Greedy algorithm for large scale

Nonnegative matrix/tensor decomposition

LIDS student conference 2015

Qingqing Huang

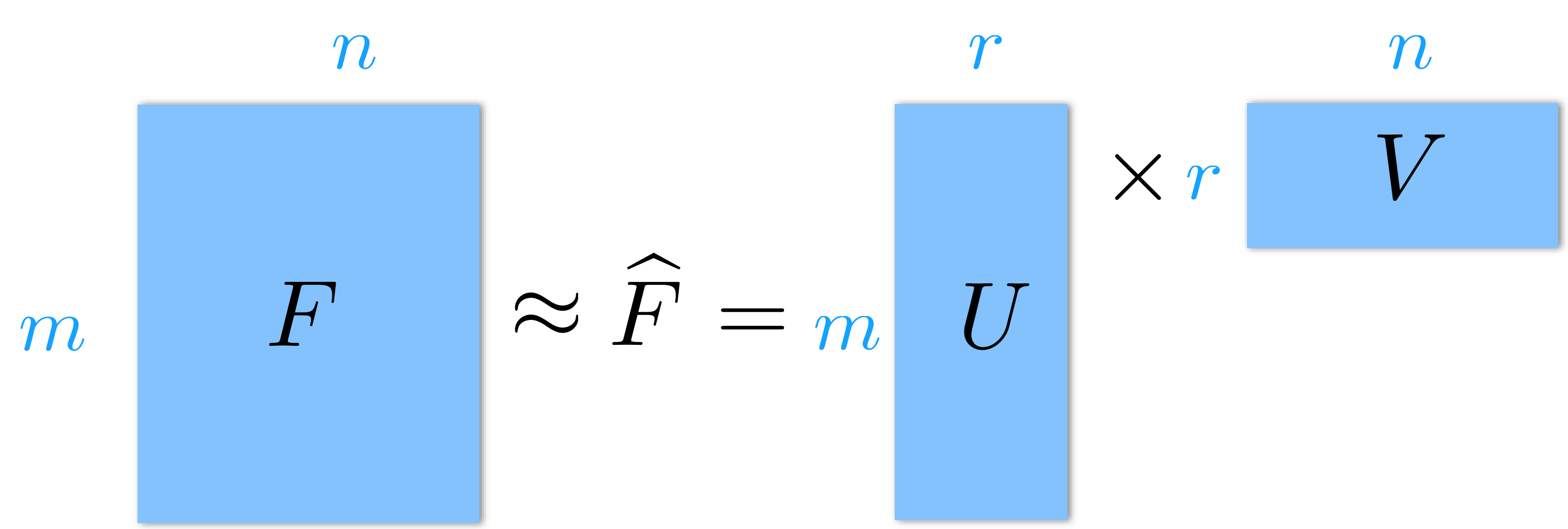
Joint work with Tong Zhang at Baidu

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



Nonnegative **matrix** factorization

- Problem

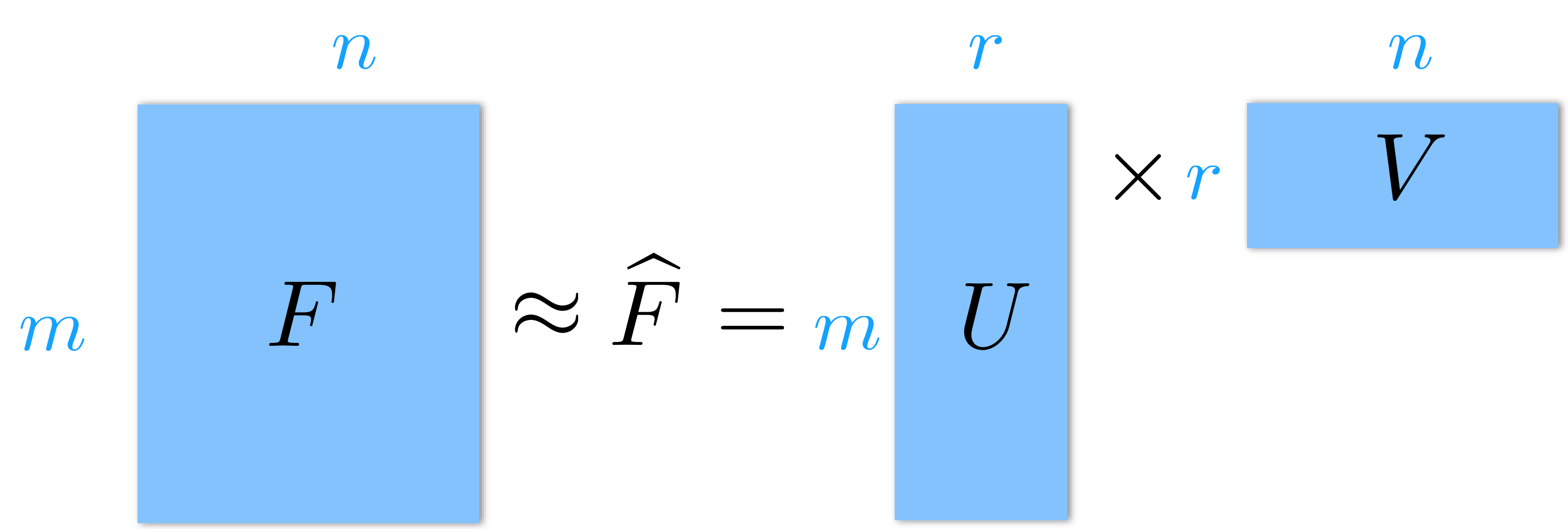
$$\text{Given } F \in \mathbb{R}_+^{n \times m} \text{ and } r, \text{ find } U \in \mathbb{R}_+^{n \times r},\ V \in \mathbb{R}_+^{r \times m} \text{ such that } F \approx UV.$$

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

$\|F - UV\|_F$

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

Regularization for sparsity...



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

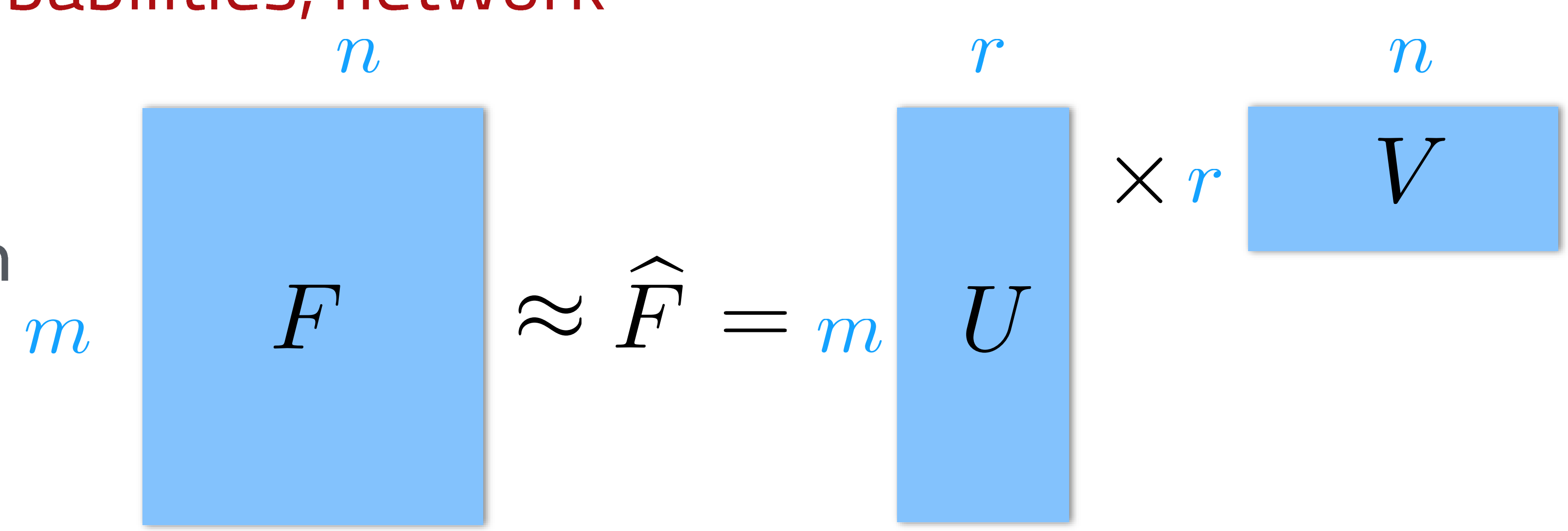
$D_{KL}(F\|UV)$

Regularization for sparsity…

- 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

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

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

$\|F - UV\|_F$

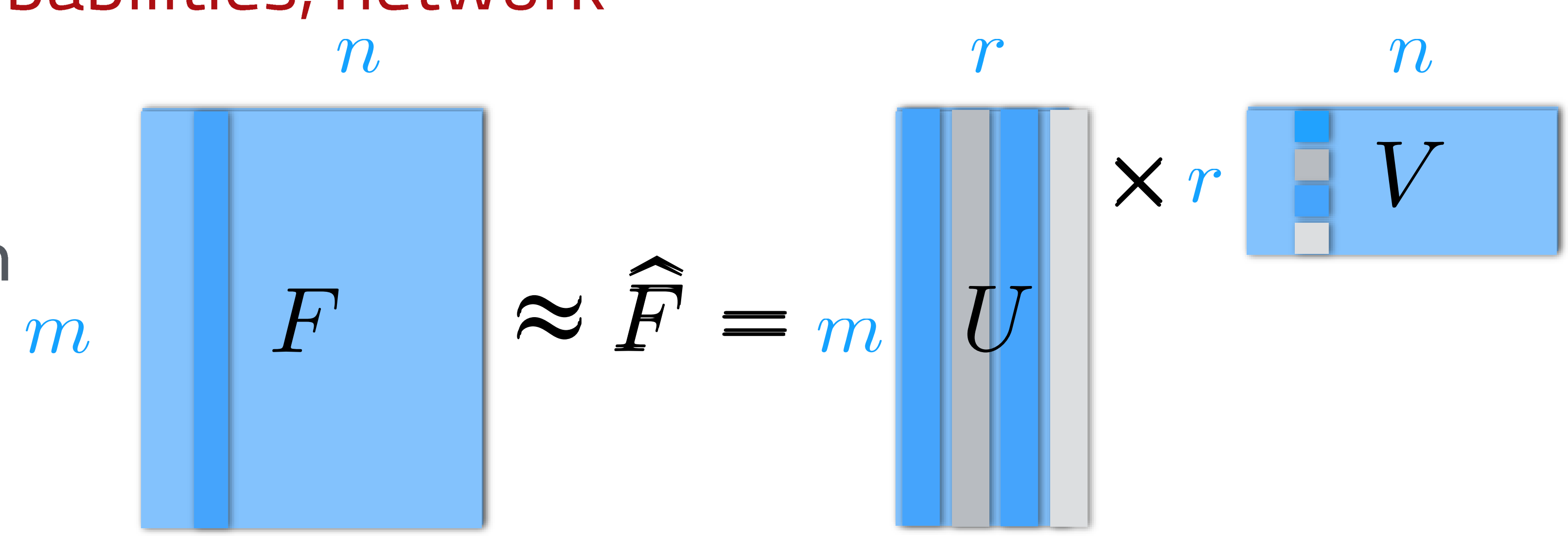
$D_{KL}(F\|UV)$

Regularization for sparsity...

- Applications (why not PCA, Eckart-Young)

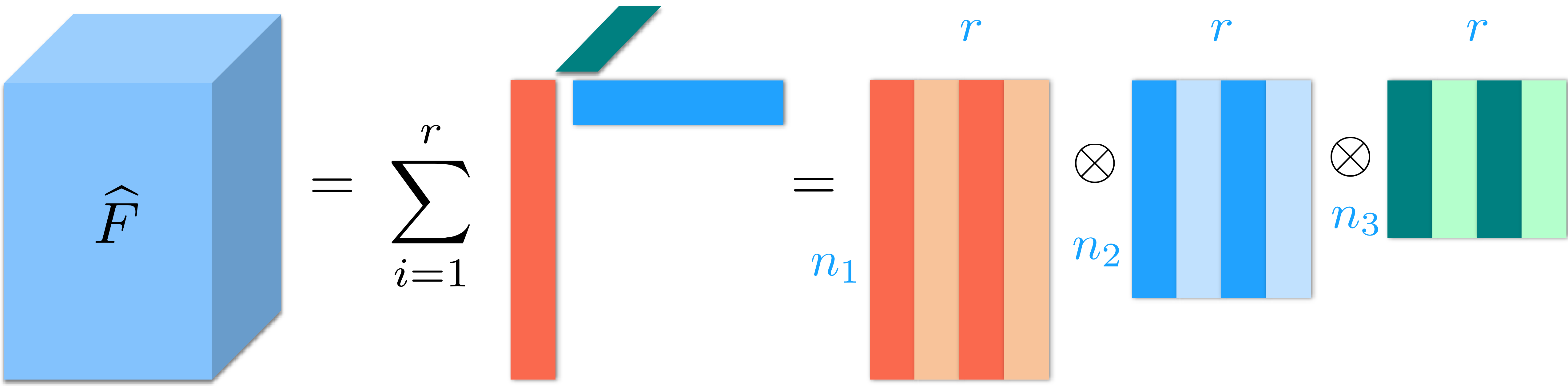
Nonnegative signals, probabilities, network

- ✓ Image compression
- ✓ Sound source separation
- ✓ Spectral clustering
- ✓ Topic model learning
- ✓ Hidden markov model learning



Nonnegative **tensor** factorization

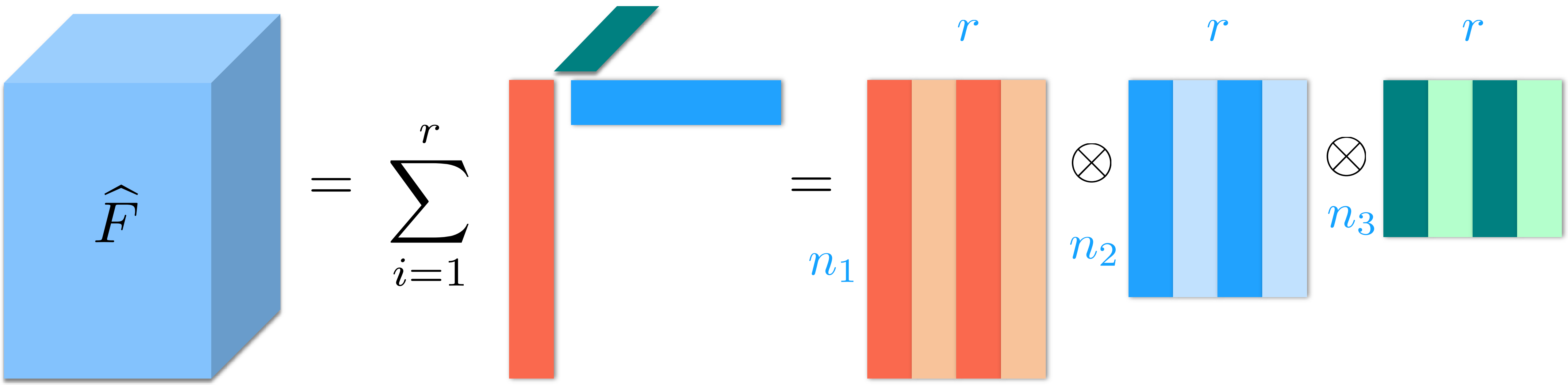
- Problem $\min_{U^{(1)} \in \mathbb{R}_+^{n_1 \times r}, \ldots, U^{(d)} \in \mathbb{R}_+^{n_d \times r}} R(F, U^{(1)} \otimes U^{(2)} \otimes \ldots \otimes U^{(d)})$



- 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)

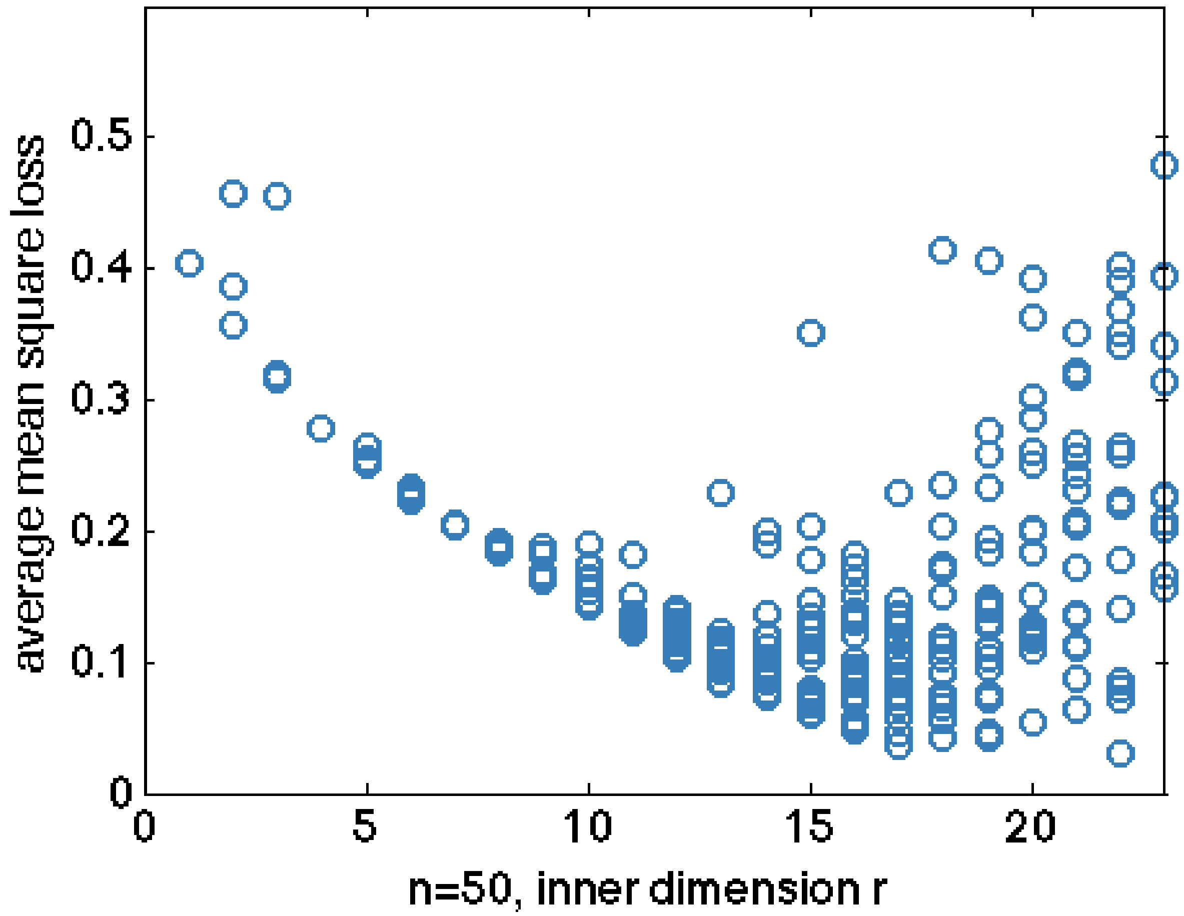
Literature

$$\|F - UV\|_F \qquad D_{KL}(F\|UV)$$

- Alternating optimization method

  high variance especially for large scale problem

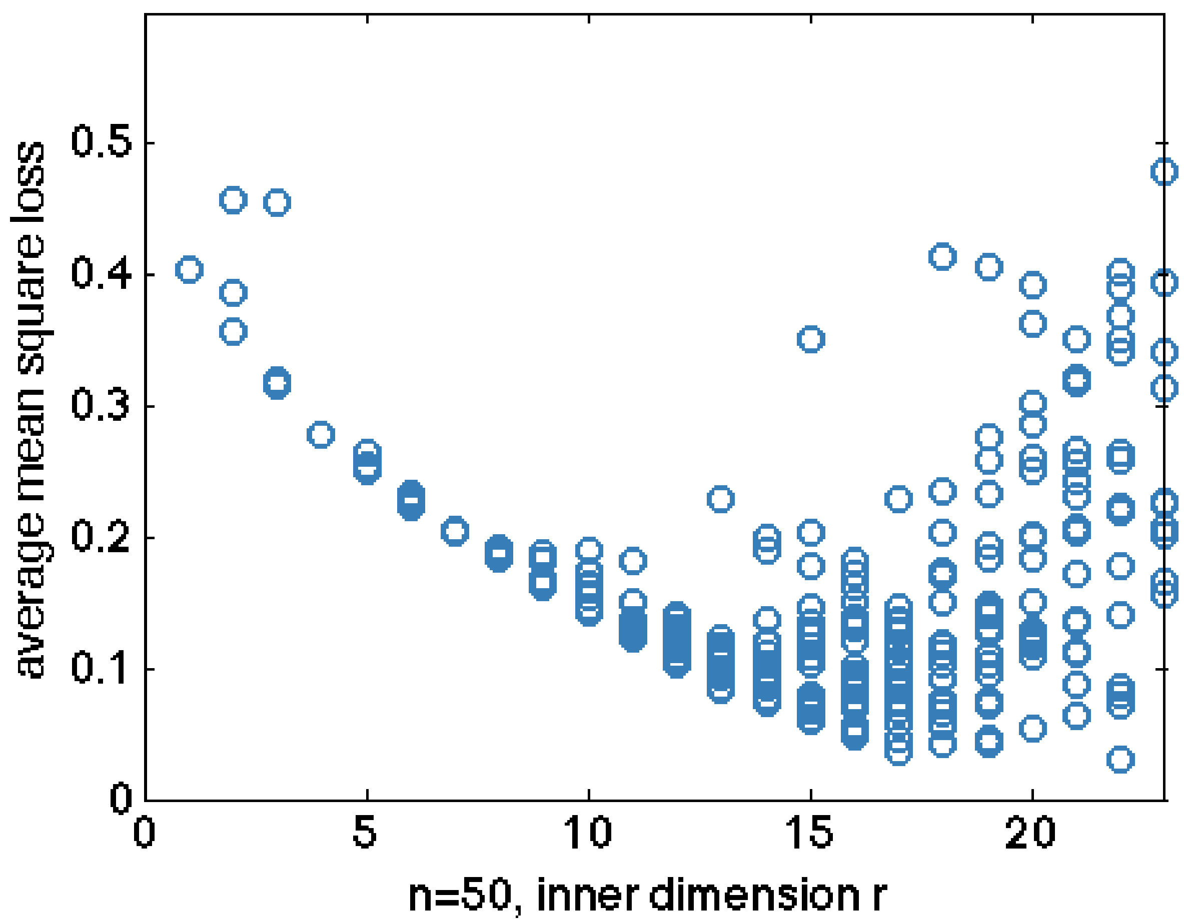
  can we solve it in a more controlled way?



Literature

$$\|F - UV\|_F \qquad D_{KL}(F\|UV)$$

- 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

	Alternating optimization, EM-based algorithm (Lee Seung 2001) (Hoyer 2004)	Exact recovery algorithms (Arora et al 2012) (Recht et al 2012)	Proposed algorithm
computation	fast	$O(n^{r^2})$	$O(r\ \mathrm{poly}(nm))$
guarantee	start from random initialization, converge to local optima	provably	$R(F, \widehat{F}_r) \leq R(F, \widehat{F}_s^*) + \epsilon$
robustness	optimization based, agnostic	assumptions: exact factorization / anchor word / random generation of factors	optimization based, agnostic

Literature

	Alternating optimization, EM-based algorithm (Lee Seung 2001) (Hoyer 2004)	Exact recovery algorithms (Arora et al 2012) (Recht et al 2012)	Proposed algorithm
computation	fast	$O(n^{r^2})$	$O(r \text{ poly}(nm))$
guarantee	start from random initialization, converge to local optima	provably	$R(F, \widehat{F}_r) \leq R(F, \widehat{F}_s^*) + \epsilon$
robustness	optimization based, agnostic	assumptions: exact factorization / anchor word / random generation of factors	optimization based, agnostic

Literature

	Alternating optimization, EM-based algorithm (Lee Seung 2001) (Hoyer 2004)	Exact recovery algorithms (Arora et al 2012) (Recht et al 2012)	Proposed algorithm
computation	fast	$O(n^{r^2})$	$O(r \text{ poly}(nm))$
guarantee	start from random initialization, converge to local optima	provably	$R(F, \widehat{F}_r) \leq R(F, \widehat{F}_s^*) + \epsilon$
robustness	optimization based, agnostic	assumptions: exact factorization / anchor word / random generation of factors	optimization based, agnostic

Literature

	Alternating optimization, EM-based algorithm (Lee Seung 2001) (Hoyer 2004)	Exact recovery algorithms (Arora et al 2012) (Recht et al 2012)	Proposed algorithm
computation	fast	$O(n^{r^2})$	$O(r \text{ poly}(nm))$
guarantee	start from random initialization, converge to local optima	provably	$R(F, \widehat{F}_r) \leq R(F, \widehat{F}_s^*) + \epsilon$
robustness	optimization based, agnostic	assumptions: exact factorization / anchor word / random generation of factors	optimization based, agnostic

Algorithm

- Observation:

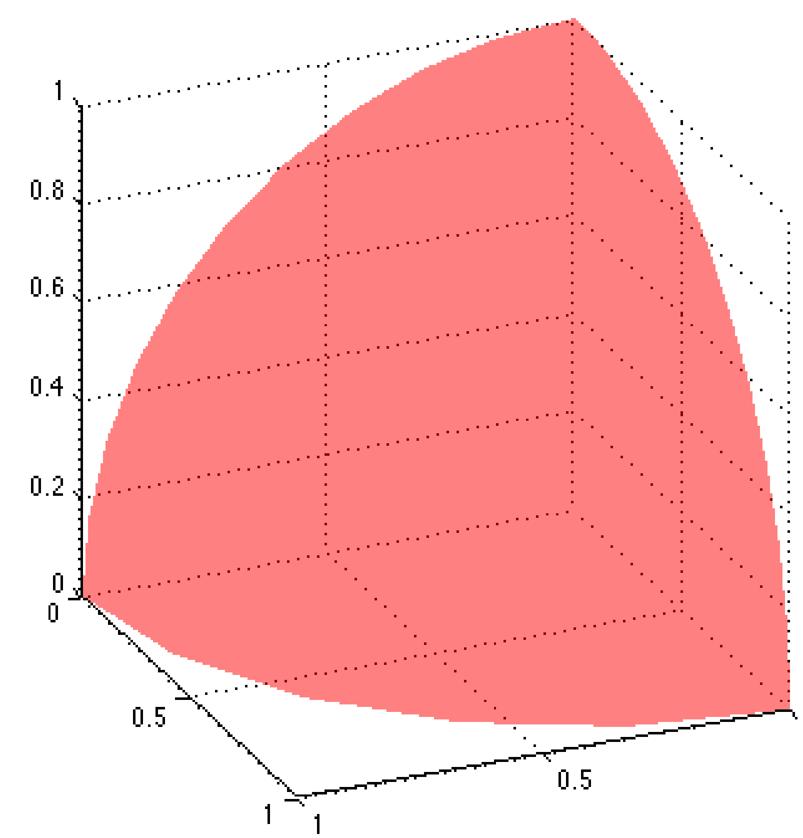
  positive weighted sum of rank-one matrices/tensors

  supported over the sphere in the positive orthant

$$F \approx \widehat{F} = \sum_{i=1}^{r} \lambda_i u_i v_i^\top \qquad u_i \in \mathcal{B}_+^m, v_i \in \mathcal{B}_+^n$$

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

$$F \approx \widehat{F} = \sum_{i=1}^{r} \lambda_i u_i^{(1)} \otimes u_i^{(2)} \otimes \ldots \otimes u_i^{(d)}$$



Algorithm

- Observation:

  positive weighted sum of rank-one matrices/tensors

  supported over the sphere in the positive orthant

$$F \approx \widehat{F} = \sum_{i=1}^{r} \lambda_i u_i v_i^\top \qquad u_i \in \mathcal{B}_+^m, v_i \in \mathcal{B}_+^n$$

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

$$F \approx \widehat{F} = \sum_{i=1}^{r} \lambda_i u_i^{(1)} \otimes u_i^{(2)} \otimes \ldots \otimes u_i^{(d)}$$

- Eckart-Young fails… but can we still find one at a time?
- Greedy feature selection (Frank-Wolfe)

  incremental, greedy, first order method

Incremental algorithm

$$F \approx \widehat{F} = \sum_{i=1}^{r} \lambda_i u_i v_i^\top, \qquad u_i \in \mathcal{B}_+^m, v_i \in \mathcal{B}_+^n$$

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

Incremental algorithm

$$F \approx \widehat{F} = \sum_{i=1}^{r} \lambda_i u_i v_i^\top, \qquad u_i \in \mathcal{B}_+^m, v_i \in \mathcal{B}_+^n$$

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

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

  - ✓ Maximizing the decreasing rate of loss function at $\widehat{F}_{t-1}$

Incremental algorithm

$$F \approx \widehat{F} = \sum_{i=1}^{r} \lambda_i u_i v_i^\top, \qquad u_i \in \mathcal{B}_+^m, v_i \in \mathcal{B}_+^n$$

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

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

✓ Maximizing the decreasing rate of loss function at $\widehat{F}_{t-1}$

- Step 2. Weight update (not on λ_i's)

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

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

Incremental algorithm

$$F \approx \widehat{F} = \sum_{i=1}^{r} \lambda_i u_i v_i^\top, \qquad u_i \in \mathcal{B}_+^m, v_i \in \mathcal{B}_+^n$$

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

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

  ✓ Maximizing the decreasing rate of loss function at $\widehat{F}_{t-1}$

- Step 2. Weight update (not on λ_i's)

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

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

$$V_t = W_t \widetilde{V}_t \qquad \widehat{F}_t = U_t V_t$$

Guarantee $R(F, \widehat{F}_t) \leq R(F, \widehat{F}^*_r) + ?$

Guarantee $R(F,\widehat{F}_t) \leq R(F,\widehat{F}_r^*) + ?$

- One round improvement

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

Guarantee $R(F, \widehat{F}_t) \leq R(F, \widehat{F}_r^*) + ?$

- One round improvement

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

- After t rounds

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

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

Guarantee $R(F, \widehat{F}_t) \leq R(F, \widehat{F}_r^*) + ?$

- One round improvement

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

- After t rounds

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

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

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

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

Can we solve the "simpler" problems efficiently?

Rank one problem (matrix)

- Greedy feature selection step $$\min_{u \in \mathcal{B}_+^m} u^\top \underbrace{\left(\nabla_X R(F, X)\big|_{\widehat{F}_{t-1}} \right)}_{Q} u$$

- Asymmetric case can be reduced to symmetric case

Rank one problem (matrix)

- Greedy feature selection step $\min_{u \in \mathcal{B}_+^m} u^\top \underbrace{\left(\nabla_X R(F, X)\big|_{\widehat{F}_{t-1}} \right)}_{Q} u$
- SDP relaxation for quadratic program

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

$$\begin{aligned} \text{such that: } & X \succeq 0, X \text{ rank one} \\ & X_{i,j} \geq 0, \ \forall i, j, \\ & Trace(X) = 1. \end{aligned}$$

- Asymmetric case can be reduced to symmetric case

Rank one problem (matrix)

- Greedy feature selection step $\min_{u \in \mathcal{B}_{+}^{m}} u^{\top} \underbrace{\left(\nabla_X R(F, X)\big|_{\widehat{F}_{t-1}}\right)}_{Q} u$
- SDP relaxation for quadratic program

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

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

$$X_{i,j} \geq 0, \ \forall i, j,$$

$$Trace(X) = 1.$$

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 $$\min_{u \in \mathcal{B}^m_+} u^\top \underbrace{\left(\nabla_X R(F, X) \big|_{\widehat{F}_{t-1}} \right)}_{Q} u$$
- SDP relaxation for quadratic program

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

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

$$X_{i,j} \geq 0, \ \forall i, j,$$

$$Trace(X) = 1.$$

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 $\min_{u \in \mathcal{B}_+^n} Q(\underbrace{u, u, \ldots, u}_{d})$
- General polynomial optimization over (multi) positive spheres

- Asymmetric case can be reduced to symmetric case

Rank one problem (tensor)

- Greedy feature selection step $\min_{u \in \mathcal{B}_+^n} Q(\underbrace{u, u, \ldots, u}_{d})$
- General polynomial optimization over (multi) positive spheres
- Reduce to a QP (auxiliary variables of monomials)

$$z = \left[u_1^{d/2}, u_1^{d/2-1} u_2, \ldots, u_1^{d/2-2} u_2 u_3, \ldots, u_n^{d/2}\right] \in \mathbb{R}^{\widetilde{n}}$$

- Asymmetric case can be reduced to symmetric case

Rank one problem (tensor)

- Greedy feature selection step $\min_{u \in \mathcal{B}_+^n} Q(\underbrace{u, u, \ldots, u}_{d})$
- General polynomial optimization over (multi) positive spheres
- Reduce to a QP (auxiliary variables of monomials)

$$z = \left[u_1^{d/2}, u_1^{d/2-1} u_2, \ldots, u_1^{d/2-2} u_2 u_3, \ldots, u_n^{d/2}\right] \in \mathbb{R}^{\widetilde{n}}$$

- Adopt SDP relaxation $Z = zz^\top$ monomials of degree d

$$\min_{Z \in \mathbb{R}_{sym}^{\widetilde{n} \times \widetilde{n}}} Trace(\widetilde{Q} Z)$$

$$\text{such that: } Z \succeq 0, \text{ rank one,}$$

$$Z_{i,j} \geq 0, \ \forall i, j \leq \widetilde{n},$$

$$Trace(P_0 Z) = \sum_{i_1, \ldots, i_{d/2} \in [n]} u_{i_1}^2 u_{i_2}^2 \ldots u_{i_{d/2}}^2 = 1.$$

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
    look for efficient algorithm -- more assumptions, or approximate solution

	Alternating optimization, EM-based algorithm (Lee Seung 2001) (Hoyer 2004)	Exact recovery algorithms (Arora et al 2012) (Recht et al 2012)	Proposed algorithm
computation	fast	$O(n^{r^2})$	$O(r \text{ poly}(nm))$
guarantee	start from random initialization, converge to local optima	provably	$R(F, \widehat{F}_r) \leq R(F, \widehat{F}_s^*) + \epsilon$
robustness	optimization based, agnostic	assumptions: exact factorization / anchor word / random generation of factors	optimization based, agnostic

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

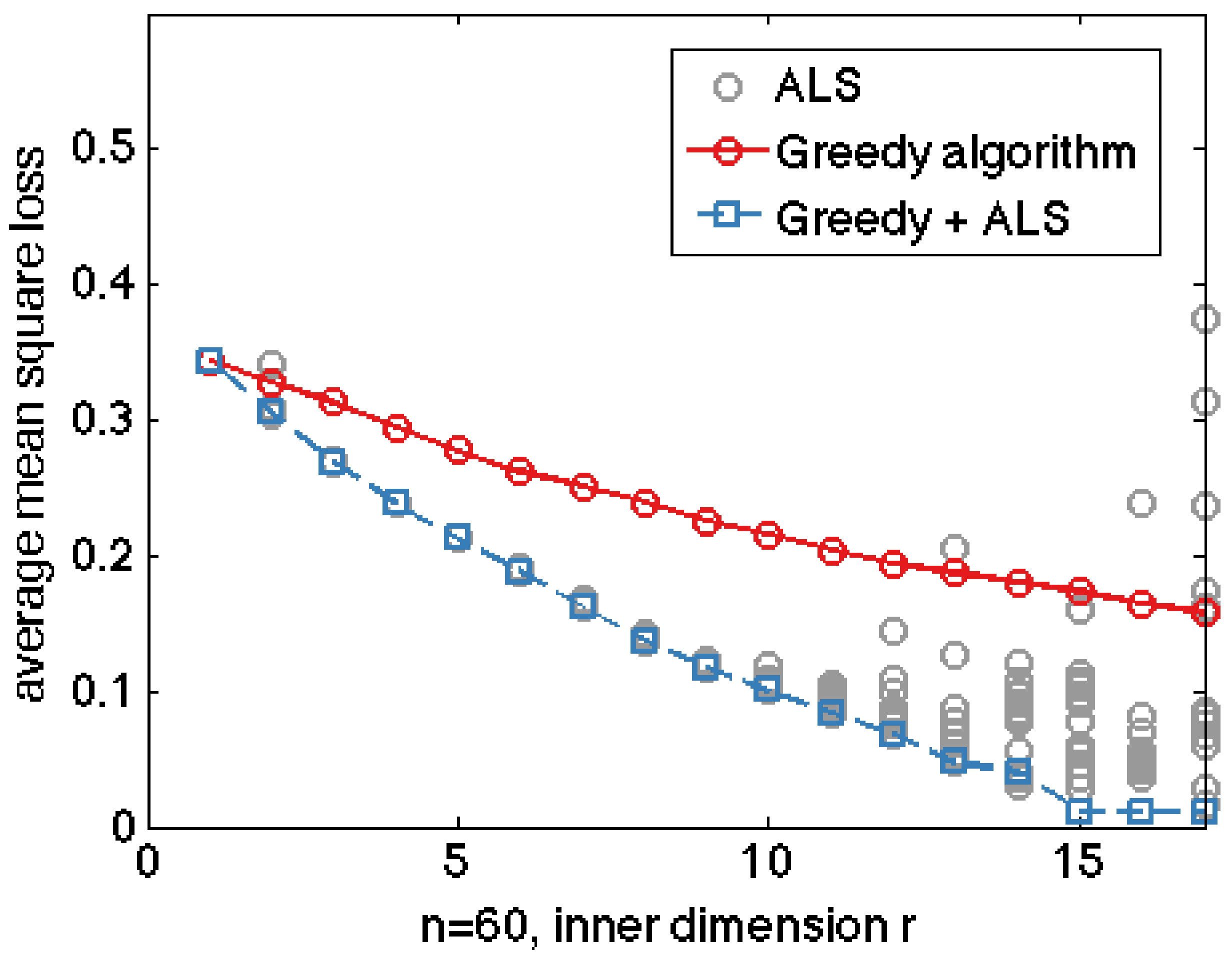
Numerical example

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

Numerical example

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

Use sequential algorithm for initial point of alternating improvement



Greedy selection + weight update

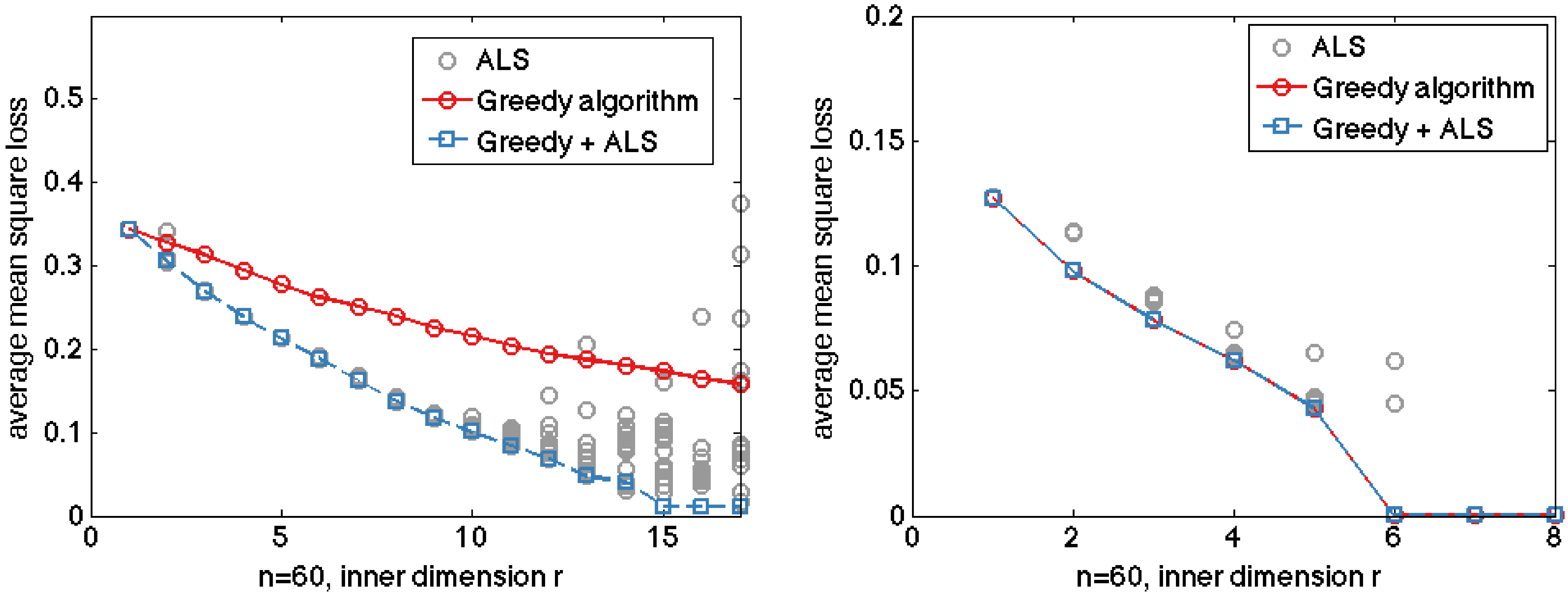
One time ALS improvement

Numerical example

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

Use sequential algorithm for initial point of alternating improvement

Sequential algorithm is exact if the matrix is orthogonally decomposable



Greedy selection + weight update

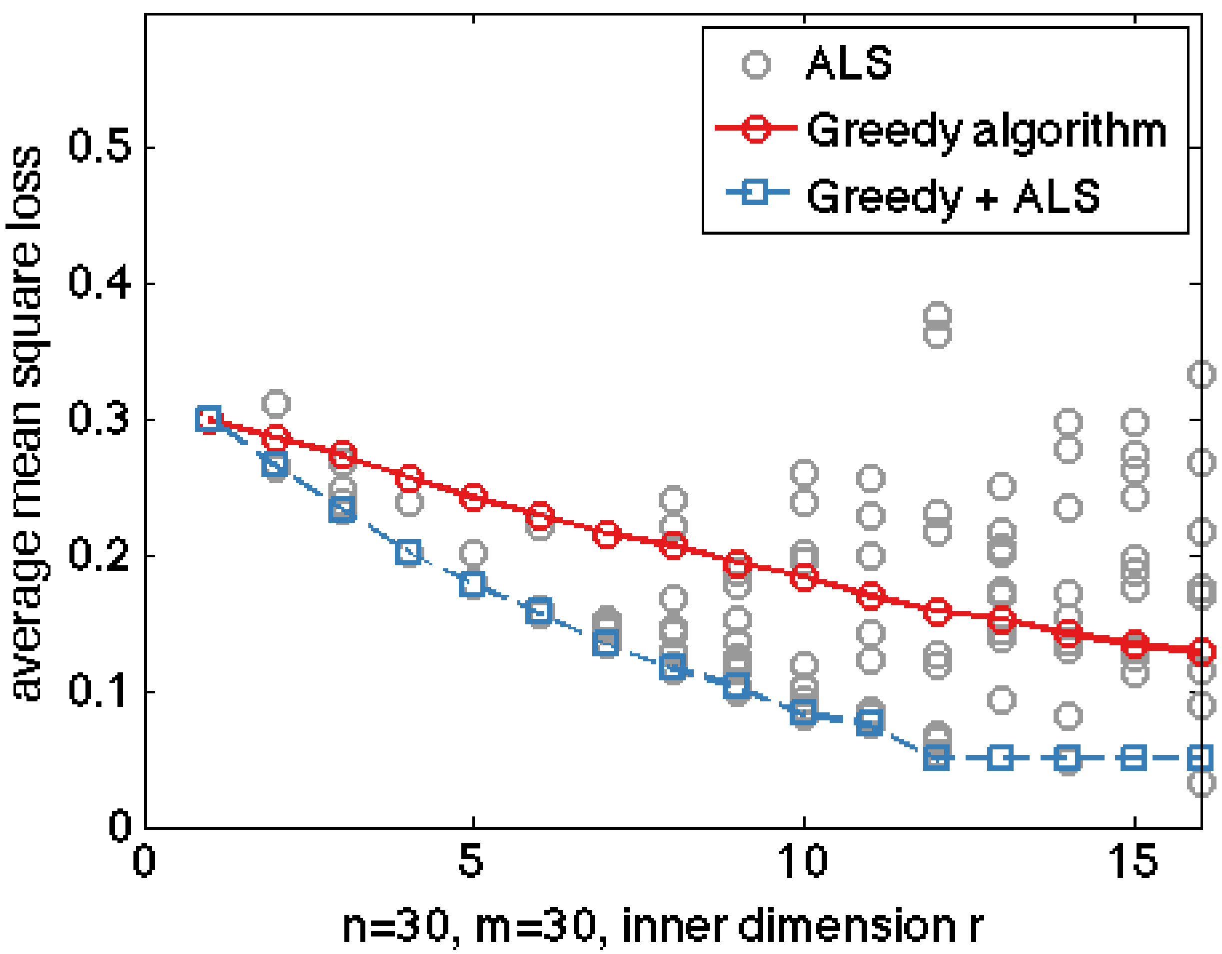
One time ALS improvement

Numerical example

- Asymmetric matrix, $n = m = 30$

Numerical example

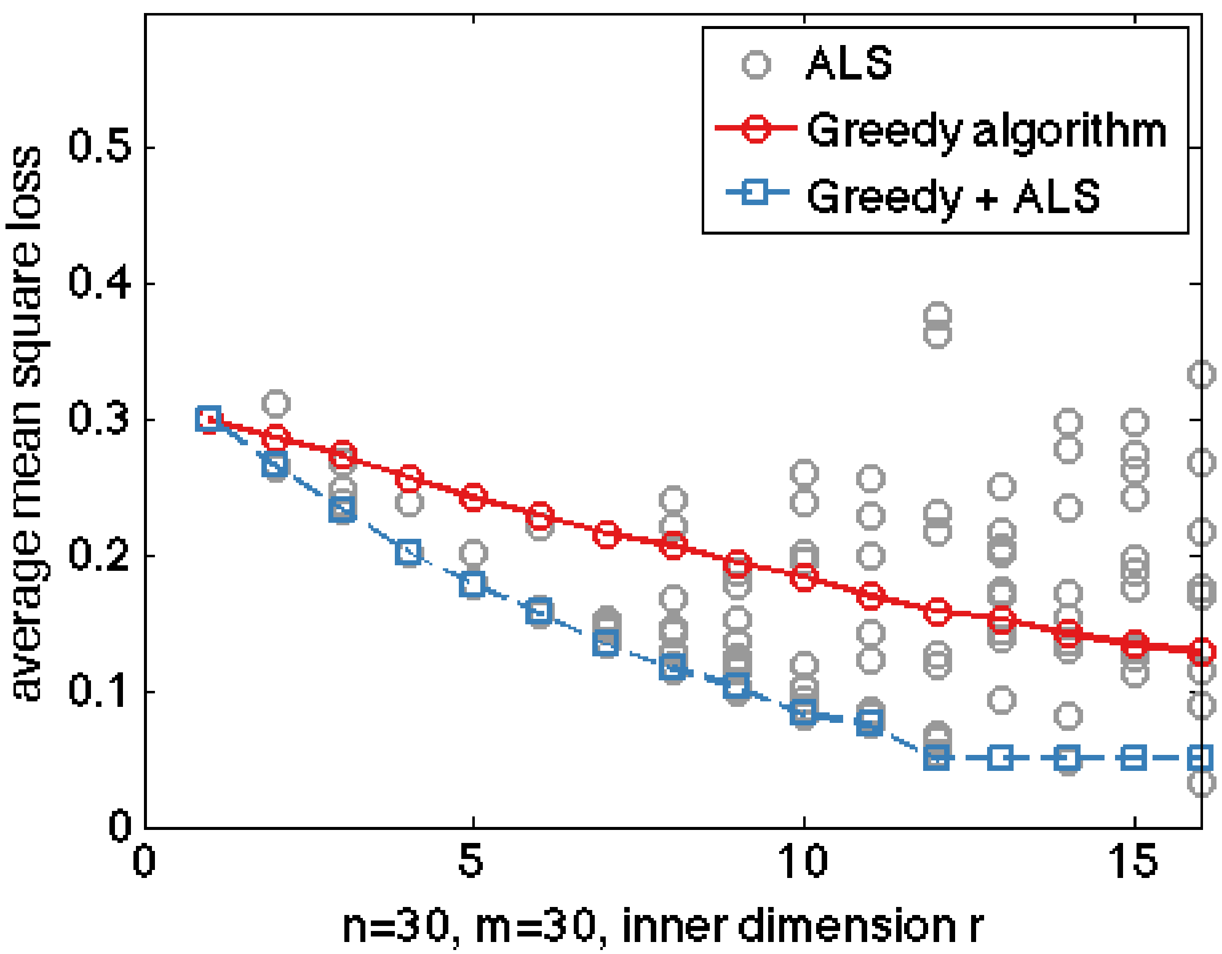
- Asymmetric matrix, n = m = 30

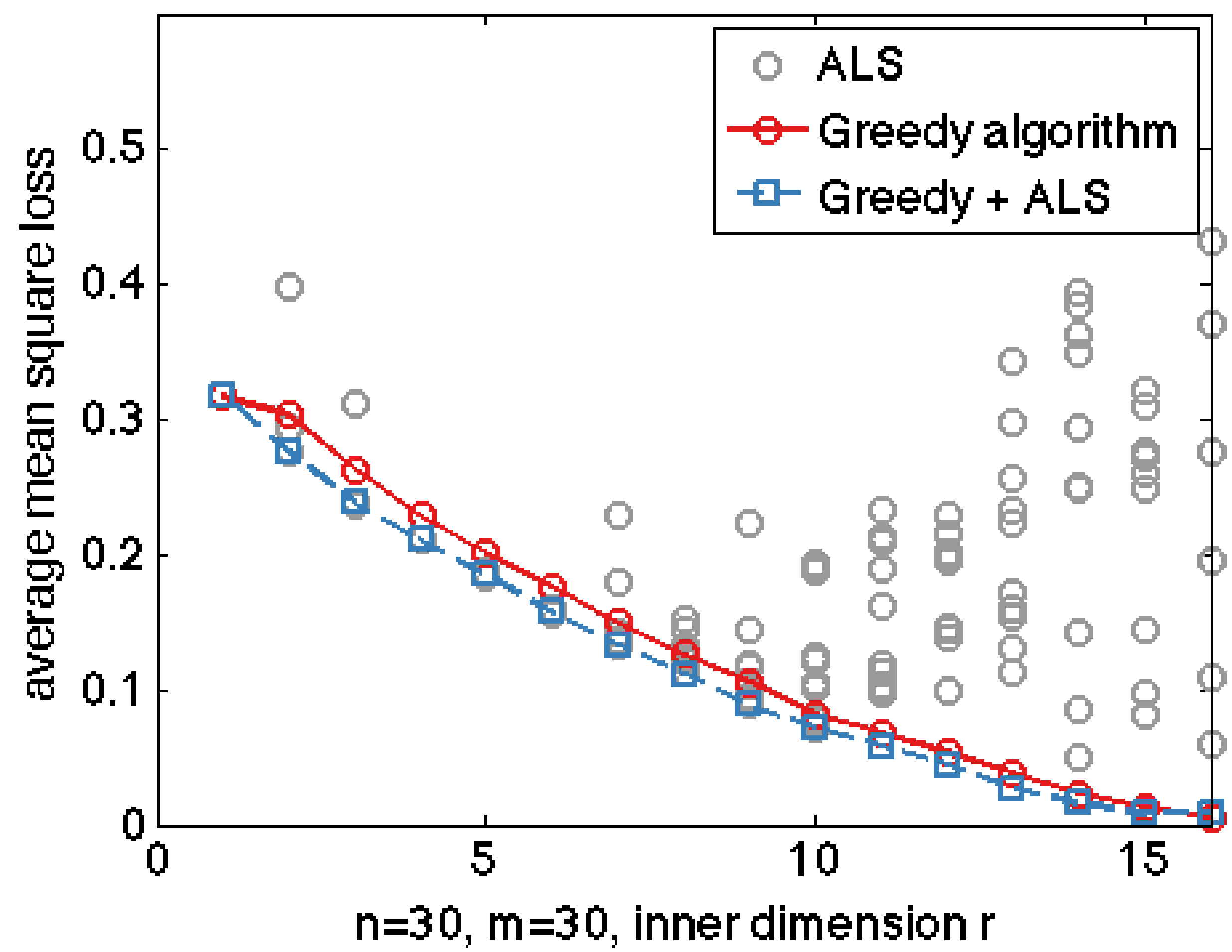


Greedy selection + weight update
One ALS improvement

Numerical example

- Asymmetric matrix, n = m = 30



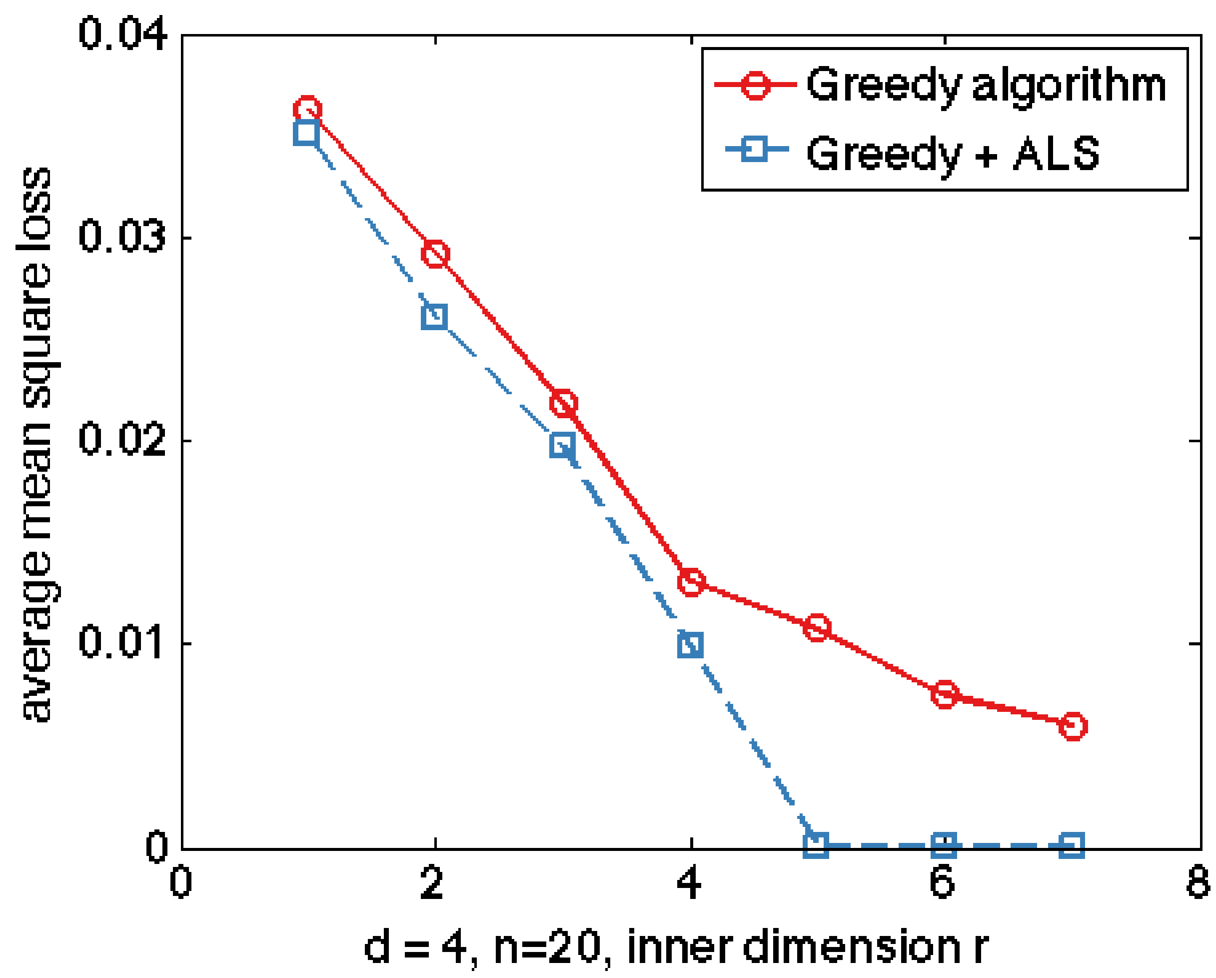


Greedy selection + weight update
One ALS improvement

Greedy selection + weight update + ALS
One ALS improvement

Numerical example

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



Thank you

LIDS student conference 2015