MANIFOLD OPTIMIZATION IN DATA SCIENCE

Max Pfeffer

Networks Seminar

IISc Bangalore

March 3, 2026

MATRIX FACTORIZATIONS

MATRIX FACTORIZATIONS

- Dimensionality Reduction

MATRIX FACTORIZATIONS

- Dimensionality Reduction
- Feature Extraction

MATRIX FACTORIZATIONS

- Dimensionality Reduction
- Feature Extraction
  - ↳ PCA

MATRIX FACTORIZATIONS

- Dimensionality Reduction
- Feature Extraction
- Clustering

MATRIX FACTORIZATIONS

- Dimensionality Reduction
- Feature Extraction
- Clustering
  - ↳ Unsupervised Learning

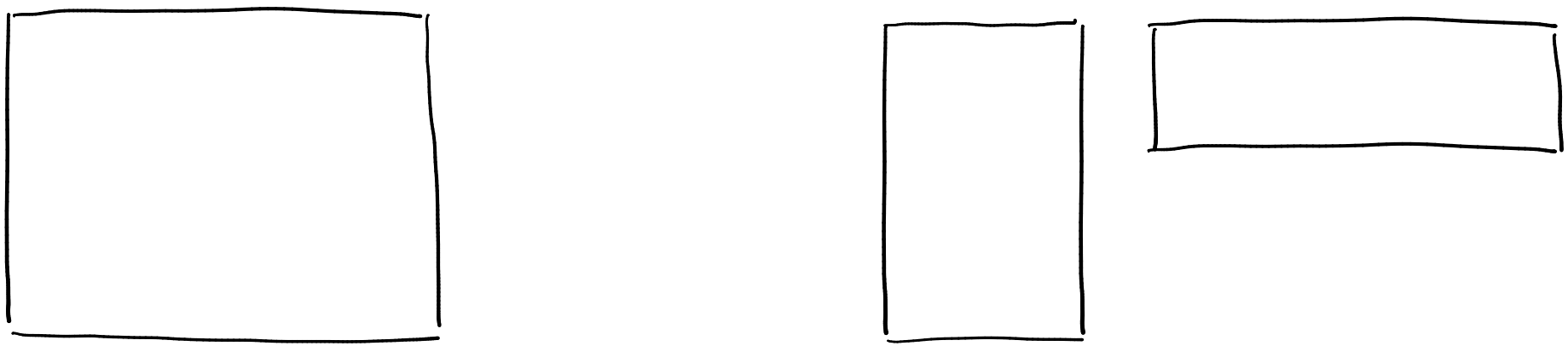
MATRIX FACTORIZATIONS

$$M \in \mathbb{R}^{m \times n}$$

MATRIX FACTORIZATIONS

$M \in \mathbb{R}^{m \times n}$ $X \in \mathbb{R}^{m \times r}$ $Y \in \mathbb{R}^{n \times r}$

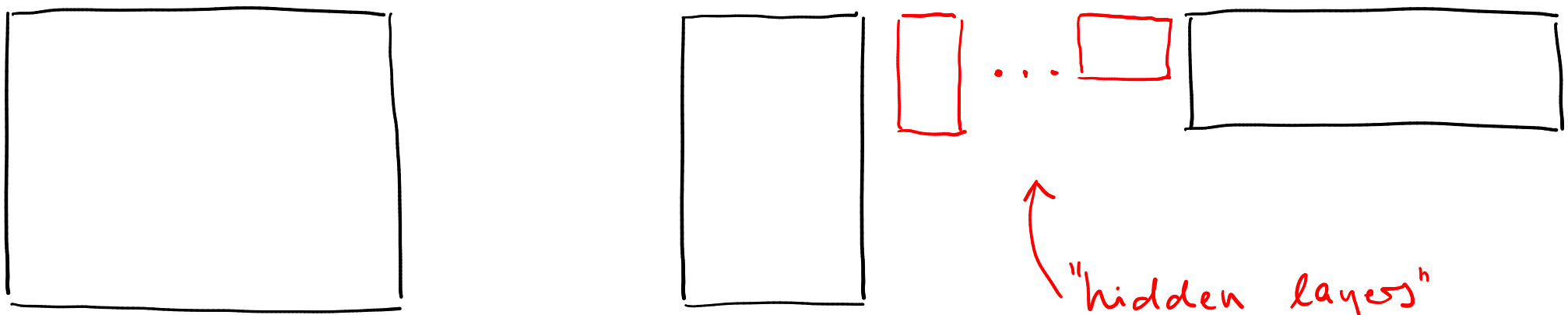
$M = X Y^T$



MATRIX FACTORIZATIONS

$M \in \mathbb{R}^{m \times n}$ $\quad X \in \mathbb{R}^{m \times r_1}$ $\quad A_i \in \mathbb{R}^{r_i \times r_{i+1}}$ $\quad Y \in \mathbb{R}^{n \times r_\ell}$

$M = X A_1 \cdots A_\ell Y^T$



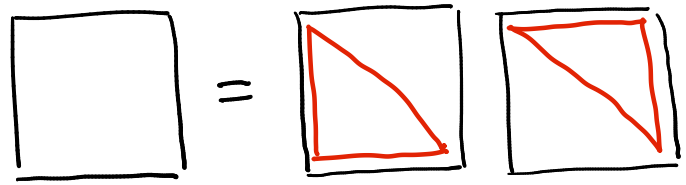
MATRIX FACTORIZATIONS

Examples:

MATRIX FACTORIZATIONS

Examples:

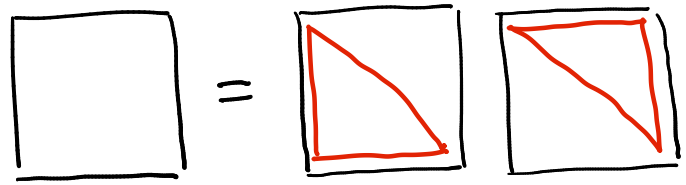
- Cholesky decomposition: $M = CC^T$, $M = M^T$



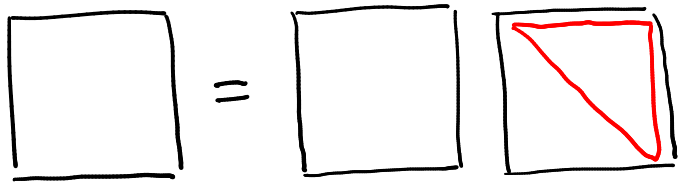
MATRIX FACTORIZATIONS

Examples:

- Cholesky decomposition: $M = CC^T$, $M = M^T$



- QR-decomposition: $M = QR$, $Q^TQ = I$

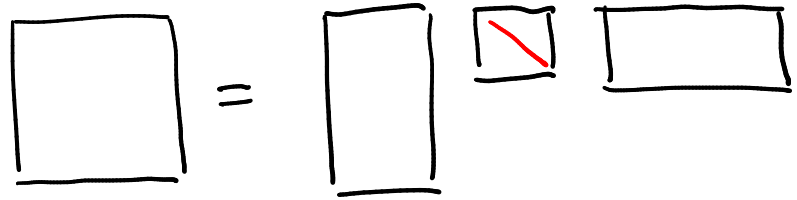


MATRIX FACTORIZATIONS

Examples:

- Singular-value-decomposition: (SVD)

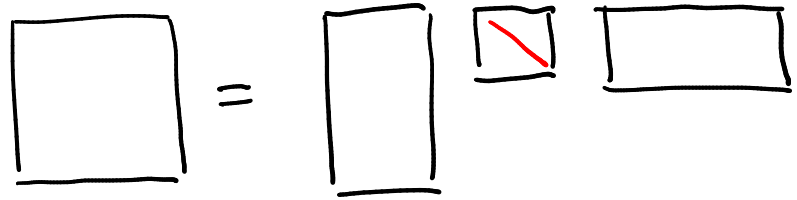
$M = U \Sigma V^T, \quad U^T U = V^T V = I$



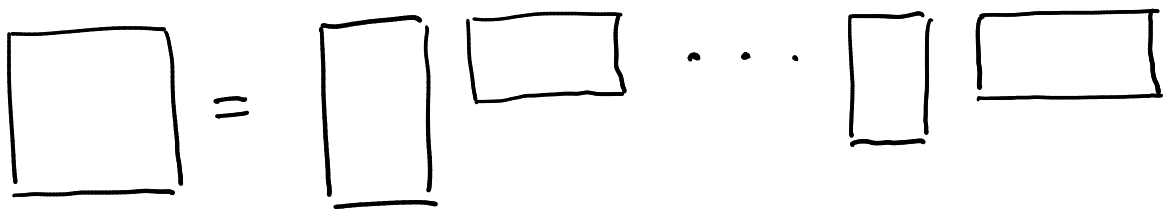
MATRIX FACTORIZATIONS

Examples:

- Singular-value-decomposition: $M = U \Sigma V^T$, $U^T U = V^T V = I$
  (SVD)



- Linear neural network: $M = A_1 \cdots A_\ell$



TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

TENSOR FACTORIZATIONS

$T \in \mathbb{R}^{n_1 \times \dots n_d} \longrightarrow$ order d

TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

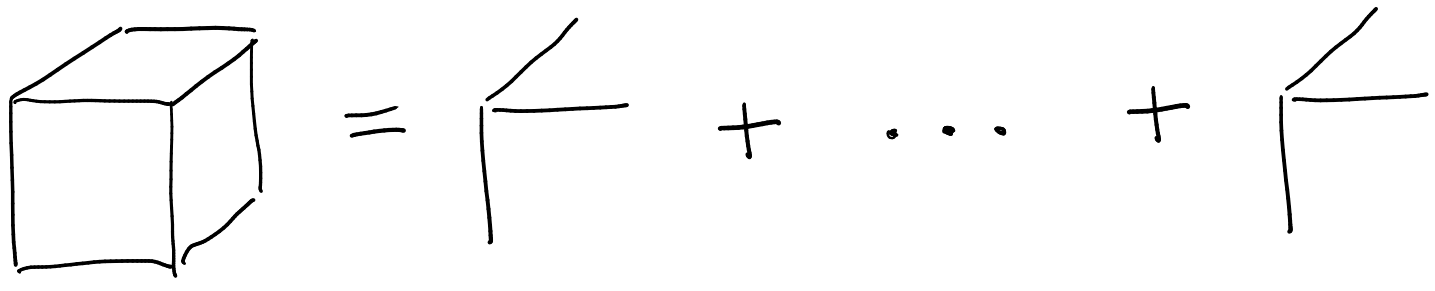
TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Canonical Polyadic (CP) decomposition:

$$T = \sum_{i=1}^{R} u_1^{(i)} \otimes \cdots \otimes u_d^{(i)} \quad , \quad u_j^{(i)} \in \mathbb{R}^{n_j}$$



TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Canonical Polyadic (CP) decomposition:

$$T = \sum_{i=1}^{R} u_1^{(i)} \otimes \cdots \otimes u_d^{(i)} \quad , \quad u_j^{(i)} \in \mathbb{R}^{n_j}$$

+ "usually" unique

TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Canonical Polyadic (CP) decomposition:

$$T = \sum_{i=1}^{R} u_1^{(i)} \otimes \cdots \otimes u_d^{(i)} \quad , \quad u_j^{(i)} \in \mathbb{R}^{n_j}$$

  + "usually" unique
  + easy to interpret

TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Canonical Polyadic (CP) decomposition:

$$T = \sum_{i=1}^{R} u_1^{(i)} \otimes \cdots \otimes u_d^{(i)} \quad , \quad u_j^{(i)} \in \mathbb{R}^{n_j}$$

+ "usually" unique
+ easy to interpret
− difficult to compute

TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Canonical Polyadic (CP) decomposition:

$$T = \sum_{i=1}^{R} u_1^{(i)} \otimes \cdots \otimes u_d^{(i)} \quad , \quad u_j^{(i)} \in \mathbb{R}^{n_j}$$

  + "usually" unique
  + easy to interpret
  − difficult to compute
  − not closed

TENSOR FACTORIZATIONS

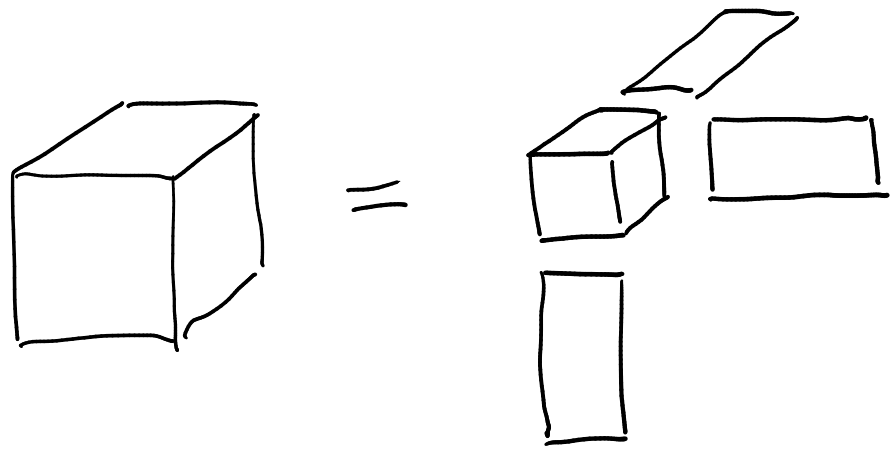
$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Tuckers decomposition:

$$T = C \circ_1 U_1 \circ_2 U_2 \cdots \circ_d U_d \quad , \quad C \in \mathbb{R}^{r_1 \times \dots \times r_d}$$

$$U_i \in \mathbb{R}^{n_i \times r_i}$$



TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Tucker decomposition:

$$T = C \circ_1 U_1 \circ_2 U_2 \cdots \circ_d U_d \,, \quad C \in \mathbb{R}^{r_1 \times \dots \times r_d}, \quad U_i \in \mathbb{R}^{n_i \times r_i}$$

+ easy to compute

TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Tucker decomposition:

$$T = C \circ_1 U_1 \circ_2 U_2 \cdots \circ_d U_d \,, \quad C \in \mathbb{R}^{r_1 \times \dots \times r_d}, \quad U_i \in \mathbb{R}^{n_i \times r_i}$$

  + easy to compute
  + can be made unique

TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Tuckers decomposition:

$$T = C \circ_1 U_1 \circ_2 U_2 \cdots \circ_d U_d \,, \quad C \in \mathbb{R}^{r_1 \times \dots \times r_d}$$
$$U_i \in \mathbb{R}^{n_i \times r_i}$$

  + easy to compute
  + can be made unique
  - still exponential in d

TENSOR FACTORIZATIONS

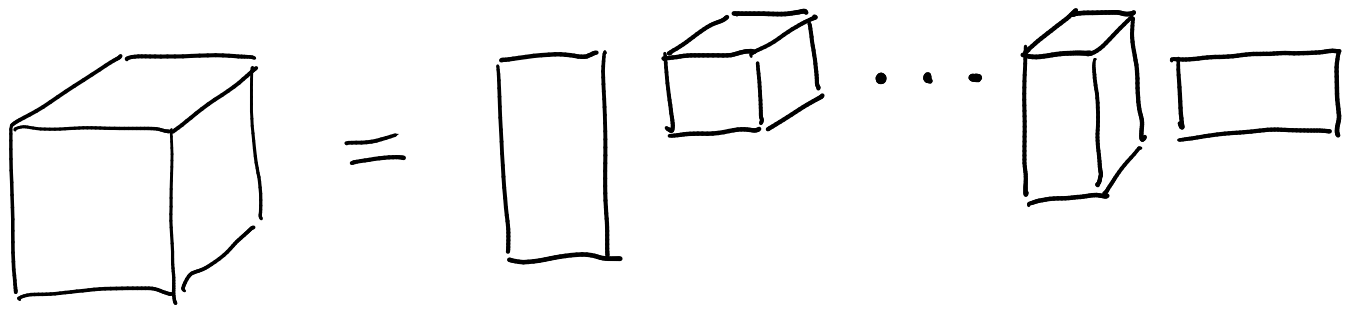
$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Tensor Train (TT) decomposition:

$$T = U_1 \bullet U_2 \bullet \cdots \bullet U_d \quad , \quad U_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$$

$$r_0 = r_d = 1$$



TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Tensor Train (TT) decomposition:

$$T = U_1 \bullet U_2 \bullet \cdots \bullet U_d \quad , \quad U_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$$

$$r_0 = r_d = 1$$

+ better storage than Tucker

TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Tensor Train (TT) decomposition:

$$T = U_1 \bullet U_2 \bullet \cdots \bullet U_d \quad , \quad U_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$$

$$r_0 = r_d = 1$$

+ better storage than Tucker
+ easy to compute

TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Tensor Train (TT) decomposition:

$$T = U_1 \bullet U_2 \bullet \cdots \bullet U_d \quad , \quad U_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$$

$$r_0 = r_d = 1$$

+ better storage than Tucker
+ easy to compute
+ can be made unique

TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- Tensor Train (TT) decomposition:

$$T = U_1 \bullet U_2 \bullet \cdots \bullet U_d \quad , \quad U_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$$

$$r_0 = r_d = 1$$

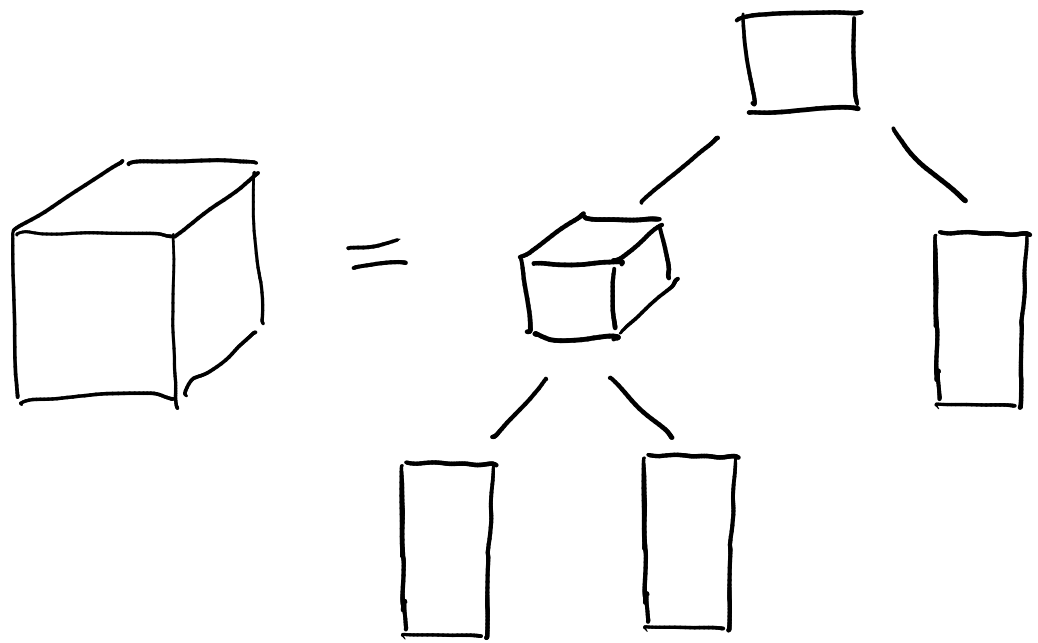
+ better storage than Tucker
+ easy to compute
+ can be made unique
- interpretation not straightforward

TENSOR FACTORIZATIONS

$$T \in \mathbb{R}^{n_1 \times \dots n_d}$$

Formats:

- General tree tensor networks:



JOINT FACTORIZATIONS

Take collection of matrices $M_1, \ldots, M_K$

and/or tensors $\mathcal{T}_1, \ldots, \mathcal{T}_L$.

JOINT FACTORIZATIONS

Take collection of matrices $M_1, \ldots, M_K$

and/or tensors $T_1, \ldots, T_L$.

Decompose them with some shared factors.

JOINT FACTORIZATIONS

Take collection of matrices $M_1, \dots, M_K$

and/or tensors $T_1, \dots, T_L$.

Decompose them with some shared factors.

Example: $X = [x_1 \cdots x_R]$

$$M = X Y^T \quad , \quad T = \sum_{i=1}^{R} x_i \otimes u_1^{(i)} \otimes \cdots \otimes u_{d-1}^{(i)}$$

CONSTRAINTS

It is often beneficial to impose the factors with constraints.

CONSTRAINTS

Some examples:

CONSTRAINTS

Some examples:

- Orthogonality: $A^T A = I$

CONSTRAINTS

Some examples:

- Orthogonality: $A^T A = I$
  - can enforce uniqueness

CONSTRAINTS

Some examples:

- Orthogonality: $A^T A = I$
  - can enforce uniqueness
  - is a form of independence

CONSTRAINTS

Some examples:

- Orthogonality: $A^T A = I$
- (Column) - normalization: $\operatorname{diag}(A^T A) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$

CONSTRAINTS

Some examples:

- Orthogonality: $A^TA = I$
- (Column) - normalization: $\text{diag}(A^TA) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$
  - when orthogonality is too much to ask

CONSTRAINTS

Some examples:

- Orthogonality: $A^TA = I$
- (Column) - normalization: $\operatorname{diag}(A^TA) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$
- Nonnegativity: $A \in \mathbb{R}_+^{m \times r}$

CONSTRAINTS

Some examples:

- Orthogonality: $A^TA = I$
- (Column) - normalization: $\text{diag}(A^TA) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$
- Nonnegativity: $A \in \mathbb{R}_+^{m \times r}$
  - when the data is also nonnegative

CONSTRAINTS

Some examples:

- Orthogonality: $A^TA = I$
- (Column) - normalization: $\operatorname{diag}(A^TA) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$
- Nonnegativity: $A \in \mathbb{R}_+^{m \times r}$
- Sparsity: $\|A\|_0 < s$

CONSTRAINTS

Some examples:

- Orthogonality: $A^TA = I$
- (Column) - normalization: $\operatorname{diag}(A^TA) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$
- Nonnegativity: $A \in \mathbb{R}_+^{m \times r}$
- Sparsity: $\|A\|_0 < s$
  - for feature extraction

CONSTRAINTS

Some examples:

- Orthogonality: $A^TA = I$
- (Column) - normalization: $\operatorname{diag}(A^TA) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$
- Nonnegativity: $A \in \mathbb{R}_+^{m \times r}$
- Sparsity: $\|A\|_0 < s$
- Boolean: $A \in \{0,1\}^{m \times r}$

CONSTRAINTS

Some examples:

- Orthogonality: $A^TA = I$
- (Column) - normalization: $\operatorname{diag}(A^TA) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$
- Nonnegativity: $A \in \mathbb{R}_+^{m \times r}$
- Sparsity: $\|A\|_0 < s$
- Boolean: $A \in \{0,1\}^{m \times r}$
  - possibly also in boolean (max-times) algebra

CONSTRAINTS

Some examples:

- Orthogonality: $A^TA = I$
- (Column) - normalization: $\operatorname{diag}(A^TA) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$
- Nonnegativity: $A \in \mathbb{R}_+^{m \times r}$
- Sparsity: $\|A\|_0 < s$
- Boolean: $A \in \{0,1\}^{m \times r}$
  - possibly also in boolean (max-times) algebra
  - for truth values

CONSTRAINTS

Some examples:

- Orthogonality: $A^TA = I$
- (Column) - normalization: $\operatorname{diag}(A^TA) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$
- Nonnegativity: $A \in \mathbb{R}_+^{m \times r}$
- Sparsity: $\|A\|_0 < s$
- Boolean: $A \in \{0,1\}^{m \times r}$
- Statistical independence (of the columns)

CONSTRAINTS

Some examples:

- Orthogonality: $A^TA = I$
- (Column) - normalization: $\operatorname{diag}(A^TA) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$
- Nonnegativity: $A \in \mathbb{R}_+^{m \times r}$
- Sparsity: $\|A\|_0 < s$
- Boolean: $A \in \{0,1\}^{m \times r}$
- Statistical independence (of the columns)
- many more

THEORETICAL PROBLEMS

THEORETICAL PROBLEMS

- Computability

THEORETICAL PROBLEMS

- Computability
- Uniqueness

THEORETICAL PROBLEMS

- Computability
- Uniqueness
- Closedness

THEORETICAL PROBLEMS

- Computability
- Uniqueness
- Closedness
- rank $M \leq$ constrained rank M

ALGORITHMS

The question remains how these factorizations are computed.

ALGORITHMS

The question remains how these factorizations are computed.

Some known examples:

ALGORITHMS

The question remains how these factorizations are computed.

Some known examples:

- LU-decomposition can be computed exactly using the Gauss-algorithm.

ALGORITHMS

The question remains how these factorizations are computed.

Some known examples:

- LU-decomposition can be computed exactly using the Gauss-algorithm.
- QR can be computed using Householder transformations.

ALGORITHMS

The question remains how these factorizations are computed.

Some known examples:

- LU-decomposition can be computed exactly using the Gauss-algorithm.
- QR can be computed using Householder transformations.
- SVD can be computed to arbitrary accuracy in polynomial time.

ALGORITHMS

Otherwise, we need optimization algorithms.

ALGORITHMS

Otherwise, we need optimization algorithms.

General form:

ALGORITHMS

Otherwise, we need optimization algorithms.

General form:

$$\min_{X,Y} \|M - XY^T\|_F^2$$

ALGORITHMS

Otherwise, we need optimization algorithms.

General form:

$$\min_{X,Y} \|M - XY^T\|_F^2$$

for each matrix / tensor

ALGORITHMS

Otherwise, we need optimization algorithms.

General form:

$$\min_{X,Y} \|M - XY^T\|_F^2$$

format of the decomposition

ALGORITHMS

Otherwise, we need optimization algorithms.

General form:

$$\min_{X,Y \in S} \|M - XY^T\|_F^2$$

↖ subset constraints

ALGORITHMS

Otherwise, we need optimization algorithms.

General form:

$$\min_{X,Y} \|M - XY^T\|_F^2 + \lambda g(X,Y)$$

penalty for constraints

ALGORITHMS

Otherwise, we need optimization algorithms.

General form:

$$\min_{X, Y_1, Y_2} \| M_1 - X Y_1^T \|_F^2 + \| M_2 - X Y_2^T \|$$

joint factorization

ALGORITHMS

Some techniques:

ALGORITHMS

Some techniques:

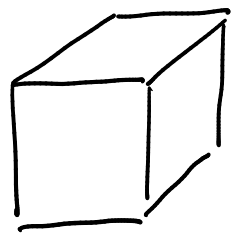
- Some factorizations can be computed by successive QR / SVD:

ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD:

Tucker-format

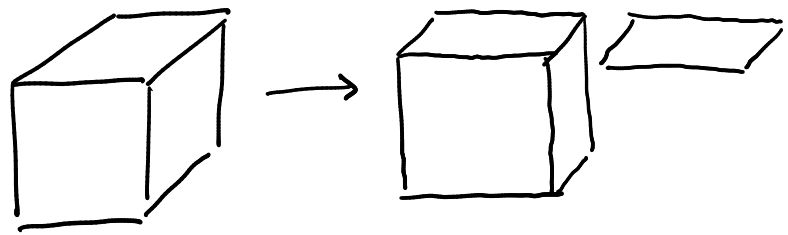


ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD:

Tucker-format

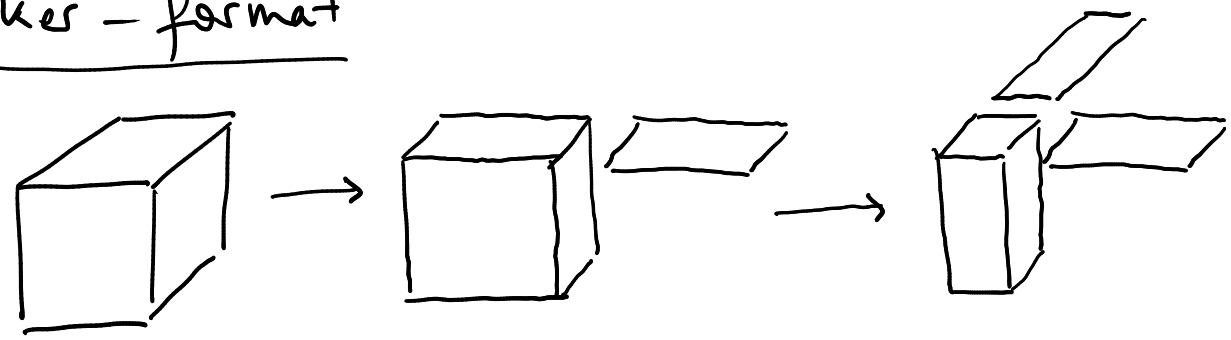


ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD:

Tucker-format

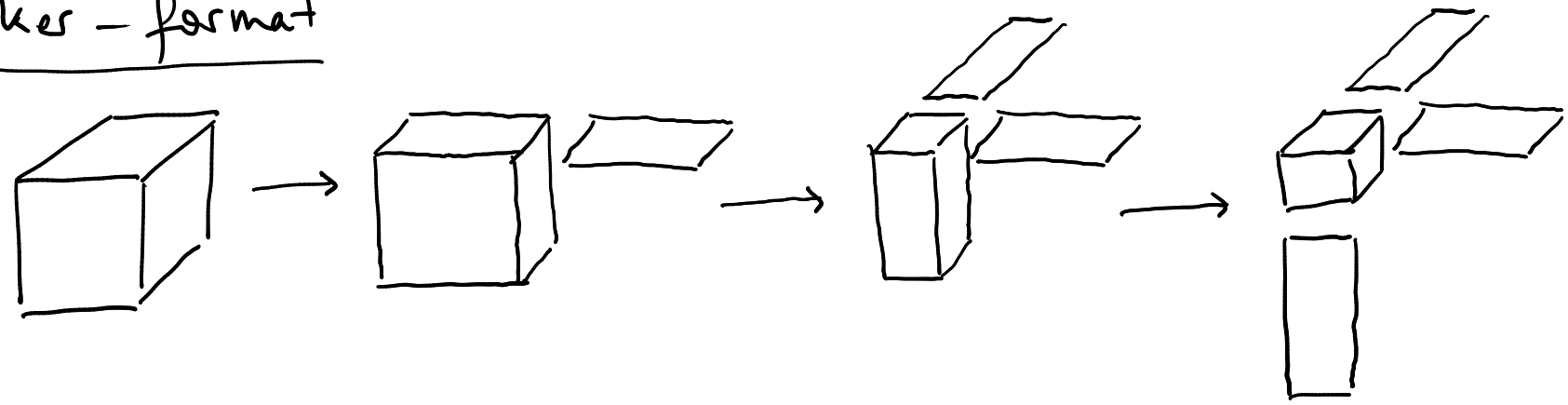


ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD:

Tucker-format



ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD.
- Alternating least squares (ALS):

ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD.
- Alternating least squares (ALS):

$$\min_{X,Y} \| M - XY^T \|_F^2$$

ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD.
- Alternating least squares (ALS):

$$\min_{X} \| M - X Y^T \|_F^2$$

Fix Y, minimize for X

ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD.
- Alternating least squares (ALS):

$$\min_{Y} \| M - XY^T \|_F^2$$

Fix X, minimize for Y

ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD.
- Alternating least squares (ALS):

$$\min_{X} \| M - XY^T \|_F^2$$

↻ repeat

ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD.
- Alternating least squares (ALS):
  - cost will always decrease

ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD.
- Alternating least squares (ALS):
  - cost will always decrease
  - convergence to critical point not always guaranteed

ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD.
- Alternating least squares (ALS):
  - cost will always decrease
  - convergence to critical point not always guaranteed
  - often fast but can be slow

ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD.
- Alternating least squares (ALS):
- All-at-once-optimization

ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD.
- Alternating least squares (ALS).
- All-at-once optimization

ALGORITHMS

Some techniques:

- Some factorizations can be computed by successive QR / SVD.
- Alternating least squares (ALS).
- All-at-once-optimization
  → Riemannian optimization

RIEMANNIAN OPTIMIZATION

If the factors are constrained to a

Riemannian manifold

RIEMANNIAN OPTIMIZATION

If the factors are constrained to a

Riemannian manifold

$$\min f(X)$$

RIEMANNIAN OPTIMIZATION

If the factors are constrained to a

Riemannian manifold

$$\min f(X) \quad \text{for} \quad X \in S$$

RIEMANNIAN OPTIMIZATION

If the factors are constrained to a

Riemannian manifold

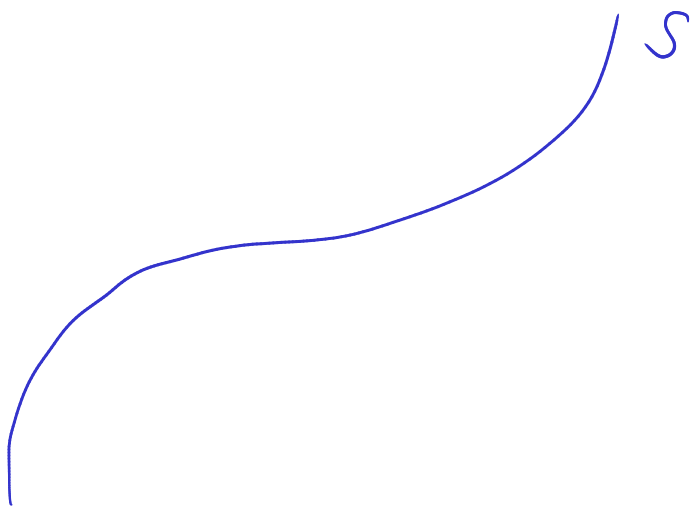
$$\min f(X) \quad \text{for} \quad X \in \mathcal{S} \text{ , e.g. } f(X) = \frac{1}{2}\|M - XY^T\|_F^2$$

RIEMANNIAN OPTIMIZATION

If the factors are constrained to a

Riemannian manifold

$$\min f(X) \quad \text{for} \quad X \in \mathcal{S}, \quad \text{e.g.} \quad f(X) = \frac{1}{2}\|M - XY^T\|_F^2$$

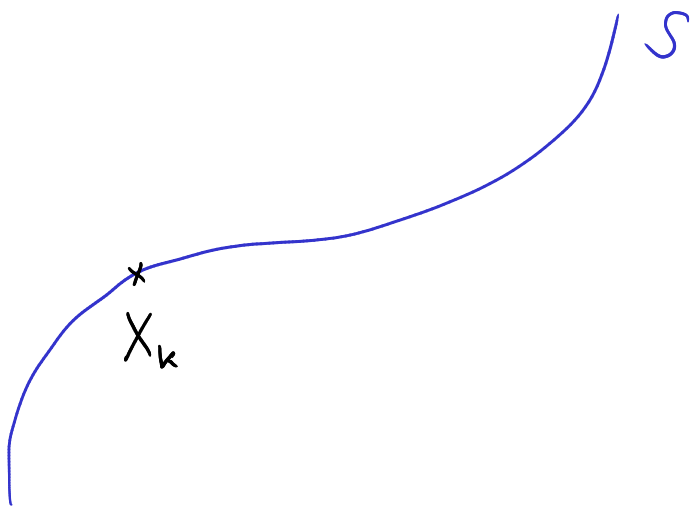


RIEMANNIAN OPTIMIZATION

If the factors are constrained to a

Riemannian manifold

$$\min f(X) \quad \text{for} \quad X \in \mathcal{S}, \quad \text{e.g.} \quad f(X) = \frac{1}{2}\|M - XY^T\|_F^2$$



RIEMANNIAN OPTIMIZATION

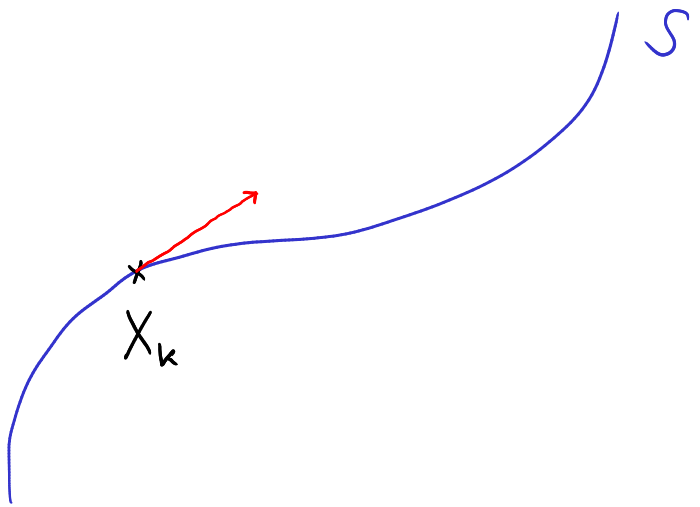
If the factors are constrained to a

Riemannian manifold

$$\min f(X) \quad \text{for} \quad X \in \mathcal{S}, \quad \text{e.g.} \quad f(X) = \frac{1}{2}\|M - XY^T\|_F^2$$

1. Compute Riemannian gradient

$$\operatorname{grad} f(X_k) \in T_{X_k}\mathcal{S}$$



RIEMANNIAN OPTIMIZATION

If the factors are constrained to a

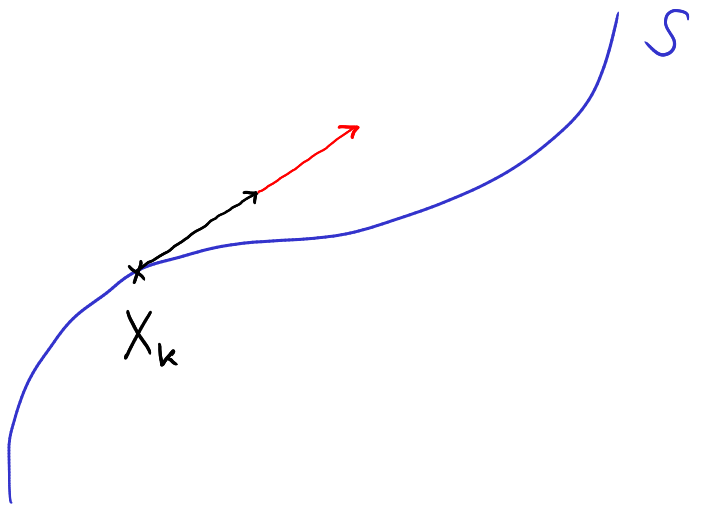
Riemannian manifold

$$\min f(X) \quad \text{for} \quad X \in \mathcal{S}, \quad \text{e.g.} \quad f(X) = \frac{1}{2}\|M - XY^T\|_F^2$$

1. Compute Riemannian gradient

   $\operatorname{grad} f(X_k) \in T_{X_k}\mathcal{S}$

2. Find appropriate stepsize



RIEMANNIAN OPTIMIZATION

If the factors are constrained to a

Riemannian manifold

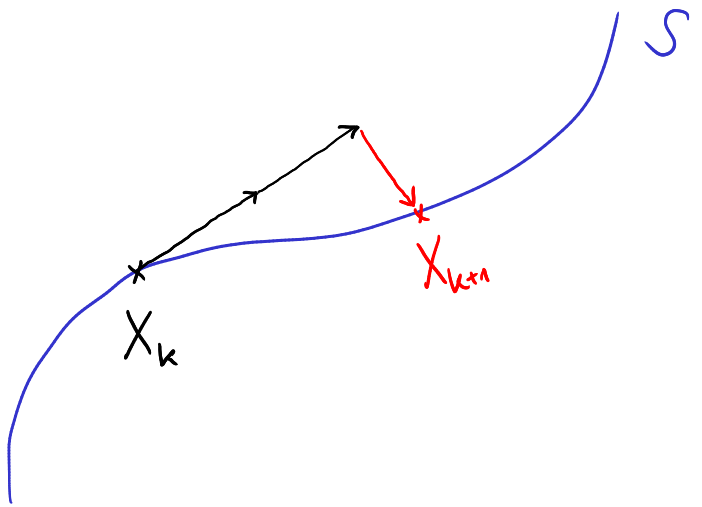
$$\min f(X) \quad \text{for} \quad X \in \mathcal{S}, \quad \text{e.g.} \quad f(X) = \frac{1}{2}\|M - XY^T\|_F^2$$

1. Compute Riemannian gradient

   $\operatorname{grad} f(X_k) \in T_{X_k}\mathcal{S}$

2. Find appropriate stepsize

3. Move on the manifold



RIEMANNIAN OPTIMIZATION

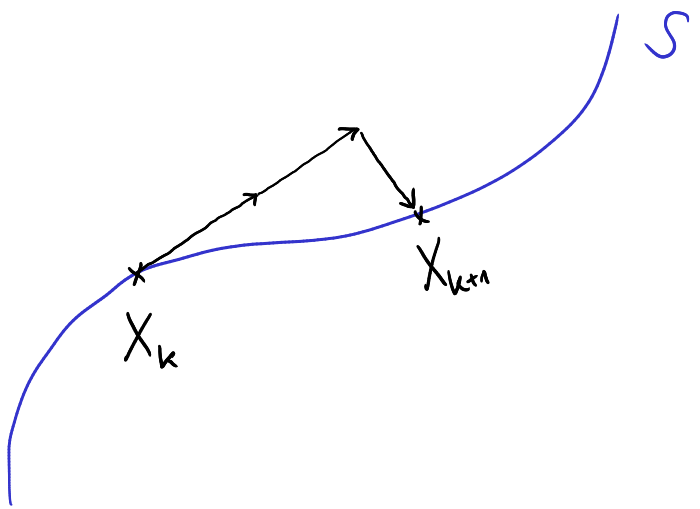
If the factors are constrained to a

Riemannian manifold

$$\min f(X) \quad \text{for} \quad X \in \mathcal{S}, \quad \text{e.g.} \quad f(X) = \frac{1}{2}\|M - XY^T\|_F^2$$

1. Compute Riemannian gradient
   $\operatorname{grad} f(X_k) \in T_{X_k}\mathcal{S}$
2. Find appropriate stepsize
3. Move on the manifold

} many improvements possible!



RIEMANNIAN OPTIMIZATION

Examples for smooth manifolds:

RIEMANNIAN OPTIMIZATION

Examples for smooth manifolds:

- Stiefel manifold: $St(n,r) = \{ X \in \mathbb{R}^{n \times r} : X^T X = I \}$

RIEMANNIAN OPTIMIZATION

Examples for smooth manifolds:

- Stiefel manifold: $St(n,r) = \{ X \in \mathbb{R}^{n \times r} : X^T X = I \}$
- Oblique manifold: $Ob(m,n) = \{ X \in \mathbb{R}^{m \times n} : \| x_i \|_2 = 1 , i = 1, \dots, n \}$

RIEMANNIAN OPTIMIZATION

Examples for smooth manifolds:

- Stiefel manifold: $St(n,r) = \{ X \in \mathbb{R}^{n \times r} : X^T X = I \}$
- Oblique manifold: $Ob(m,n) = \{ X \in \mathbb{R}^{m \times n} : \| x_i \|_2 = 1 , i = 1, \dots, n \}$
- Fixed-rank manifold: $\mathbb{R}_r^{m \times n} = \{ X \in \mathbb{R}^{m \times n} : \text{rank}(X) = r \}$

RIEMANNIAN OPTIMIZATION

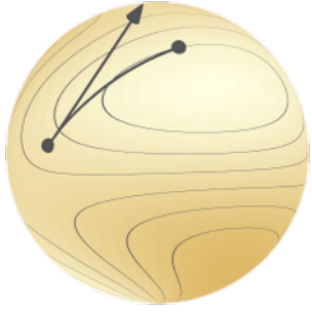
Examples for smooth manifolds:

- Stiefel manifold: $St(n,r) = \{X \in \mathbb{R}^{n \times r} : X^T X = I\}$
- Oblique manifold: $Ob(m,n) = \{X \in \mathbb{R}^{m \times n} : \|x_i\|_2 = 1, i = 1, \dots, n\}$
- Fixed-rank manifold: $\mathbb{R}_r^{m \times n} = \{X \in \mathbb{R}^{m \times n} : \text{rank}(X) = r\}$
- Products thereof

RIEMANNIAN OPTIMIZATION

Examples for smooth manifolds:

- Stiefel manifold: $St(n,r) = \{X \in \mathbb{R}^{n \times r} : X^T X = I\}$
- Oblique manifold: $Ob(m,n) = \{X \in \mathbb{R}^{m \times n} : \|x_i\|_2 = 1, i = 1, \dots, n\}$
- Fixed-rank manifold: $\mathbb{R}_r^{m \times n} = \{X \in \mathbb{R}^{m \times n} : \text{rank}(X) = r\}$
- Products thereof



Manopt

PyManopt

Manopt.jl

RIEMANNIAN OPTIMIZATION

Smooth constraints:

RIEMANNIAN OPTIMIZATION

Smooth constraints:

- Nonnegativity: $X = B \circ B$

RIEMANNIAN OPTIMIZATION

Smooth constraints:

- Nonnegativity: $X = B \circ B$

  + B on Oblique manifold $\Rightarrow$ L_1-normalized columns

RIEMANNIAN OPTIMIZATION

Smooth constraints:

- Nonnegativity: $X = B \circ B$

  + B on Oblique manifold $\Rightarrow$ L_1-normalized columns

Nonsmooth (convex) constraints:

- Sparsity: $\min f(X) + \lambda \|X\|_1$

RIEMANNIAN OPTIMIZATION

Smooth constraints:

- Nonnegativity: $X = B \circ B$

  + B on Oblique manifold $\Rightarrow$ L_1-normalized columns

Nonsmooth (convex) constraints:

- Sparsity: $\min f(X) + \lambda \|X\|_1$

  $\longrightarrow$ nonsmooth optimization on Riemannian manifolds

APPLICATIONS

CANCER CLASSIFICATION

CANCER CLASSIFICATION

We have m patients and 3 different data sets:

CANCER CLASSIFICATION

We have m patients and 3 different data sets:

$G \in \mathbb{R}_{+}^{m \times n_1}$ - Gene expression data

CANCER CLASSIFICATION

We have m patients and 3 different data sets:

$G \in \mathbb{R}_+^{m \times n_1}$ – Gene expression data

$M \in \mathbb{R}_+^{m \times n_2}$ – Methylation data

CANCER CLASSIFICATION

We have m patients and 3 different data sets:

$G \in \mathbb{R}_+^{m \times n_1}$ – Gene expression data

$M \in \mathbb{R}_+^{m \times n_2}$ – Methylation data

$C \in \mathbb{R}^{m \times n_3}$ – Copy number data

CANCER CLASSIFICATION

Minimization problem:

$$\min \frac{1}{2}\|G - U V_1^T\|_F^2 + \frac{1}{2}\|M - U V_2^T\|_F^2 + \frac{1}{2}\|C - U V_3^T\|_F^2$$

CANCER CLASSIFICATION

Minimization problem:

$$\min \ \frac{1}{2}\|G - U V_1^T\|_F^2 + \frac{1}{2}\|M - U V_2^T\|_F^2 + \frac{1}{2}\|C - U V_3^T\|_F^2$$

s.t. U, V_1, V_2 nonnegative

V_1, V_2, V_3 column-normalized

CANCER CLASSIFICATION

Minimization problem:

$$\min \ \frac{1}{2}\|G - U V_1^T\|_F^2 + \frac{1}{2}\|M - U V_2^T\|_F^2 + \frac{1}{2}\|C - U V_3^T\|_F^2$$

s.t. U, V_1, V_2 nonnegative

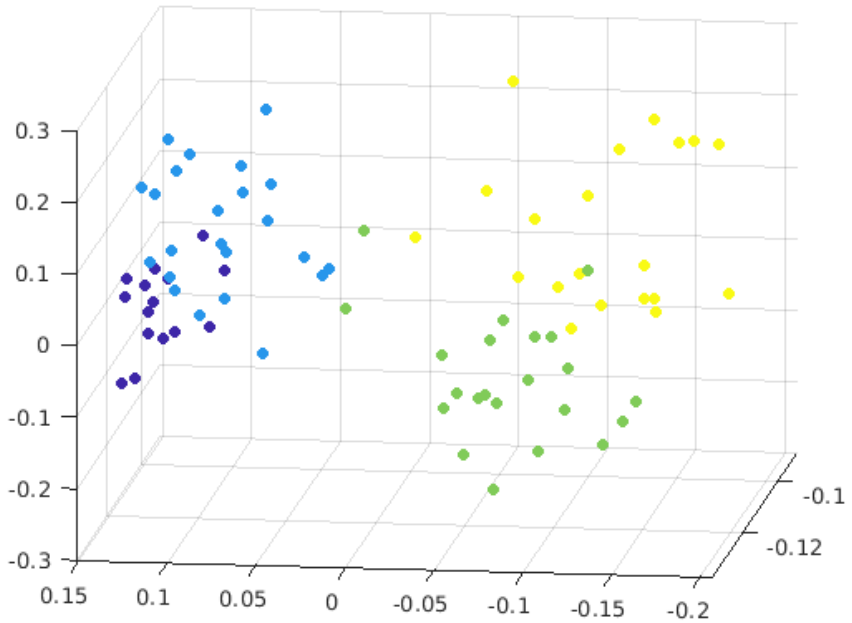
V_1, V_2, V_3 column-normalized

$U = Z \circ Z$, $V_1 = W_1 \circ W_1$, $V_2 = W_2 \circ W_2$

$V_3 \in Ob(n_3, r)$, $W_1 \in Ob^4(n_1, r)$, $W_2 \in Ob^4(n_2, r)$

CANCER CLASSIFICATION

For uveal melanoma data (TCGA):



First 3 principal components of U matrix

M. Pfeffer, A. Uschmajew, A. Amaro, U. Pfeffer:
Data fusion techniques for the integration of multi-domain genomic data from uveal melanoma.
Cancers, 11 (2019) 10, 1434

A. Amaro, **M. Pfeffer**, U. Pfeffer, F. Reggiani:
Evaluation and Comparison of Multi-Omics Data Integration Methods for Subtyping of Cutaneous Melanoma.
Biomedicines 10(12) (2022)

SIMULTANEOUS EIGENSPACES

SIMULTANEOUS EIGENSPACES

Let $A : \mathbb{R}^{m \times n} \longrightarrow \mathbb{R}^{m \times n}$ be a linear operator.

SIMULTANEOUS EIGENSPACES

Let $A : \mathbb{R}^{m \times n} \longrightarrow \mathbb{R}^{m \times n}$ be a linear operator.

Task: Find p lowest eigenvectors $X_1, \ldots, X_p$.

SIMULTANEOUS EIGENSPACES

Let $\mathcal{A} : \mathbb{R}^{m \times n} \longrightarrow \mathbb{R}^{m \times n}$ be a linear operator.

Task: Find p lowest eigenvectors $X_1, \ldots, X_p$.

Assume: X_i have low rank r.

SIMULTANEOUS EIGENSPACES

Let $\mathcal{A} : \mathbb{R}^{m \times n} \longrightarrow \mathbb{R}^{m \times n}$ be a linear operator.

Task: Find p lowest eigenvectors $X_1, \ldots, X_p$.

Assume: X_i have low rank r.

More restictive: X_i share the same column space $\text{span}(U)$, $U \in \mathbb{R}^{m \times r}$.

SIMULTANEOUS EIGENSPACES

Let $A : \mathbb{R}^{m \times n} \longrightarrow \mathbb{R}^{m \times n}$ be a linear operator.

Task: Find p lowest eigenvectors $X_1, \dots, X_p$.

Assume: X_i have low rank r.

More restictive: X_i share the same column space $\text{span}(U)$, $U \in \mathbb{R}^{m \times v}$.

Motivation: · Laplace Operator $\mathcal{L} = \Delta \otimes I + I \otimes \Delta$

SIMULTANEOUS EIGENSPACES

Let $\mathcal{A} : \mathbb{R}^{m \times n} \longrightarrow \mathbb{R}^{m \times n}$ be a linear operator.

Task: Find p lowest eigenvectors $X_1, \ldots, X_p$.

Assume: X_i have low rank r.

More restictive: X_i share the same column space $\operatorname{span}(U)$, $U \in \mathbb{R}^{m \times v}$.

Motivation:
- Laplace Operator $\mathcal{L} = \Delta \otimes I + I \otimes \Delta$
- Application in Quantum Chemistry

SIMULTANEOUS EIGENSPACES

Naive approach: · Compute $X_1, \dots, X_p$

SIMULTANEOUS EIGENSPACES

Naive approach:

- Compute $X_1, \dots, X_p$
- Set $X = [\mathrm{vec}(X_1), \dots, \mathrm{vec}(X_p)]$

SIMULTANEOUS EIGENSPACES

Naive approach:

- Compute $X_1, \dots, X_p$
- Set $$X = \left[\mathrm{vec}(X_1), \dots, \mathrm{vec}(X_p) \right]$$
- Reshape & truncate rank with SVD

SIMULTANEOUS EIGENSPACES

Naive approach:

- Compute $X_1, \dots, X_p$
- Set $X = \left[\mathrm{vec}(X_1), \dots, \mathrm{vec}(X_p)\right]$
- Reshape & truncate rank with SVD

Problems:

- Need to compute full eigenvectors

SIMULTANEOUS EIGENSPACES

Naive approach:

- Compute $X_1, \dots, X_p$
- Set $X = \left[\mathrm{vec}(X_1), \dots, \mathrm{vec}(X_p)\right]$
- Reshape & truncate rank with SVD

Problems:

- Need to compute full eigenvectors
- Solution optimal only in Frobenius norm and not in "norm" given by A

SIMULTANEOUS EIGENSPACES

Alternative approach:

Minimize $f(X) = \text{trace}\left(X^T A(X)\right)$

for $X \in St(mn, p) \cap \mathbb{R}_r^{m \times np}$

SIMULTANEOUS EIGENSPACES

Alternative approach:

Minimize $f(X) = \text{trace}\left(X^T A(X)\right)$

for $X \in St(mn, p) \cap \mathbb{R}_r^{m \times np}$

$X^T X = I$

SIMULTANEOUS EIGENSPACES

Alternative approach:

Minimize $f(X) = \operatorname{trace}\left(X^T A(X)\right)$

for $X \in St(mn, p) \cap \mathbb{R}_r^{m \times np}$

$\uparrow$ $\operatorname{rank}\left([X]_m^{np}\right) = r$

SIMULTANEOUS EIGENSPACES

Alternative approach:

Minimize $f(X) = \operatorname{trace}\left(X^T A(X)\right)$

for $X \in St(mn, p) \cap \mathbb{R}_r^{mxnp}$

transversal intersection

SIMULTANEOUS EIGENSPACES

Alternative approach:

Minimize $f(X) = \operatorname{trace}\left(X^T A(X)\right)$

for $X \in St(mn, p) \cap \mathbb{R}_r^{m \times np}$

transversal intersection

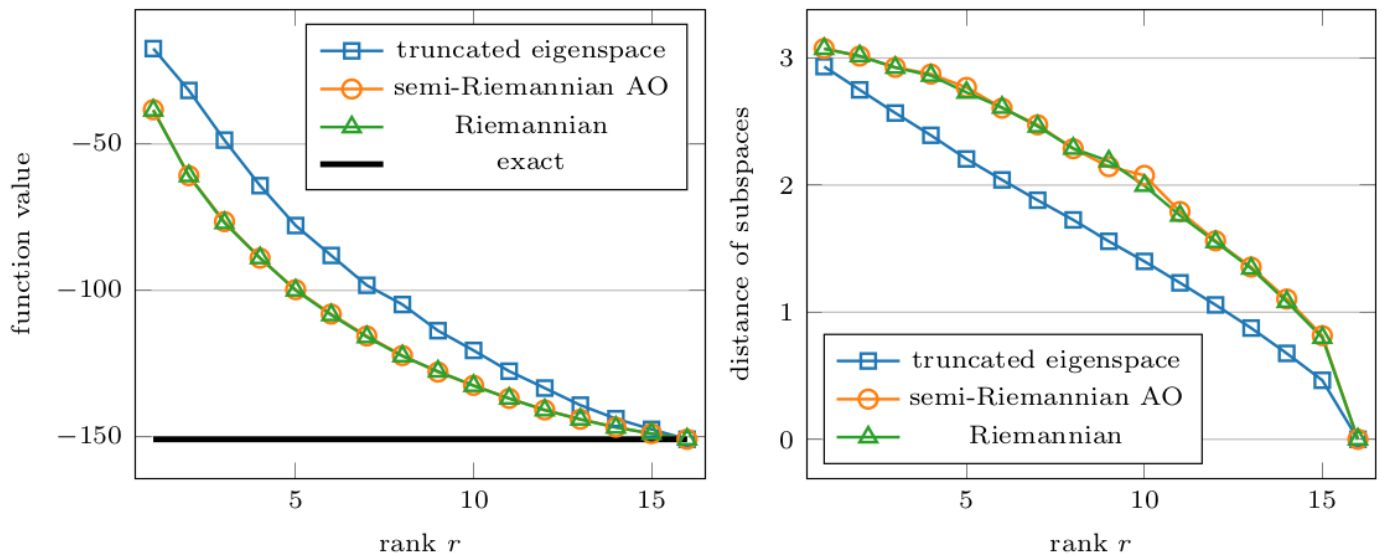
$\Rightarrow$ Riemannian manifold

SIMULTANEOUS EIGENSPACES

Alternative approach:

Minimize $f(X) = \operatorname{trace}\left(X^T A(X)\right)$

for $X \in St(mn, p) \cap \mathbb{R}_r^{m \times np}$



C. Krumnow, **M. Pfeffer**, A. Uschmajew:
Computing eigenspaces with low rank constraints.
SIAM Journal on Scientific Computing, 43 (2021) 1, p. 586-608

LOW-RANK & ROW-SPARSE RECOVERY

LOW-RANK & ROW-SPARSE RECOVERY

We aim to recover a low-rank and row-sparse signal X from m measurements:

LOW-RANK & ROW-SPARSE RECOVERY

We aim to recover a low-rank and row-sparse signal X from m measurements:

$$\min_{X \in \mathbb{R}_r^{M \times N}} \frac{1}{2} \| \mathcal{A}(X) - y \|_2^2 + \lambda \| X \|_{1,2}$$

LOW-RANK & ROW-SPARSE RECOVERY

We aim to recover a low-rank and row-sparse signal X from m measurements:

$$\min_{X \in \mathbb{R}_r^{n \times N}} \frac{1}{2} \| \mathcal{A}(X) - y \|_2^2 + \lambda \| X \|_{1,2}$$

Measurement operator

$$\mathcal{A} : \mathbb{R}^{n \times N} \to \mathbb{R}^m$$

LOW-RANK & ROW-SPARSE RECOVERY

We aim to recover a low-rank and row-sparse signal X from m measurements:

$$\min_{X \in \mathbb{R}_r^{M \times N}} \frac{1}{2} \| \mathcal{A}(X) - y \|_2^2 + \lambda \| X \|_{1,2}$$

y: measurements

LOW-RANK & ROW-SPARSE RECOVERY

We aim to recover a low-rank and row-sparse signal X from m measurements:

$$\min_{X \in \mathbb{R}_r^{M \times N}} \frac{1}{2} \| \mathcal{A}(X) - y \|_2^2 + \lambda \| X \|_{1,2}$$

↖ Rank-r manifold

LOW-RANK & ROW-SPARSE RECOVERY

We aim to recover a low-rank and row-sparse signal X from m measurements:

$$\min_{X \in \mathbb{R}_r^{M \times N}} \frac{1}{2} \| \mathcal{A}(X) - y \|_2^2 + \lambda \| X \|_{1,2}$$

λ: penalty parameter

LOW-RANK & ROW-SPARSE RECOVERY

We aim to recover a low-rank and row-sparse signal X from m measurements:

$$\min_{X \in \mathbb{R}_r^{M \times N}} \frac{1}{2} \| \mathcal{A}(X) - y \|_2^2 + \lambda \| X \|_{1,2}$$

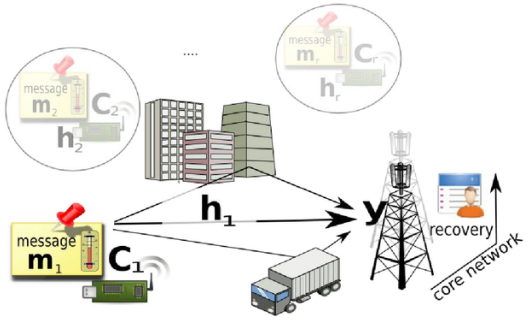
Application:

LOW-RANK & ROW-SPARSE RECOVERY

We aim to recover a low-rank and row-sparse signal X from m measurements:

$$\min_{X \in \mathbb{R}_r^{M \times N}} \frac{1}{2} \| \mathcal{A}(X) - y \|_2^2 + \lambda \| X \|_{1,2}$$

Application: Blind deconvolution

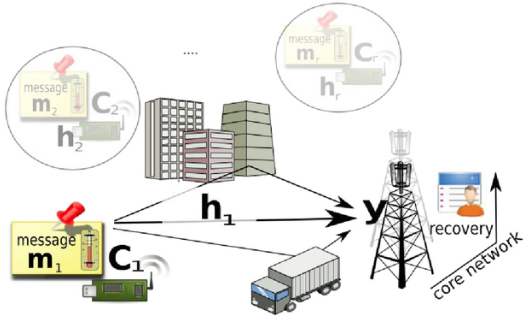


LOW-RANK & ROW-SPARSE RECOVERY

We aim to recover a low-rank and row-sparse signal X from m measurements:

$$\min_{X \in \mathbb{R}_r^{M \times N}} \frac{1}{2} \| \mathcal{A}(X) - y \|_2^2 + \lambda \| X \|_{1,2}$$

Application: Blind deconvolution



$\longrightarrow$ rank $(X) = 1$

LOW-RANK & ROW-SPARSE RECOVERY

Riemannian IHT:

LOW-RANK & ROW-SPARSE RECOVERY

Riemannian IHT:

- Perform Riemannian gradient step.

LOW-RANK & ROW-SPARSE RECOVERY

Riemannian IHT:

- Perform Riemannian gradient step.
- Truncate rows with small norm.

LOW-RANK & ROW-SPARSE RECOVERY

Riemannian IHT:

- Perform Riemannian gradient step.
- Truncate rows with small norm.

+ fast

LOW-RANK & ROW-SPARSE RECOVERY

Riemannian IHT:

- Perform Riemannian gradient step.
- Truncate rows with small norm.

+ fast

− Sparsity needs to be known

LOW-RANK & ROW-SPARSE RECOVERY

Riemannian IHT:

- Perform Riemannian gradient step.
- Truncate rows with small norm.

+ fast

− Sparsity needs to be known

Riemannian proximal gradient:

$$X_{k+1} = \operatorname{prox}_g^{\mathbb{R}_r^{n\times N}}\left(\mathcal{R}_{X_k}(\operatorname{grad} f(X_k))\right)$$

LOW-RANK & ROW-SPARSE RECOVERY

Riemannian IHT:

- Perform Riemannian gradient step.
- Truncate rows with small norm.

\+ fast

\- Sparsity needs to be known

Riemannian proximal gradient:

$$X_{k+1} = \operatorname{prox}_g^{\mathbb{R}_r^{n\times N}}\left(\mathcal{R}_{X_k}(\operatorname{grad} f(X_k))\right)$$

Riemannian gradient step for $f(X) = \frac{1}{2}\|\mathcal{A}(X) - y\|_2^2$

LOW-RANK & ROW-SPARSE RECOVERY

Riemannian IHT:

- Perform Riemannian gradient step.
- Truncate rows with small norm.

+ fast

− Sparsity needs to be known

Riemannian proximal gradient:

$$X_{k+1} = \operatorname{prox}_g^{\mathbb{R}_r^{n\times N}}\left(\mathcal{R}_{X_k}(\operatorname{grad} f(X_k))\right)$$

Proximal map for $g(X) = \lambda \|X\|_{1,2}$ on the low-rank manifold!

LOW-RANK & ROW-SPARSE RECOVERY

Riemannian IHT:

- Perform Riemannian gradient step.
- Truncate rows with small norm.

\+ fast

\- Sparsity needs to be known

Riemannian proximal gradient:

$$X_{k+1} = \operatorname{prox}_g^{\mathbb{R}_r^{m\times N}}\left(\mathcal{R}_{X_k}(\operatorname{grad} f(X_k))\right)$$

It holds: $\operatorname{prox}_g^{\mathbb{R}_r^{m\times N}}(Y) = S_{1,2}^{\lambda}(Y)$ for $Y \in \mathbb{R}_r^{M\times N}$

LOW-RANK & ROW-SPARSE RECOVERY

Riemannian IHT:

- Perform Riemannian gradient step.
- Truncate rows with small norm.

+ fast

− Sparsity needs to be known

Riemannian proximal gradient:

$$X_{k+1} = \operatorname{prox}_g^{\mathbb{R}_r^{n\times N}}\left(\mathcal{R}_{X_k}(\operatorname{grad} f(X_k))\right)$$

It holds: $\operatorname{prox}_g^{\mathbb{R}_r^{n\times N}}(Y) = S_{1,2}^{\lambda}(Y)$ for $Y \in \mathbb{R}_r^{M\times N}$

↑ row-soft thresholding

LOW-RANK & ROW-SPARSE RECOVERY

Riemannian IHT:

- Perform Riemannian gradient step.
- Truncate rows with small norm.

+ fast

− Sparsity needs to be known

Riemannian proximal gradient:

$$X_{k+1} = \operatorname{prox}_g^{\mathbb{R}_r^{n\times N}}\left(\mathcal{R}_{X_k}(\operatorname{grad} f(X_k))\right)$$

It holds: $\operatorname{prox}_g^{\mathbb{R}_r^{n\times N}}(Y) = S_{1,2}^{\lambda}(Y)$ for $Y \in \mathbb{R}_r^{M\times N}$

+ Sparsity can be unknown

LOW-RANK & ROW-SPARSE RECOVERY

Riemannian IHT:

- Perform Riemannian gradient step.
- Truncate rows with small norm.

+ fast

− Sparsity needs to be known

Riemannian proximal gradient:

$$X_{k+1} = \operatorname{prox}_g^{\mathbb{R}_r^{n\times N}}\left(\mathcal{R}_{X_k}(\operatorname{grad} f(X_k))\right)$$

It holds: $\operatorname{prox}_g^{\mathbb{R}_r^{n\times N}}(Y) = S_{1,2}^{\lambda}(Y)$ for $Y \in \mathbb{R}_r^{M\times N}$

+ Sparsity can be unknown

− slow (so far)

LOW-RANK & ROW-SPARSE RECOVERY

	Adaptive IHT		Adaptive RIHT		RPG	
ε	Iterations	CPU time	Iterations	CPU time	Iterations	CPU time
Random Rank One Measurements						
10^{-1}	196	0.4131s	196	0.2586s	6691	9.5539s
10^{-3}	831	1.7495s	831	1.1103s	11490	16.675s
10^{-5}	1583	3.3613s	1582	2.0928s	16095	23.931s
Fourier Measurements						
10^{-1}	8	0.1455s	10	0.0691s	4211	25.331s
10^{-3}	25	0.4592s	30	0.2189s	8941	58.069s
10^{-5}	45	0.8268s	46	0.3392s	13545	91.412s

H. Eisenmann, F. Krahmer, **M. Pfeffer**, A. Uschmajew:
Riemannian thresholding methods for row-sparse and low-rank matrix recovery.
Numerical Algorithms (2022)

COMPLETELY POSITIVE MATRICES

Let $M \in \mathbb{R}_+^{n \times n}$ be a symmetric nonnegative matrix.

COMPLETELY POSITIVE MATRICES

Let $M \in \mathbb{R}_+^{n \times n}$ be a symmetric nonnegative matrix.

- $M \in DNN_n$: M is also positive semidefinite.

COMPLETELY POSITIVE MATRICES

Let $M \in \mathbb{R}_+^{n \times n}$ be a symmetric nonnegative matrix.

- $M \in DNN_n$: M is also positive semidefinite.
- $M \in \mathcal{CP}_n$: $M = XX^T$ for $X \in \mathbb{R}_+^{n \times r}$ nonnegative.

COMPLETELY POSITIVE MATRICES

Let $M \in \mathbb{R}_+^{n \times n}$ be a symmetric nonnegative matrix.

- $M \in DNN_n$: M is also positive semidefinite.
- $M \in \mathcal{CP}_n$: $M = XX^T$ for $X \in \mathbb{R}_+^{n \times r}$ nonnegative.

It turns out: $\mathcal{CP}_n \subsetneq DNN_n$ for $n > 4$.

COMPLETELY POSITIVE MATRICES

Let $M \in \mathbb{R}_+^{n \times n}$ be a symmetric nonnegative matrix.

- $M \in DNN_n$: M is also positive semidefinite.
- $M \in \mathcal{CP}_n$: $M = XX^T$ for $X \in \mathbb{R}_+^{n \times r}$ nonnegative.

It turns out: $\mathcal{CP}_n \subsetneq DNN_n$ for $n > 4$.

Facts:
- Finding X is a co-NP-complete problem.
- For $M \in \partial\mathcal{CP}_5$, X is unique.
- The polynomial equations describing $\partial\mathcal{CP}_5$ have very high degree.

COMPLETELY POSITIVE MATRICES

Minimize

$$f(B) = \| M - (B \circ B)(B \circ B)^T \|_F^2$$

COMPLETELY POSITIVE MATRICES

Minimize

$$f(B) = \| M - (B \circ B)(B \circ B)^T \|_F^2$$

$$A = \begin{pmatrix} 163 & 108 & 27 & 4 & 4 & 27 & 108 \\ 108 & 163 & 108 & 27 & 4 & 4 & 27 \\ 27 & 108 & 163 & 108 & 27 & 4 & 4 \\ 4 & 27 & 108 & 163 & 108 & 27 & 4 \\ 4 & 4 & 27 & 108 & 163 & 108 & 27 \\ 27 & 4 & 4 & 27 & 108 & 163 & 108 \\ 108 & 27 & 4 & 4 & 27 & 108 & 163 \end{pmatrix}$$

COMPLETELY POSITIVE MATRICES

$$B = \begin{pmatrix}
\sqrt{27} & \frac{51}{\sqrt{27}} & \sqrt{27} & 0 & 0 & 0 & 0 & \sqrt{\frac{2}{3}} & 0 & 0 & \sqrt{6} & \sqrt{6} & 0 & 0 \\
0 & \sqrt{27} & \frac{51}{\sqrt{27}} & \sqrt{27} & 0 & 0 & 0 & 0 & \sqrt{\frac{2}{3}} & 0 & 0 & \sqrt{6} & \sqrt{6} & 0 \\
0 & 0 & \sqrt{27} & \frac{51}{\sqrt{27}} & \sqrt{27} & 0 & 0 & 0 & 0 & \sqrt{\frac{2}{3}} & 0 & 0 & \sqrt{6} & \sqrt{6} \\
0 & 0 & 0 & \sqrt{27} & \frac{51}{\sqrt{27}} & \sqrt{27} & 0 & \sqrt{6} & 0 & 0 & \sqrt{\frac{2}{3}} & 0 & 0 & \sqrt{6} \\
0 & 0 & 0 & 0 & \sqrt{27} & \frac{51}{\sqrt{27}} & \sqrt{27} & \sqrt{6} & \sqrt{6} & 0 & 0 & \sqrt{\frac{2}{3}} & 0 & 0 \\
\sqrt{27} & 0 & 0 & 0 & 0 & \sqrt{27} & \frac{51}{\sqrt{27}} & 0 & \sqrt{6} & \sqrt{6} & 0 & 0 & \sqrt{\frac{2}{3}} & 0 \\
\frac{51}{\sqrt{27}} & \sqrt{27} & 0 & 0 & 0 & 0 & \sqrt{27} & 0 & 0 & \sqrt{6} & \sqrt{6} & 0 & 0 & \sqrt{\frac{2}{3}}
\end{pmatrix}.$$

M. Pfeffer, J. Samper: **The cone of 5×5 completely positive matrices.** (2021)

THANK YOU !