When \( A = A^* (I + E) \), how is \( E \) changed by the update?

\[
x = \phi_\alpha(A^\dagger A^* x^*) \approx \phi_\alpha(x^* - Ex^*)
\]

Much less noise after \( \phi_\alpha \)
Analysis: Change of Error Matrix

Decode: $x \leftarrow \phi_\alpha(A^\dagger y)$
Update: $A \leftarrow (1 - \eta_1)A + \eta_2 \Delta$, $\Delta = \mathbb{E}[(y - y')(x - x')^\top]$ 

• For a small constant $\nu$

$E \leftarrow (1 - \eta_1)E - \eta_2 \nu E_+$

• Therefore, $\|E_+\| + \beta\|E_-\|$ decreases
More General Results

1. Each column of $Y$ is generated i.i.d. by

$$y = A^*x^* + \xi$$

- The decoding is

$$x = \phi_\alpha(A^\dagger y) \approx \phi_\alpha(x^* - Ex^* + A^\dagger \xi)$$

- So can* tolerate large adversarial noise,
- and tolerate zero-mean noise much larger than signal $x_i^*$’s

2. Distribution of $x_i^*$’s only needs to satisfy some moment conditions
Conclusion

• Beyond worse case analysis of NMF
  • generative model with mild condition on the feature matrix $A$
• Provable guarantee of alternating update algorithm
• Strong denoising effect by ReLU + non-negativity
Conclusion

• Beyond worse case analysis of NMF
  • generative model with mild condition on the feature matrix $A$
• Provable guarantee of alternating update algorithm
• Strong denoising effect by ReLU + non-negativity

Thanks! Q&A