Greedy algorithm for large scale
Nonnegative matrix/tensor decomposition

LIDS student conference 2015
Qingqiqing Huang

Joint work with Tong Zhang at Baidu
Nonnegative matrix factorization

- Problem

Given $F \in \mathbb{R}_+^{n \times m}$ and $r$, find $U \in \mathbb{R}_+^{n \times r}$, $V \in \mathbb{R}_+^{r \times m}$ such that $F \approx UV$.

Non-convex, NP-hard $\min_{U \in \mathbb{R}_+^{n \times r}, V \in \mathbb{R}_+^{r \times m}} R(F, UV)$
Nonnegative matrix factorization

Problem

Given $F \in \mathbb{R}_+^{n \times m}$ and $r$, find $U \in \mathbb{R}_+^{n \times r}$, $V \in \mathbb{R}_+^{r \times m}$ such that $F \approx UV$.

Non-convex, NP-hard

\[
\min_{U \in \mathbb{R}_+^{n \times r}, V \in \mathbb{R}_+^{r \times m}} R(F, UV) \quad \|F - UV\|_F \\
D_{KL}(F \| UV)
\]

Regularization for sparsity...

\[
\begin{array}{c}
m \\
F \\ \approx \hat{F} = \begin{array}{c}
m \\
F \\ \approx \hat{F} = m \\
U \\
\times r \\
V
\end{array}
\end{array}
\]
Nonnegative **matrix** factorization

- **Problem**

  Given $F \in \mathbb{R}^{n \times m}_+$ and $r$, find $U \in \mathbb{R}^{n \times r}_+$, $V \in \mathbb{R}^{r \times m}_+$ such that $F \approx UV$.

  Non-convex, NP-hard  
  \[
  \min_{U \in \mathbb{R}^{n \times r}_+, V \in \mathbb{R}^{r \times m}_+} R(F, UV) \quad \|F - UV\|_F \quad D_{KL}(F\|UV)
  \]

  Regularization for sparsity...

- **Applications (why not PCA, Eckart-Young)**

  Nonnegative signals, probabilities, network

  ✔ Image compression
  ✔ Sound source separation
  ✔ Spectral clustering
  ✔ Topic model learning
  ✔ Hidden markov model learning
Nonnegative matrix factorization

- Problem

Given $F \in \mathbb{R}^{n \times m}_+$ and $r$, find $U \in \mathbb{R}^{n \times r}_+$, $V \in \mathbb{R}^{r \times m}_+$ such that $F \approx UV$.

Non-convex, NP-hard

$$\min_{U \in \mathbb{R}^{n \times r}_+, V \in \mathbb{R}^{r \times m}_+} R(F, UV)$$

$$\|F - UV\|_F$$

$$D_{KL}(F \| UV)$$

Regularization for sparsity...

- Applications (why not PCA, Eckart-Young)

Nonnegative signals, probabilities, network

- Image compression
- Sound source separation
- Spectral clustering
- Topic model learning
- Hidden markov model learning
Nonnegative tensor factorization

- Problem \( \min_{U^{(1)} \in \mathbb{R}_{+}^{n_1 \times r}, \ldots, U^{(d)} \in \mathbb{R}_{+}^{n_d \times r}} R(F, U^{(1)} \otimes U^{(2)} \otimes \ldots \otimes U^{(d)}) \)

- Tensor product: multi-linear, homogeneous
Nonnegative tensor factorization

- Problem: \( \min_{U^{(1)} \in \mathbb{R}_{+}^{n_1 \times r}, \ldots, U^{(d)} \in \mathbb{R}_{+}^{n_d \times r}} R(F, U^{(1)} \otimes U^{(2)} \otimes \ldots \otimes U^{(d)}) \)

- Tensor product: multi-linear, homogeneous

- A hard problem even without the positive constraint

- Applications (natural multi-dimensional data, image, video, moments)
Literature

\[ \|F - UV\|_F \quad D_{KL}(F\|UV) \]

- Alternating optimization method
- High variance especially for large scale problem
- Can we solve it in a more controlled way?
Literature

\[ \|F - UV\|_F \quad D_{KL}(F \| UV) \]

- Alternating optimization method
  high variance especially for large scale problem
  can we solve it in a more controlled way?

- Recent theoretical work on exact recovery under assumptions
  can we still solve it in a agnostic way?
## Literature

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>computation</strong></td>
<td>fast</td>
<td>$O(n^{r^2})$</td>
<td>$O(r \ poly(nm))$</td>
</tr>
<tr>
<td><strong>guarantee</strong></td>
<td>start from random initialization, converge to local optima</td>
<td>provably</td>
<td>$R(F, \hat{F}_r) \leq R(F, \hat{F}_s^*) + \epsilon$</td>
</tr>
<tr>
<td><strong>robustness</strong></td>
<td>optimization based, agnostic</td>
<td>assumptions: exact factorization / anchor word / random generation of factors</td>
<td>optimization based, agnostic</td>
</tr>
</tbody>
</table>
## Literature

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>computation</strong></td>
<td>fast</td>
<td>$O(n^{r^2})$</td>
<td>$O(r \text{ poly}(nm))$</td>
</tr>
<tr>
<td><strong>guarantee</strong></td>
<td>start from random initialization, converge to local optima</td>
<td>provably</td>
<td>$R(F, \widehat{F}_r) \leq R(F, \widehat{F}^*_s) + \epsilon$</td>
</tr>
<tr>
<td><strong>robustness</strong></td>
<td>optimization based, agnostic</td>
<td>assumptions: exact factorization / anchor word / random generation of factors</td>
<td>optimization based, agnostic</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>computation</strong></td>
<td>fast</td>
<td>$O(n^{r^2})$</td>
<td>$O(r \text{ poly}(nm))$</td>
</tr>
<tr>
<td><strong>guarantee</strong></td>
<td>start from random initialization, converge to local optima</td>
<td>provably</td>
<td>$R(F, \hat{F}<em>r) \leq R(F, \hat{F}</em>{s}^*) + \epsilon$</td>
</tr>
<tr>
<td><strong>robustness</strong></td>
<td>optimization based, agnostic</td>
<td>assumptions: exact factorization / anchor word / random generation of factors</td>
<td>optimization based, agnostic</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>computation</td>
<td>fast</td>
<td>$O(n^{r^2})$</td>
<td>$O(r \ poly(nm))$</td>
</tr>
<tr>
<td>guarantee</td>
<td>start from random initialization, converge to local optima</td>
<td>provably</td>
<td>$R(F, \hat{F}_r) \leq R(F, \hat{F}_s^*) + \epsilon$</td>
</tr>
<tr>
<td>robustness</td>
<td>optimization based, agnostic</td>
<td>assumptions: exact factorization / anchor word / random generation of factors</td>
<td>optimization based, agnostic</td>
</tr>
</tbody>
</table>
Algorithm

- Observation:
  positive weighted sum of rank-one matrices/tensors
  supported over the sphere in the positive orthant

\[F \approx \hat{F} = \sum_{i=1}^{r} \lambda_i u_i v_i^\top\]

\[u_i \in \mathcal{B}_+^m, \ v_i \in \mathcal{B}_+^n\]

\[\mathcal{B}_+^m = \{u \geq 0, \|u\|_2 = 1\}\]
Algorithm

- Observation:
  positive weighted sum of rank-one matrices/tensors supported over the sphere in the positive orthant

\[
F \approx \hat{F} = \sum_{i=1}^{r} \lambda_i u_i v_i^\top \\
F \approx \hat{F} = \sum_{i=1}^{r} \lambda_i u_i^{(1)} \otimes u_i^{(2)} \otimes \ldots \otimes u_i^{(d)}
\]

\( u_i \in \mathcal{B}_+^m, v_i \in \mathcal{B}_+^n \)

\( \mathcal{B}_+^m = \{ u \geq 0, \|u\|_2 = 1 \} \)

- Eckart-Young fails… but can we still find one at a time?

- Greedy feature selection (Frank-Wolfe)
  incremental, greedy, first order method
Incremental algorithm

\[ F \approx \hat{F} = \sum_{i=1}^{r} \lambda_i u_i v_i^\top, \quad u_i \in \mathcal{B}_+^m, v_i \in \mathcal{B}_+^n \]

At t-th round: start from a rank (t-1) \( \hat{F}_{t-1} = U_{t-1} V_{t-1} \) find a rank t NMF
Incremental algorithm

\[ F \approx \widehat{F} = \sum_{i=1}^{r} \lambda_i u_i v_i^\top, \quad u_i \in B^m_+, v_i \in B^n_+ \]

At t-th round: start from a rank (t-1) \( \widehat{F}_{t-1} = U_{t-1} V_{t-1} \) find a rank t NMF

- Step 1. Greedy feature selection

\[(u_t, v_t) = \arg\min_{u \in B^m_+, v \in B^n_+} u^\top \left( \nabla_X R(F, X) \big|_{\widehat{F}_{t-1}} \right) v\]

- Maximizing the decreasing rate of loss function at \( \widehat{F}_{t-1} \)
Incremental algorithm

\[ F \approx \hat{F} = \sum_{i=1}^{r} \lambda_i u_i v_i^\top, \quad u_i \in \mathcal{B}_+^m, v_i \in \mathcal{B}_+^n \]

At t-th round: start from a rank (t-1) \( \hat{F}_{t-1} = U_{t-1} V_{t-1} \) find a rank t NMF

- **Step 1. Greedy feature selection**
  \[
  (u_t, v_t) = \arg\min_{u \in \mathcal{B}_+^m, v \in \mathcal{B}_+^n} u^\top \left(\nabla_X R(F, X) | \hat{F}_{t-1}\right) v
  \]

  ✔ Maximizing the decreasing rate of loss function at \( \hat{F}_{t-1} \)

- **Step 2. Weight update (not on \( \lambda_i \)'s)**
  \[
  U_t = [U_{t-1}, u_t], \quad \tilde{V}_t = [V_{t-1}; v_{t-1}^\top]
  \]
  \[
  W_t = \arg\min_{W \in \mathbb{R}_{+}^{t \times t}} R(F, U_t W_t V_t) \quad t \times t, \text{ convex - easy}
  \]
Incremental algorithm

\[
F \approx \hat{F} = \sum_{i=1}^{r} \lambda_i u_i v_i^\top, \quad u_i \in \mathcal{B}_+^m, v_i \in \mathcal{B}_+^n
\]

At t-th round: start from a rank (t-1) \( \hat{F}_{t-1} = U_{t-1}V_{t-1} \) find a rank t NMF

✦ Step 1. Greedy feature selection

\[
(u_t, v_t) = \arg \min_{u \in \mathcal{B}_+^m, v \in \mathcal{B}_+^n} u^\top \left( \nabla_X R(F, X) \bigg|_{\hat{F}_{t-1}} \right) v
\]

✓ Maximizing the decreasing rate of loss function at \( \hat{F}_{t-1} \)

✦ Step 2. Weight update (not on \( \lambda_i \)'s)

\[
U_t = [U_{t-1}, u_t], \quad \tilde{V}_t = [V_{t-1}; v_{t-1}^\top]
\]

\[
W_t = \arg \min_{W_t \in \mathbb{R}_+^{t \times t}} R(F, U_t W_t V_t) \quad t \times t, \text{ convex - easy}
\]

\[
V_t = W_t \tilde{V}_t \quad \hat{F}_t = U_t V_t
\]
Guarantee \( R(F, \hat{F}_t) \leq R(F, \hat{F}_r^*) + ? \)
Guarantee \( R(F, \hat{F}_t) \leq R(F, \hat{F}_r^*) + \) ?

- One round improvement

\[
R(F, \hat{F}_{t-1}) - R(F, \hat{F}_t) \geq \frac{(R(F, \hat{F}_{t-1}) - R(F, \hat{F}_r^*))^2}{2\beta (\sum_{u_i v_i^T \in I^*} \lambda_i^*)^2}
\]
Guarantee \( R(F, \hat{F}_t) \leq R(F, \hat{F}_r^*) + \varepsilon \)

- One round improvement

\[
R(F, \hat{F}_{t-1}) - R(F, \hat{F}_t) \geq \frac{(R(F, \hat{F}_{t-1}) - R(F, \hat{F}_r^*))^2}{2\beta \left( \sum_{u_i v_i^\top \in I^*} \lambda_i^* \right)^2}
\]

- After \( t \) rounds

\[
R(F, \hat{F}_t) \leq \frac{2\beta}{t}
\]

\[
R(F, \hat{F}_t) \leq R(F, \hat{F}_r^*) + \varepsilon, \quad \text{for } t \geq \frac{4\beta(R(F,0) - R(F, \hat{F}_r^*))}{\sigma \varepsilon} r.
\]
Guarantee $R(F, \widehat{F}_t) \leq R(F, \widehat{F}_r^*) + \varepsilon$?

- One round improvement

$$R(F, \widehat{F}_{t-1}) - R(F, \widehat{F}_t) \geq \frac{(R(F, \widehat{F}_{t-1}) - R(F, \widehat{F}_r^*))^2}{2\beta(\sum_{i=1}^{\lambda_i^*} u_i v_i^T \in I^* \lambda_i^*)^2}$$

- After $t$ rounds

$$R(F, \widehat{F}_t) \leq \frac{2\beta}{t}$$

$$R(F, \widehat{F}_t) \leq R(F, \widehat{F}_r^*) + \varepsilon, \quad \text{for } t \geq \frac{4\beta(R(F,0) - R(F, \widehat{F}_r^*))}{\sigma \varepsilon} r.$$ 

- So far, break the original problem into a sequence of “simpler” problems:

$$(u_t, v_t) = \arg \min_{u \in B_+^m, v \in B_+^n} u^\top \left( \nabla_X R(F, X) \big| \widehat{F}_{t-1} \right) v$$

Can we solve the “simpler” problems efficiently?
Rank one problem (matrix)

- Greedy feature selection step
  \[
  \min_{u \in \mathcal{B}_+^m} u^\top \left( \nabla_X R(F, X) \bigg|_{\hat{F}_{t-1}} \right) u
  \]

- Asymmetric case can be reduced to symmetric case
Rank one problem (matrix)

- Greedy feature selection step

\[
\min_{u \in \mathcal{B}_+^m} u^\top \left( \nabla_X R(F, X) \bigg|_{\widehat{F}_{t-1}} \right) u
\]

- SDP relaxation for quadratic program

\[
\min_{X \in \mathbb{R}^{n \times n}_{sym}} \text{Trace}(QX)
\]

such that: \( X \succeq 0, X \) rank one

\[
X_{i,j} \geq 0, \ \forall i, j,
\]

\[
\text{Trace}(X) = 1.
\]

- Asymmetric case can be reduced to symmetric case
Rank one problem (matrix)

- Greedy feature selection step

\[
\min_{u \in \mathcal{B}_+^m} u^\top \left( \nabla_X R(F, X) \bigg|_{\widehat{F}_{t-1}} \right) Q u
\]

- SDP relaxation for quadratic program

\[
\min_{X \in \mathbb{R}_{sym}^{n \times n}} Trace(QX)
\]

such that: \(X \succeq 0\), \(X\) rank one

\[
X_{i,j} \geq 0, \quad \forall i, j,
\]

\[
Trace(X) = 1.
\]

If \(X\) is rank one, then \(X = uu^\top\).

- Asymmetric case can be reduced to symmetric case
Rank one problem (matrix)

- Greedy feature selection step
  \[
  \min_{u \in B_+^m} u^\top \left( \nabla_X R(F, X \middle| \hat{F}_{t-1}) \right)_Q u
  \]

- SDP relaxation for quadratic program

\[
\min_{X \in \mathbb{R}^{n \times n}_{sym}} \text{Trace}(QX)
\]

such that: \(X \succeq 0, X \text{ rank one}\)

\[
X_{i,j} \geq 0, \quad \forall i, j,
\]

\[
\text{Trace}(X) = 1.
\]

If \(X\) is rank one, then \(X = uu^\top\).

- What if SDP solution is not rank one?
  
  ✓ Rank reduction, other relaxation form to enforce rank constraint

- Asymmetric case can be reduced to symmetric case
Rank one problem (tensor)

- Greedy feature selection step
- General polynomial optimization over (multi) positive spheres

\[
\min_{u \in B^n_+} Q\left(\underbrace{u, u, \ldots, u}_d\right)
\]

- Asymmetric case can be reduced to symmetric case
Rank one problem (tensor)

- Greedy feature selection step
  \[
  \min_{u \in B^n_+} Q_{\underbrace{u, \ldots, u}_d}
  \]

- General polynomial optimization over (multi) positive spheres

- Reduce to a QP (auxiliary variables of monomials)

\[
z = \left[ u_1^{d/2}, u_1^{d/2-1}u_2, \ldots, u_1^{d/2-2}u_2u_3, \ldots, u_n^{d/2} \right] \in \mathbb{R}^\tilde{n}
\]

- Asymmetric case can be reduced to symmetric case
Rank one problem (tensor)

- Greedy feature selection step \( \min_{u \in B_+^n} Q(\underbrace{u, u, \ldots, u}_d) \)
- General polynomial optimization over (multi) positive spheres
- Reduce to a QP (auxiliary variables of monomials)
  \[
  z = \left[ u_1^{d/2}, u_1^{d/2-1}u_2, \ldots, u_1^{d/2-2}u_2u_3, \ldots, u_n^{d/2} \right] \in \mathbb{R}^{\tilde{n}}
  \]
- Adopt SDP relaxation \( Z = zz^\top \) monomials of degree \( d \)

\[
\min_{Z \in \mathbb{R}^{\tilde{n} \times \tilde{n}}_{sym}} \text{Trace}(\tilde{Q}Z)
\]

such that: \( Z \succeq 0, \) rank one, \( Z_{i,j} \geq 0, \forall i, j \leq \tilde{n}, \)

\[
\text{Trace}(P_0Z) = \sum_{i_1, \ldots, i_{d/2} \in [n]} u_{i_1}^2 u_{i_2}^2 \ldots u_{i_{d/2}}^2 = 1.
\]
a set of linear consistency constraints

- Asymmetric case can be reduced to symmetric case
Summary before numerical examples

- Two step sequential algorithm
  - Heuristic post processing: prune least important features
  - Use it in complementary to alternating optimization methods
Summary before numerical examples

- Two step sequential algorithm
  - Heuristic post processing: prune least important features
  - Use it in complementary to alternating optimization methods

- Message
  - Tradeoff computation with guaranteed accuracy
  - A class of "Hard" ML problems, non-convex due to latent structure

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>computation</td>
<td>fast</td>
<td>$O(n^{r^2})$</td>
<td>$O(r \ poly(nm))$</td>
</tr>
<tr>
<td>guarantee</td>
<td>start from random initialization, converge to local optima</td>
<td>provably</td>
<td>$R(F, \hat{F}_r) \leq R(F, \hat{F}_s^*) + \epsilon$</td>
</tr>
<tr>
<td>robustness</td>
<td>optimization based, agnostic</td>
<td>assumptions: exact factorization / anchor word / random generation of factors</td>
<td>optimization based, agnostic</td>
</tr>
</tbody>
</table>
Summary before numerical examples

- Two step sequential algorithm
  - Heuristic post processing: prune least important features
  - Use it in complementary to alternating optimization methods

- Message
  - Tradeoff computation with guaranteed accuracy
  - A class of “Hard” ML problems, non-convex due to latent structure
    look for efficient algorithm -- more assumptions, or approximate solution

- Open problems
  - Understand SDP relaxation, variations of relaxation to enforce rank constraint
  - Large scale SDP numerical
  - Proof for guarantee on Greedy + ALS
Numerical example

- Symmetric matrix, n = 60  \( \hat{F}_t = U_t U_t^T \)
Numerical example

- Symmetric matrix, $n = 60$ \( \hat{F}_t = U_t U_t^\top \)

Use sequential algorithm for initial point of alternating improvement

![Graph showing average mean square loss](image)

**Greedy selection + weight update**

**One time ALS improvement**
Numerical example

- Symmetric matrix, $n = 60$ \( \hat{F}_t = U_t U_t^\top \)

Use sequential algorithm for initial point of alternating improvement

Sequential algorithm is exact if the matrix is orthogonally decomposable

Greedy selection + weight update
One time ALS improvement
Numerical example

- Asymmetric matrix, $n = m = 30$
Numerical example

- Asymmetric matrix, $n = m = 30$

Greedy selection + weight update
One ALS improvement
Numerical example

- Asymmetric matrix, \( n = m = 30 \)

![Graph 1](image1.png)

\( n=30, \ m=30, \text{ inner dimension } r \)

Greedy selection + weight update
One ALS improvement

![Graph 2](image2.png)

\( n=30, \ m=30, \text{ inner dimension } r \)

Greedy selection + weight update + ALS
One ALS improvement
Numerical example

- 4-th order symmetric tensor $n = 20$, true rank $r^* = 5$

![Graph showing average mean square loss vs. inner dimension $r$ with two lines: one for Greedy algorithm and another for Greedy + ALS. The graph shows a decreasing trend as $r$ increases.]