Estimation of low-rank tensors via convex optimization

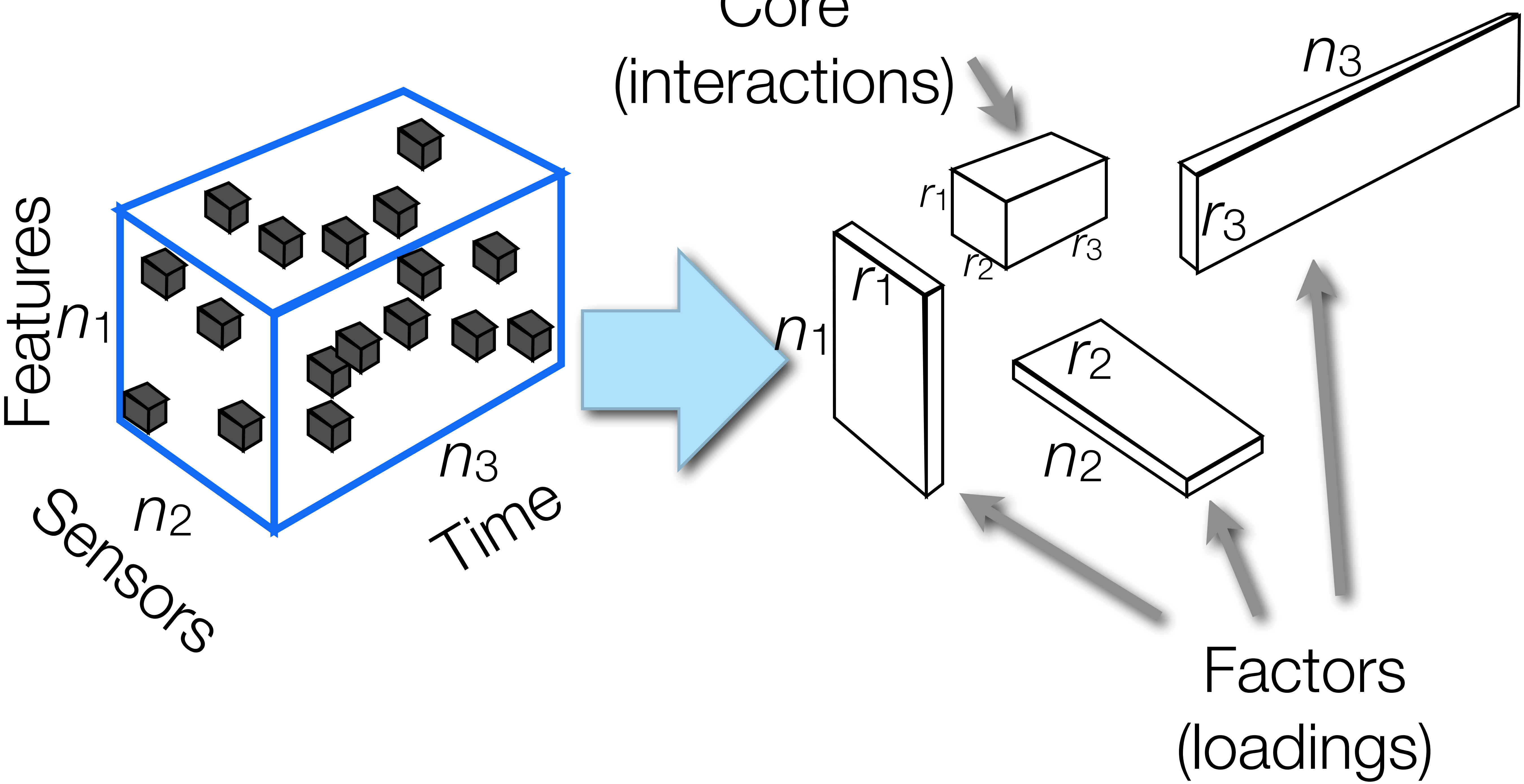
Ryota Tomioka[1], Kohei Hayashi[2], Hisashi Kashima[1]

[1]The University of Tokyo
[2]Nara Institute of Science and Technology

2011/3/23 @ TU Berlin

Convex low-rank *tensor* completion

$$X_{ijk} = \sum_{a=1}^{r_1} \sum_{b=1}^{r_2} \sum_{c=1}^{r_3} C_{abc} U_{ia}^{(1)} U_{jb}^{(2)} U_{kc}^{(c)}$$



Conventional formulation (nonconvex)

$$\underset{\mathcal{C},\boldsymbol{U}_1,\boldsymbol{U}_2,\boldsymbol{U}_3}{\text{minimize}} \|\Omega \circ (\mathcal{Y} - \mathcal{C} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3)\|_F^2 + \text{regularization.}$$

observation — mode-k product

$$\underset{\mathcal{X}}{\text{minimize}} \|\Omega \circ (\mathcal{Y} - \mathcal{X})\|_F^2 \quad \text{s.t.} \quad \text{rank}(\mathcal{X}) \leq (r_1, r_2, r_3).$$

- Alternate minimization
- Have to fix the rank beforehand

Our approach

Matrix

Estimation of
low-rank matrix
(hard)

Trace norm
minimization
(tractable)
[Fazel, Hindi, Boyd 01]

Generalization

Tensor

Estimation of
low-rank tensor
(hard)
Rank defined in the sense of
Tucker decomposition

Extended
trace norm
minimization
(tractable)

Trace norm (nuclear norm) regularization

$$\boldsymbol{X} \in \mathbb{R}^{R \times C}, \quad m = \min(R, C)$$

$$\|\boldsymbol{X}\|_* = \sum_{j=1}^{m} \sigma_j(\boldsymbol{X})$$

Linear sum of singular-values

- Roughly speaking, L1 regularization on the singular-values.
- Stronger regularization --> more zero singular-values --> low rank.
- Not obvious for tensors (no singular-values for tensors)

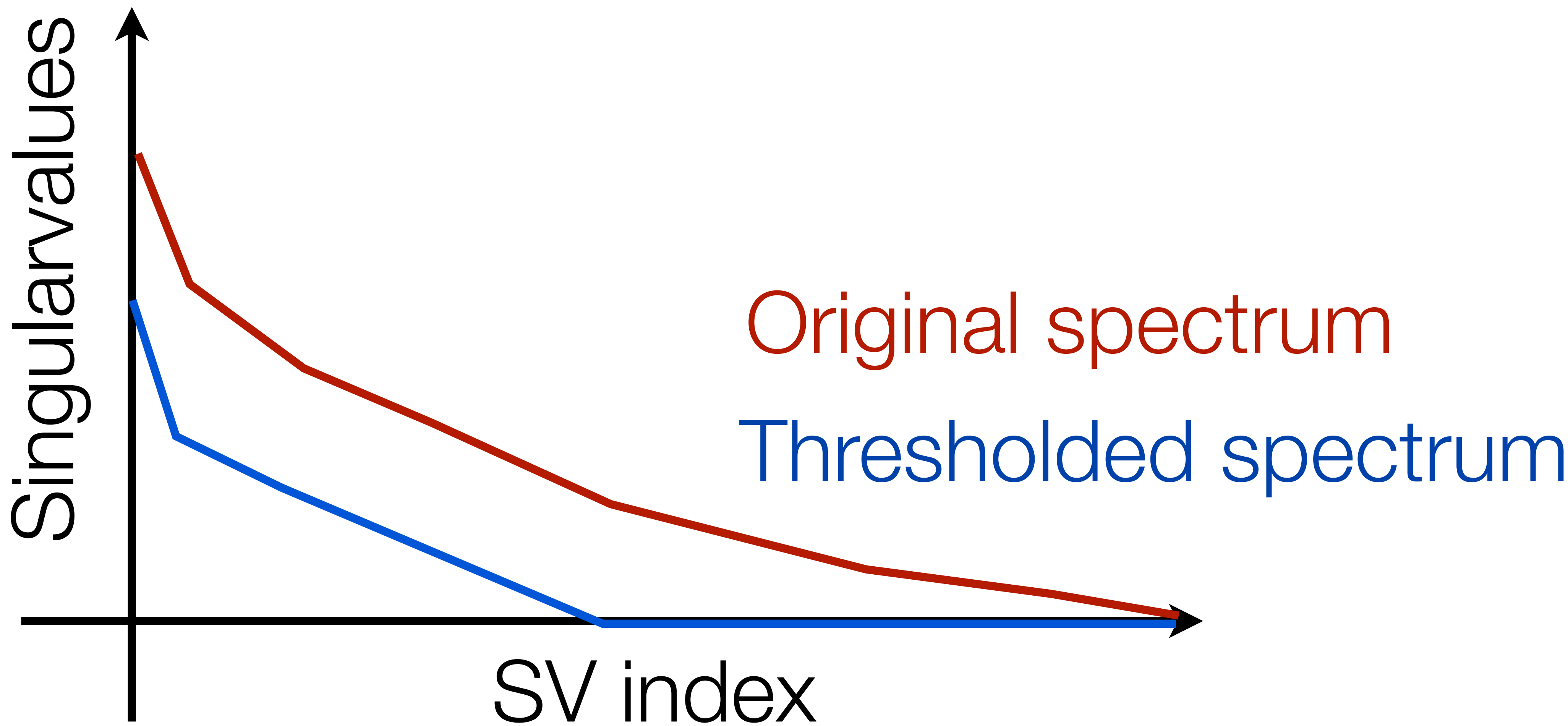
Spectral soft-threshold operation

all *observed* and *matrix* --> analytic solution

$$\mathrm{softth}(\boldsymbol{X}) = \operatorname*{argmin}_{\boldsymbol{Z}\in\mathbb{R}^{R\times C}} \left(\frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{X}\|_F^2 + \lambda\|\boldsymbol{Z}\|_*\right)$$

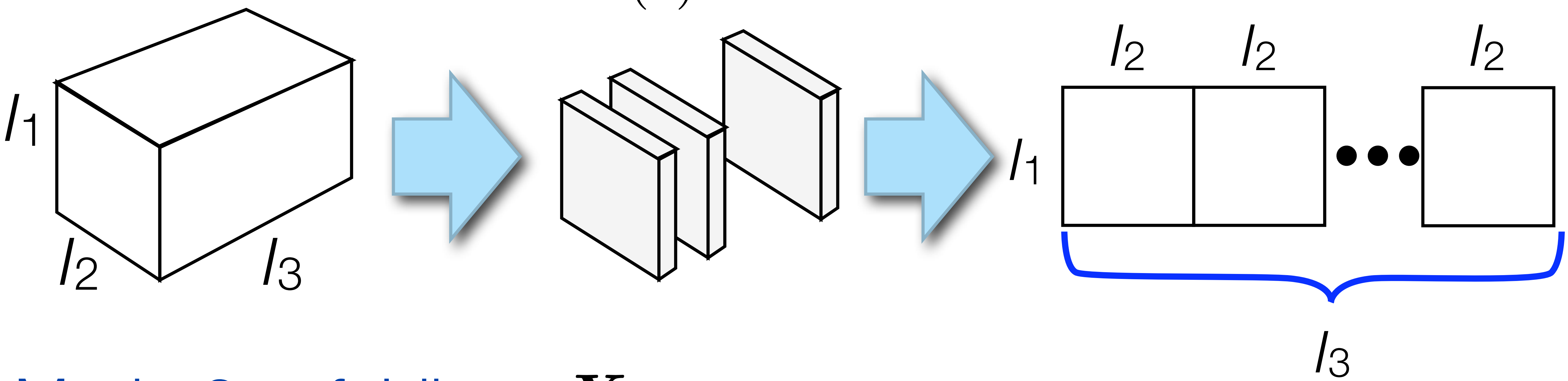
$$= \boldsymbol{U}\max(\boldsymbol{S}-\lambda, 0)\boldsymbol{V}^\top$$

where $X=USV^\top$

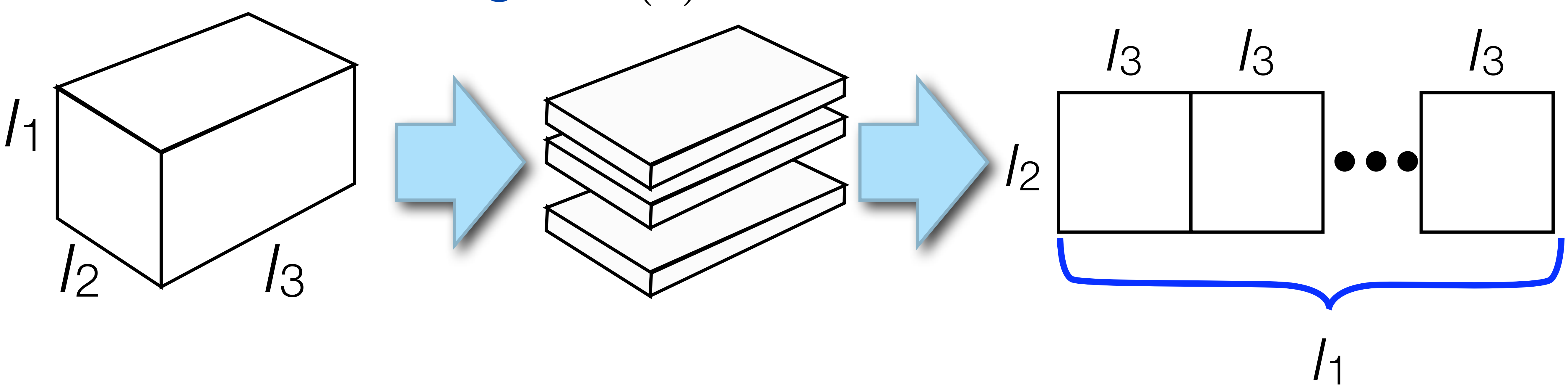


Mode-k unfolding (matricization)

Mode-1 unfolding $\boldsymbol{X}_{(1)}$

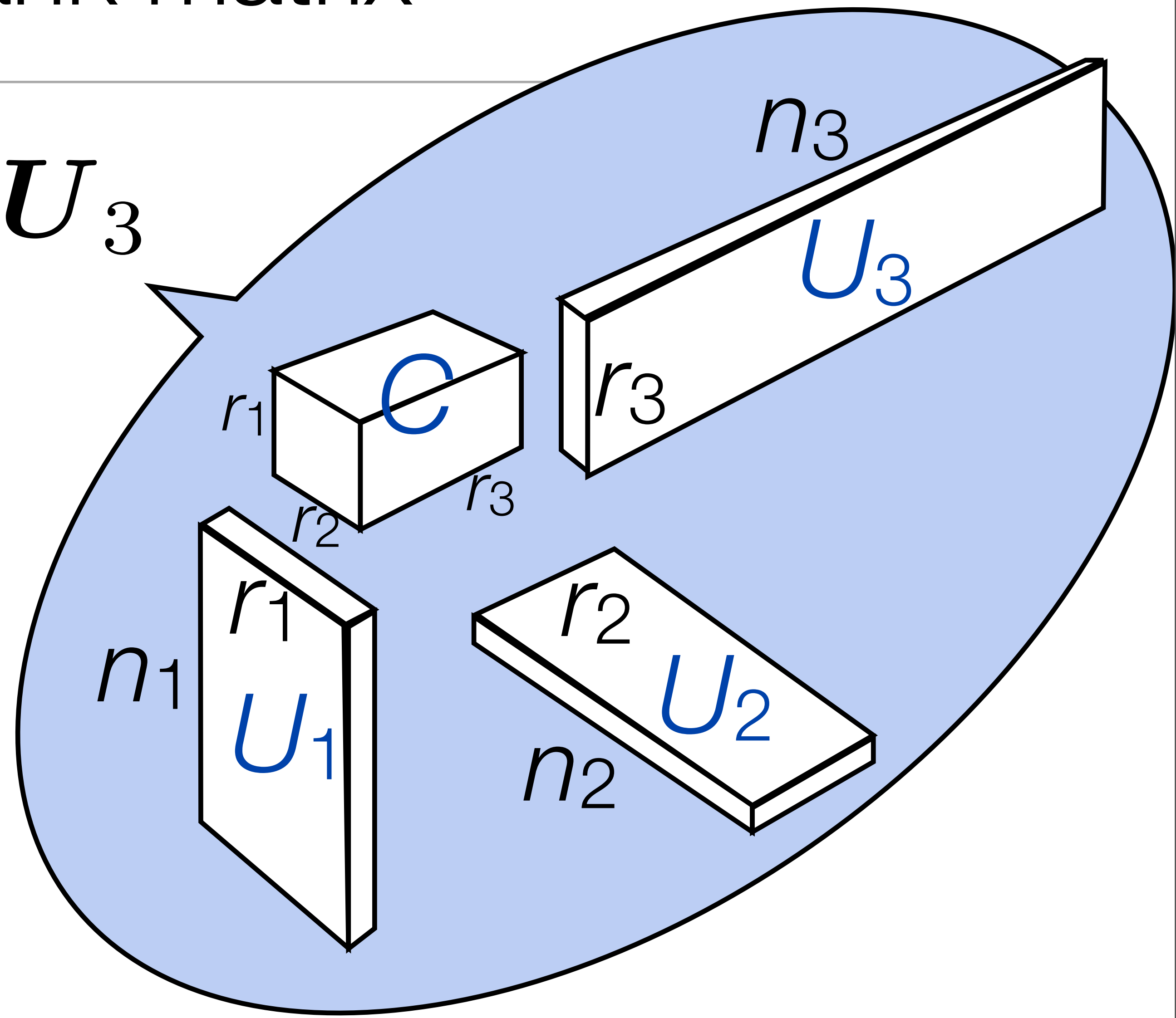


Mode-2 unfolding $\boldsymbol{X}_{(2)}$



Low-rank tensor is a low-rank matrix

$$\mathcal{X} = \mathcal{C} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3$$



Mode-1 unfolding

$$\boldsymbol{X}_{(1)} = \boldsymbol{U}_1 \boldsymbol{C}_{(1)} (\boldsymbol{U}_3 \otimes \boldsymbol{U}_2)^\top$$

rank ≦ r_1

Mode-2 unfolding

$$\boldsymbol{X}_{(2)} = \boldsymbol{U}_2 \boldsymbol{C}_{(2)} (\boldsymbol{U}_1 \otimes \boldsymbol{U}_3)^\top$$

rank ≦ r_2

Mode-3 unfolding

$$\boldsymbol{X}_{(3)} = \boldsymbol{U}_3 \boldsymbol{C}_{(3)} (\boldsymbol{U}_2 \otimes \boldsymbol{U}_1)^\top$$

rank ≦ r_3

The rank of $X_{(k)}$ is no more than the rank of $C_{(k)}$

Low-rank matrix is a low-rank tensor

- Given X=USV$^\top$ (low-rank)
- Define

$$\mathcal{C} = SV^\top$$

$$U_1 = U$$

$$U_2 = I_{n_2}$$

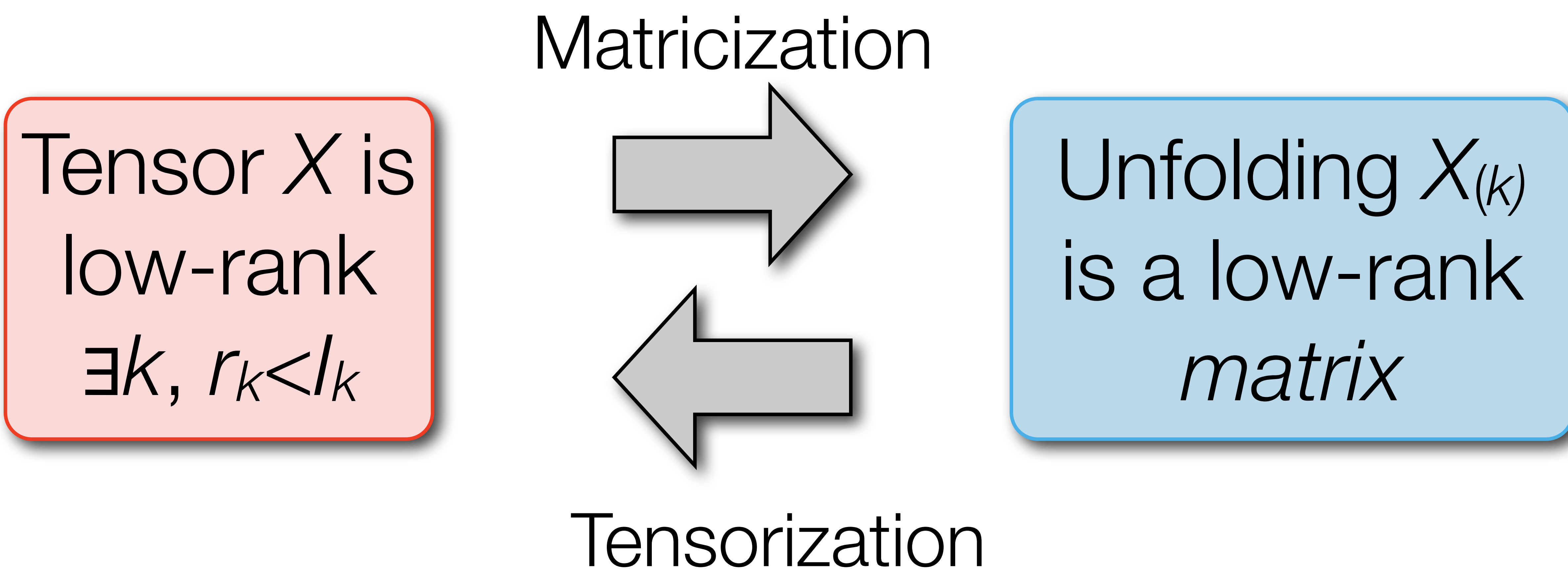
$$U_3 = I_{n_3}$$

$$\mathcal{X} = \mathcal{C} \times_1 U_1 \times_2 U_2 \times_3 U_3 \quad \text{is low-rank}$$

(at least for mode 1)

What it means

- We can use the trace norm of an unfolding of a tensor *X* to learn low-rank *X*.



Approach 1: As a matrix

- Pick a mode *k*, and hope that the tensor to be learned is low rank in mode *k*.

$$\underset{\mathcal{X}\in\mathbb{R}^{I_1\times\cdots\times I_K}}{\text{minimize}} \quad \frac{1}{2\lambda}\|\Omega\circ(\mathcal{Y}-\mathcal{X})\|_F^2+\|\boldsymbol{X}_{(k)}\|_*,$$

Pro: Basically a matrix problem

→ Theoretical guarantee (Candes & Recht 09)

Con: Have to be lucky to pick the right mode.

Approach 2: Constrained optimization

- Constrain so that each unfolding of *X* is simultaneously low rank.

$$\underset{\mathcal{X}\in\mathbb{R}^{I_1\times\cdots\times I_K}}{\text{minimize}} \quad \frac{1}{2\lambda}\|\Omega\circ(\mathcal{Y}-\mathcal{X})\|_F^2+\sum_{k=1}^{K}\gamma_k\|\boldsymbol{X}_{(k)}\|_*.$$

Pro: Jointly regularize every mode
Con: Strong constraint

γ_k: tuning parameter usually set to 1.

(See also Signoretto et al.,10; Gandy et al. 11)

Approach 3: Mixture of low-rank tensors

- Each mixture component Z_k is regularized to be low-rank only in mode-k.

$$\underset{\mathcal{Z}_1,\ldots,\mathcal{Z}_K}{\text{minimize}} \quad \frac{1}{2\lambda}\left\|\Omega \circ \left(\mathcal{Y} - \sum_{k=1}^{K} \mathcal{Z}_k\right)\right\|_F^2 + \sum_{k=1}^{K} \gamma_k \|\boldsymbol{Z}_{k(k)}\|_*,$$

Pro: Each Z_k takes care of each mode

Con: Sum is not low-rank

Optimization via Alternating Direction Method of Multipliers (ADMM)

(Gabay & Mercier 76)

- Useful when we have linear operation inside sparsity penalty

$$\underset{\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_K}}{\text{minimize}} \quad \frac{1}{2\lambda} \|\Omega(\mathcal{X}) - \boldsymbol{y}\|_F^2 + \sum_{k=1}^{K} \gamma_k \|\boldsymbol{X}_{(k)}\|_*.$$

Permutation

Optimization via Alternating Direction Method of Multipliers (ADMM)

(Gabay & Mercier 76)

- Useful when we have linear operation inside sparsity penalty

$$\underset{\mathcal{X}\in\mathbb{R}^{n_1\times\cdots\times n_K}}{\text{minimize}} \quad \frac{1}{2\lambda}\|\Omega(\mathcal{X})-\boldsymbol{y}\|_F^2+\sum_{k=1}^{K}\gamma_k\|\boldsymbol{X}_{(k)}\|_*.$$

Permutation

- Split Bregman Iteration (Goldstein & Osher) is also an ADMM

Total-variation image reconstruction:

$$\underset{\boldsymbol{x}\in\mathbb{R}^{n}}{\text{minimize}} \quad \frac{1}{2\lambda}\|\Omega(\boldsymbol{x})-\boldsymbol{y}\|^2+\sum_{j=1}^{n}\|D_j\boldsymbol{x}\|$$

2D derivative at *j*th pixel

ADMM preliminaries

- **Problem**

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x})$$

Linear operation

ADMM preliminaries

- **Problem**

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x})$$

Linear operation

ADMM preliminaries

- Problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x})$$

Linear operation

- Step 1: Split & Augment

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{z}}{\text{minimize}} \quad & f(\boldsymbol{x}) + g(\boldsymbol{z}) + \frac{\eta}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|^2 \\ \text{subject to} \quad & \boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} \end{aligned}$$

ADMM preliminaries

- Problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x})$$

Linear operation

- Step 1: Split & Augment

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{z}}{\text{minimize}} \quad & f(\boldsymbol{x}) + g(\boldsymbol{z}) + \frac{\eta}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|^2 \\ \text{subject to} \quad & \boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} \end{aligned}$$

ADMM preliminaries

- Problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x})$$

Linear operation

- Step 1: Split & Augment

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{z}}{\text{minimize}} \quad & f(\boldsymbol{x}) + g(\boldsymbol{z}) + \frac{\eta}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|^2 \\ \text{subject to} \quad & \boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} \end{aligned}$$

ADMM preliminaries

- Problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x})$$

Linear operation

- Step 1: Split & Augment

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{z}}{\text{minimize}} \quad & f(\boldsymbol{x}) + g(\boldsymbol{z}) + \frac{\eta}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|^2 \\ \text{subject to} \quad & \boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} \end{aligned}$$

- Step 2: Augmented Lagrangian function

$$L_\eta(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + g(\boldsymbol{z}) + \boldsymbol{\alpha}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}) + \frac{\eta}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|^2$$

Ordinary Lagrangian

Augmented

ADMM algorithm (Gabay & Mercier 76)

- Minimize the AL function wrt X

$$\boldsymbol{x}^{t+1} = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\text{argmin}}\, L_\eta(\boldsymbol{x}, \boldsymbol{z}^t, \boldsymbol{\alpha}^t),$$

- Minimize the AL function wrt Z

$$\boldsymbol{z}^{t+1} = \underset{\boldsymbol{z} \in \mathbb{R}^m}{\text{argmin}}\, L_\eta(\boldsymbol{x}^{t+1}, \boldsymbol{z}, \boldsymbol{\alpha}^t),$$

- Update the multiplier vector

$$\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t + \eta(\boldsymbol{A}\boldsymbol{x}^{t+1} - \boldsymbol{z}^{t+1}).$$

ADMM algorithm (Gabay & Mercier 76)

- Minimize the AL function wrt X

$$\boldsymbol{x}^{t+1} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} L_\eta(\boldsymbol{x}, \boldsymbol{z}^t, \boldsymbol{\alpha}^t),$$

- Minimize the AL function wrt Z

$$\boldsymbol{z}^{t+1} = \operatorname*{argmin}_{\boldsymbol{z}\in\mathbb{R}^m} L_\eta(\boldsymbol{x}^{t+1}, \boldsymbol{z}, \boldsymbol{\alpha}^t),$$

- Update the multiplier vector

$$\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t + \eta(\boldsymbol{A}\boldsymbol{x}^{t+1} - \boldsymbol{z}^{t+1}).$$

Every limit point of ADMM is a minimizer of the original problem. [Eckstein & Bertsekas 92]

For approach "Constraint"

- Move the permutation out of the regularizer

$$\begin{aligned} \underset{\mathcal{X}, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_K}{\text{minimize}} \quad & \frac{1}{2\lambda}\|\Omega(\mathcal{X}) - \boldsymbol{y}\|^2 + \sum_{k=1}^{K} \gamma_k \|\boldsymbol{Z}_k\|_*, \\ \text{subject to} \quad & \boldsymbol{X}_{(k)} = \boldsymbol{Z}_k \quad (k = 1, \ldots, K), \end{aligned}$$

- Augmented Lagrangian:

$$\begin{aligned} L_\eta(\mathcal{X}, \{\boldsymbol{Z}_k\}_{k=1}^K, \{\boldsymbol{A}_k\}_{k=1}^K) = & \frac{1}{2\lambda}\|\Omega(\mathcal{X}) - \boldsymbol{y}\|^2 + \sum_{k=1}^{K} \gamma_k \|\boldsymbol{Z}_k\|_* \\ & + \eta \sum_{k=1}^{K} \left(\langle \boldsymbol{A}_k, \boldsymbol{X}_{(k)} - \boldsymbol{Z}_k \rangle + \frac{1}{2}\|\boldsymbol{X}_{(k)} - \boldsymbol{Z}_k\|_F^2 \right). \end{aligned}$$

ADMM for "Constraint" $(\lambda \to 0)$

- Minimize the AL function wrt X

$$\begin{cases} \Omega(\mathcal{X}^{t+1}) = \boldsymbol{y} & \text{(observed elem.)} \\ \bar{\Omega}(\mathcal{X}^{t+1}) = \bar{\Omega}\left(\frac{1}{K}\sum_{k=1}^{K} \mathrm{tensor}_k(\boldsymbol{Z}_k^t - \boldsymbol{A}_k^t)\right) & \text{(unobserved elem.)} \end{cases}$$

- Minimize the AL function wrt Z

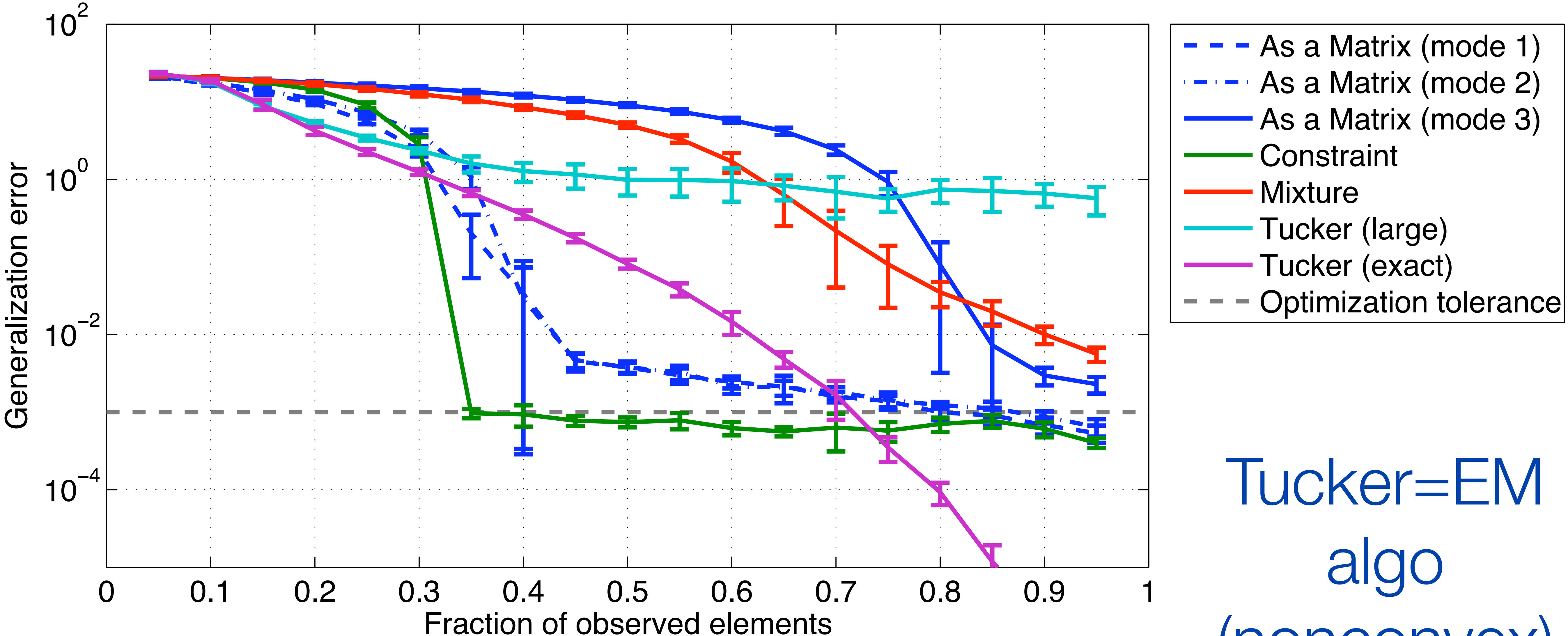
$$\boldsymbol{Z}_k^{t+1} = \mathrm{softth}_{\gamma_k/\eta}\left(\boldsymbol{X}_{(k)}^{t+1} + \boldsymbol{A}_k^t\right) \quad (k = 1, \ldots, K)$$

- Update multipliers

$$\boldsymbol{A}_k^{t+1} = \boldsymbol{A}_k^t + \left(\boldsymbol{X}_{(k)}^{t+1} - \boldsymbol{Z}_k^{t+1}\right) \quad (k = 1, \ldots, K)$$

Numerical experiment

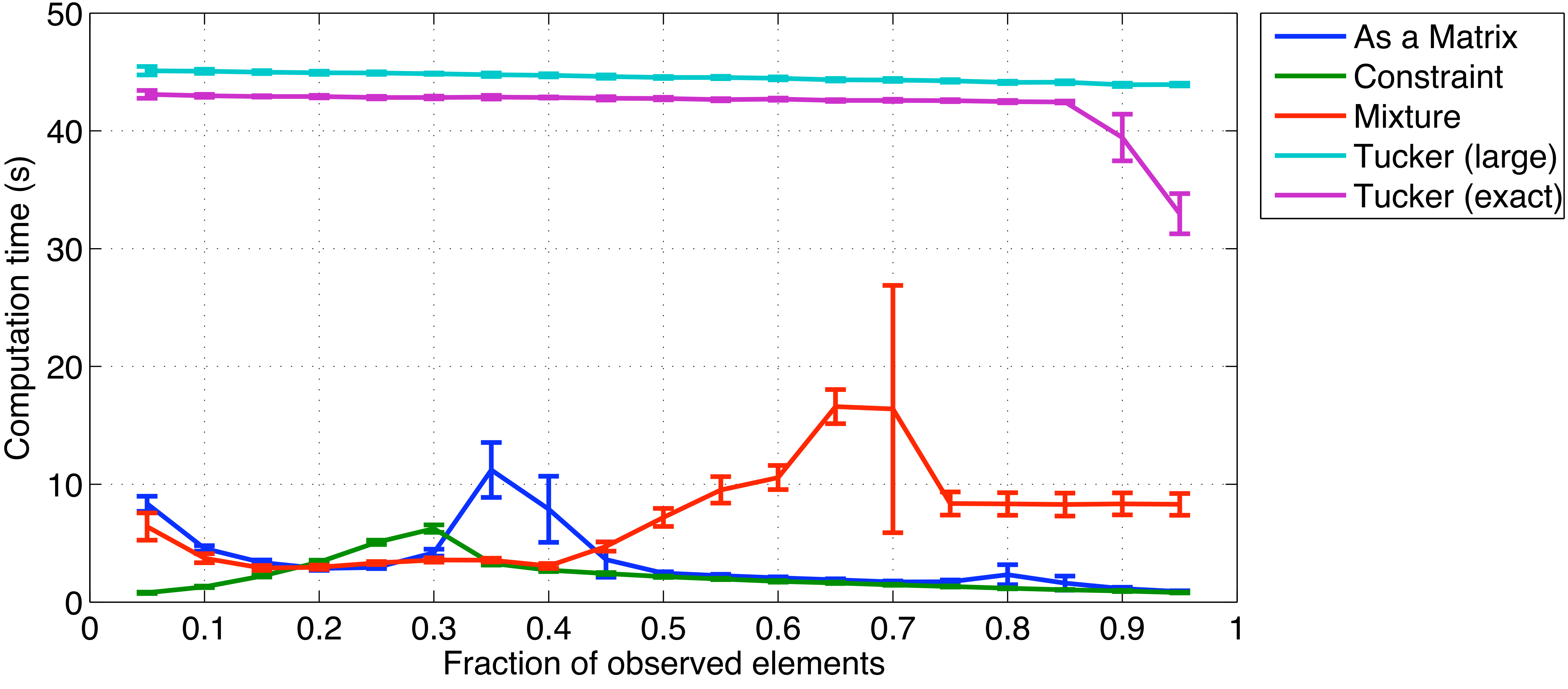
- True tensor: Size 50x50x20, rank 7x8x9. No noise (λ=0).
- Random train/test split.



Tucker=EM algo (nonconvex)

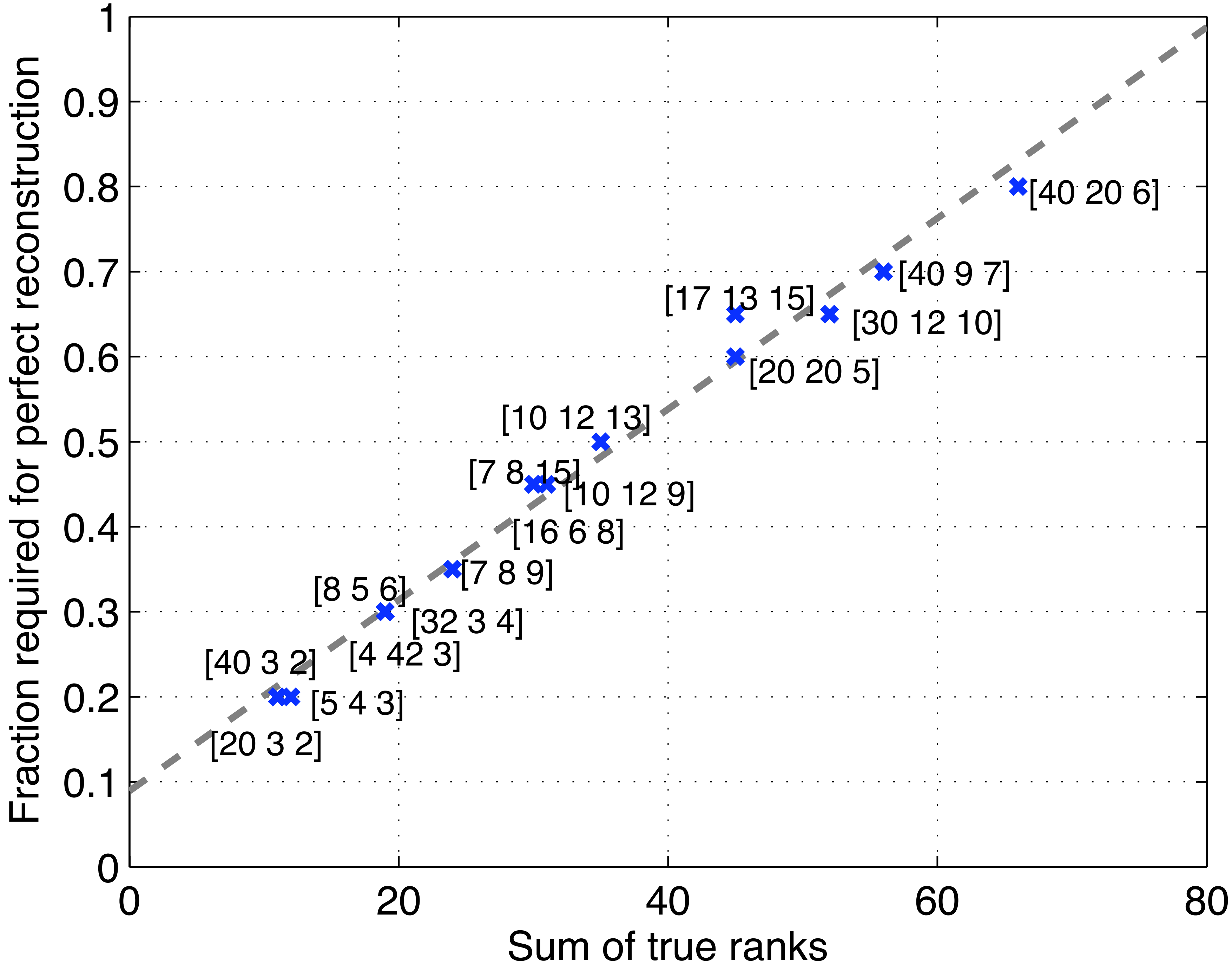
Computation time

- Convex formulation is also fast

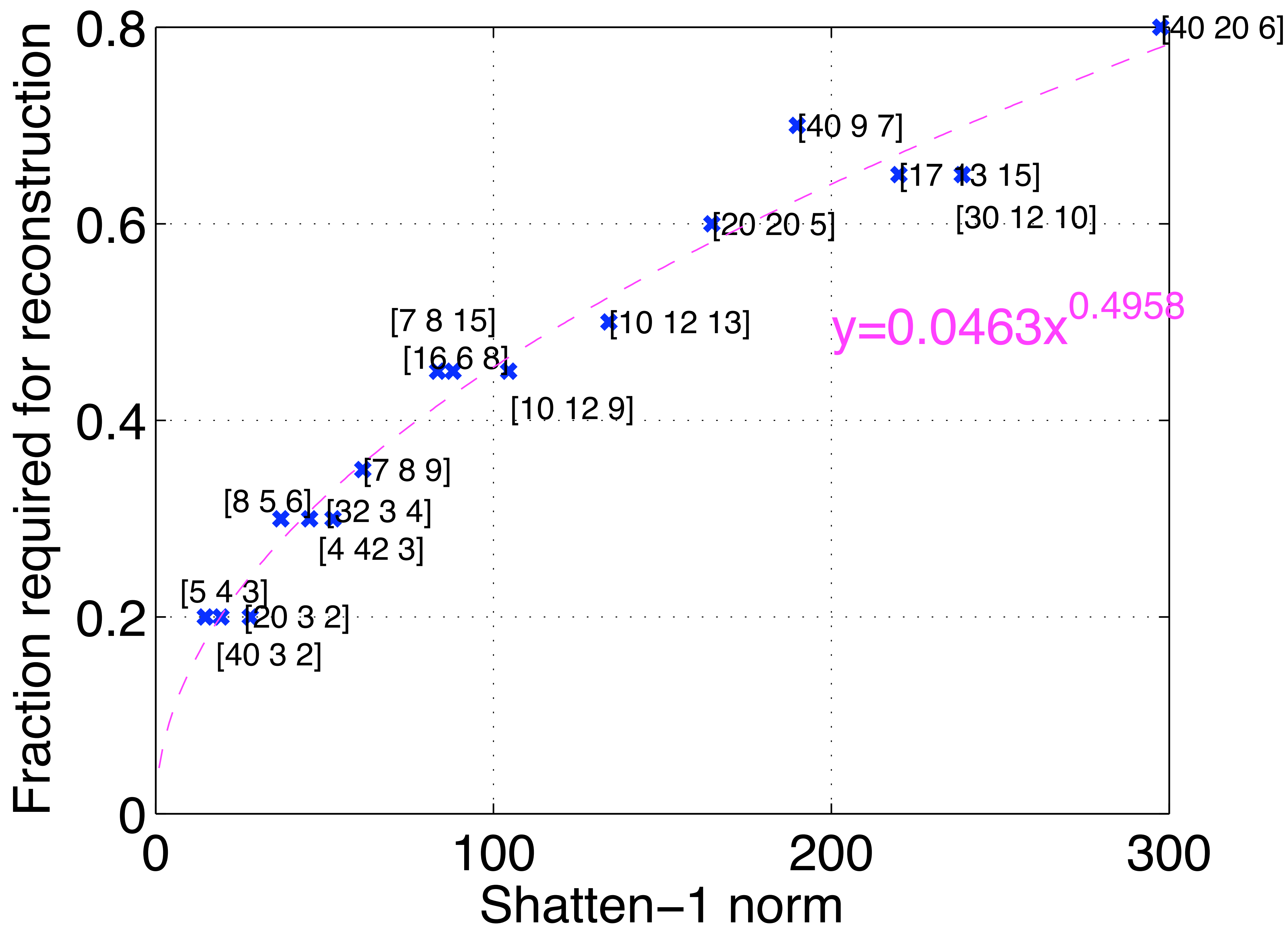


Phase transition behaviour

- Sum of true ranks= $\min(r_1, r_2 r_3) + \min(r_2, r_3 r_1) + \min(r_3, r_1 r_2)$

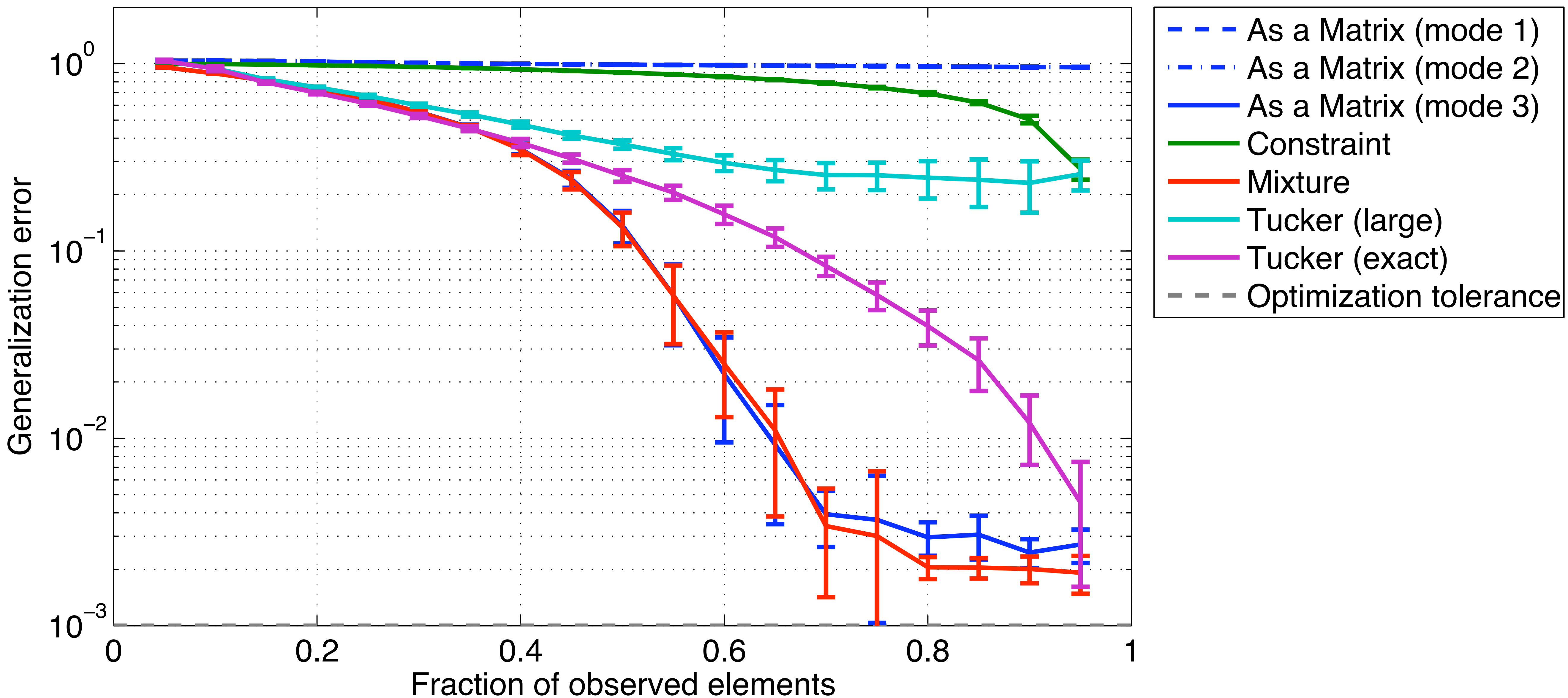


Phase transition (vs Shatten-1 norm)



“Mixture” is sometimes better

- True tensor: Size 50x50x20, rank 50x50x5. No noise (λ=0).

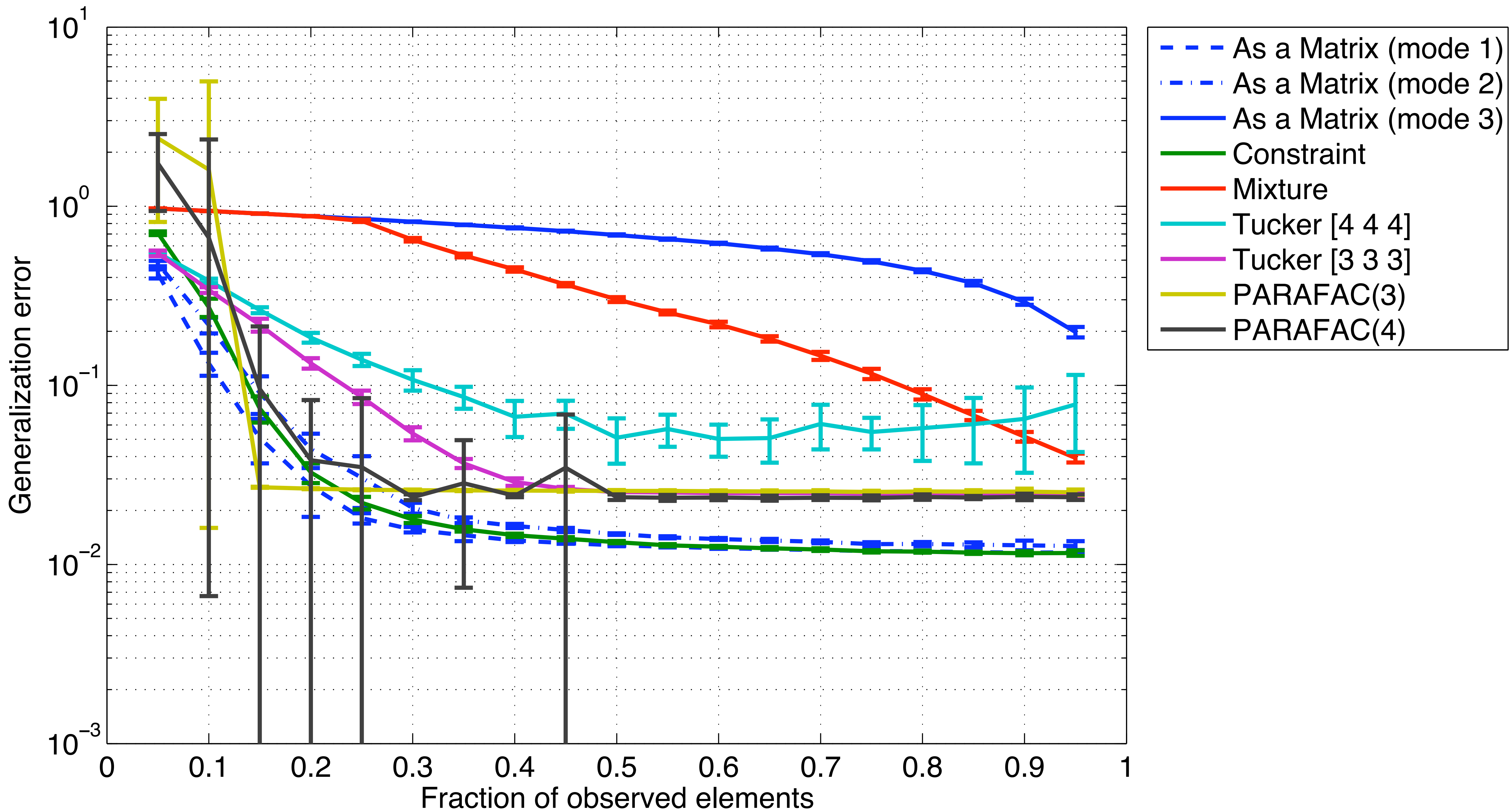


Amino acid fluorescence data [Bro & Andersson]

- Size 201x61x5.
- Five solutions with different amount of three amino acids (tyrosine, tryptophan, phenylalanine)
- Rank=3 PARAFAC is correct.
- Interested in
  - Generalization performance
  - Number of components
  - Interpretation

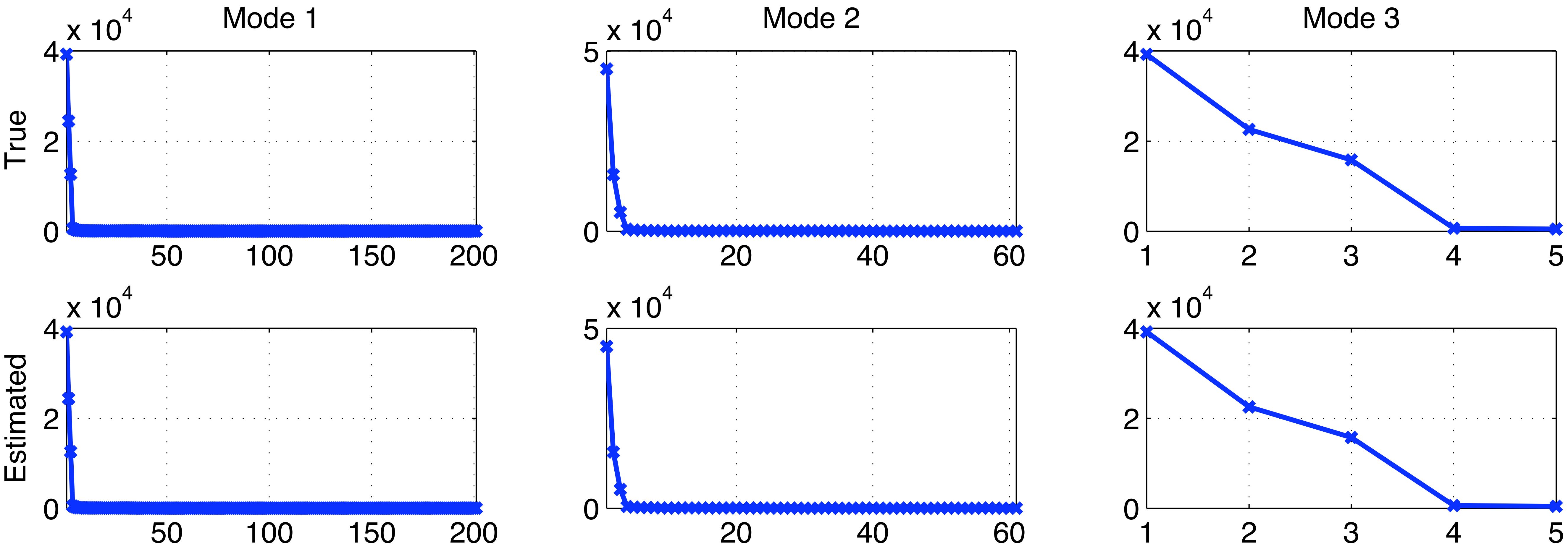
Amino acid: Generalization performance

- "Constraint" performs comparable to PARAFAC with the correct rank.



Amino acid: Singular-value spectra

Estimated spetra from half of the entries are almost identical to the truth.



Improving Interpretability

- Apply PARAFAC on the core (4x4x5) obtained by the proposed “constraint” approach.
- Separate imputation problem and interpretation problem.

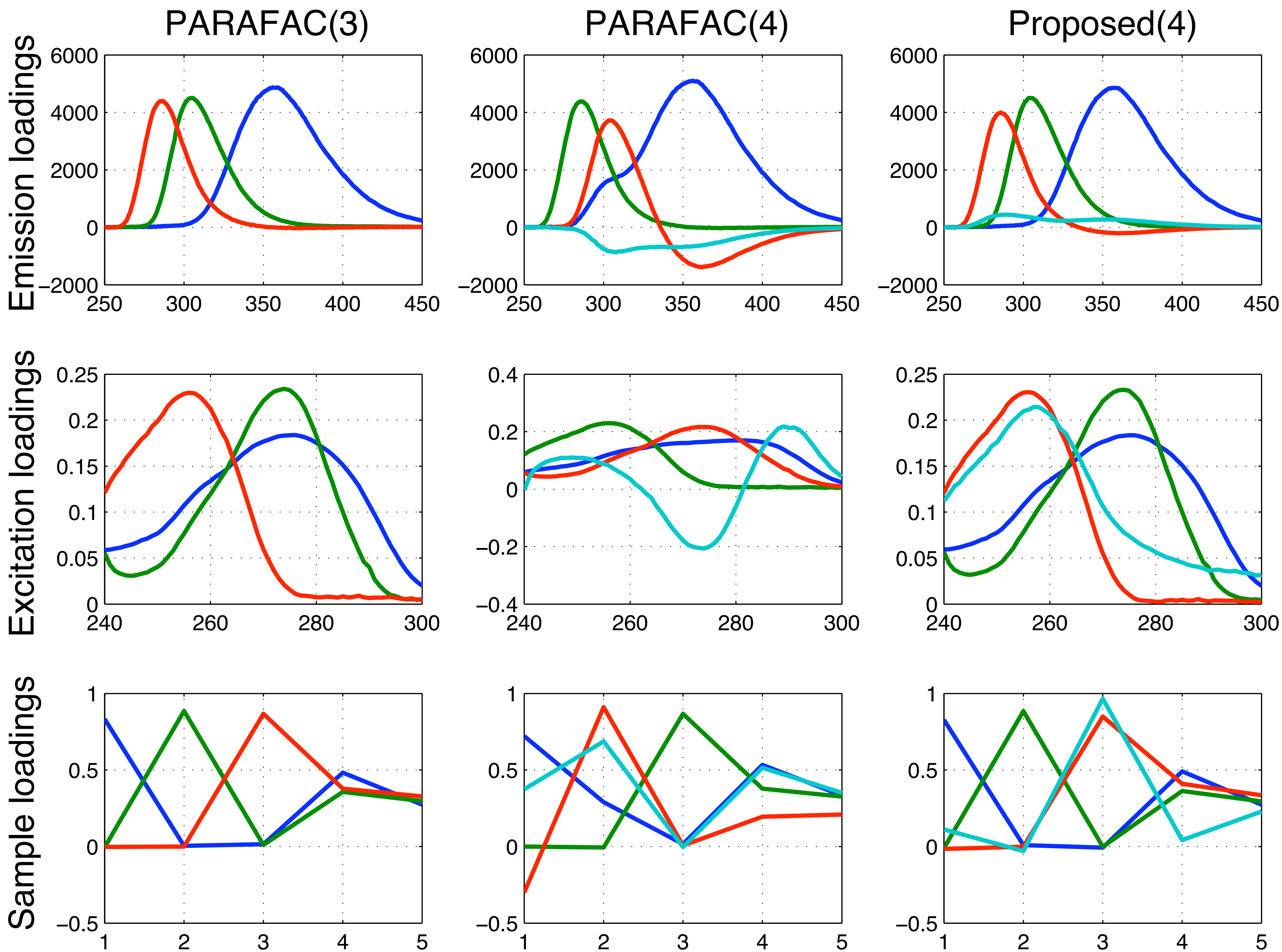
$$\mathcal{X} = \mathcal{C} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3$$

$$= (\boldsymbol{A}^{(1)} \odot \boldsymbol{A}^{(2)} \odot \boldsymbol{A}^{(3)}) \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3$$

$$= (\boldsymbol{U}_1 \boldsymbol{A}^{(1)}) \odot (\boldsymbol{U}_2 \boldsymbol{A}^{(2)}) \odot (\boldsymbol{U}_3 \boldsymbol{A}^{(3)})$$

PARAFAC

Obtained factors



Summary

- Low-rank tensor completion can be computed in a convex optimization problem using the trace norm regularization.
  - No need to specify the rank beforehand.
- Convex formulation is more accurate and faster than conventional EM-based Tucker decomposition.
- Curious "phase transition" found → compressive-sensing-type analysis is an on-going work.
- Combination of proposed+PARAFAC is useful.
- Code:
  - http://www.ibis.t.u-tokyo.ac.jp/RyotaTomioka/Softwares/Tensor

Acknowledgment

- This work was supported in part by MEXT KAKENHI 22700138, 80545583, JST PRESTO, and NTT Communication Science Laboratories.

ADMM convergence

- Step 1: ADMM is equivalent to Douglas-Rachfold Splitting in the dual

$$\boldsymbol{\alpha}^{t+1} = \mathsf{prox}_{g^*}\Big(\mathsf{prox}_{f^*(-\boldsymbol{A}^\top\cdot)}(\boldsymbol{\alpha}^t - \boldsymbol{z}^t) + \boldsymbol{z}^t\Big)$$
$$\boldsymbol{z}^{t+1} = \mathsf{prox}_{g}\Big(\mathsf{prox}_{f^*(-\boldsymbol{A}^\top\cdot)}(\boldsymbol{\alpha}^t - \boldsymbol{z}^t) + \boldsymbol{z}^t\Big)$$