COMP 414/514: Optimization – Algorithms, Complexity and Approximations

Lecture 9

Overview

- In the previous lecture, we:
  - Started talking about non-convex optimization, where non-convexity is introduced by the constraints
  - We consider the special case of sparsity
  - We provide conditions that lead to global convergence guarantees

Overview

- In the previous lecture, we:
  - Started talking about non-convex optimization, where non-convexity is introduced by the constraints
  - We consider the special case of sparsity
  - We provide conditions that lead to global convergence guarantees

- For the next 2-3 lectures, we will consider (possibly) another case of non-convex constraints: **low-rank optimization**
  - We will provide motivation, background and alternative solutions
  - We will see that this structure provides **various ways** to be.. non-convex
  - We will focus on how we can **provably and efficiently solve** such problems

Overview

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{C} \end{aligned}$$

Overview

We will consider convex objectives..

$$\min_{x} \quad f(x)$$

$$\text{s.t.} \quad x \in \mathcal{C}$$

..over non−convex constraints

Overview

We will consider convex objectives..

$$\min_{x} \quad f(x)$$

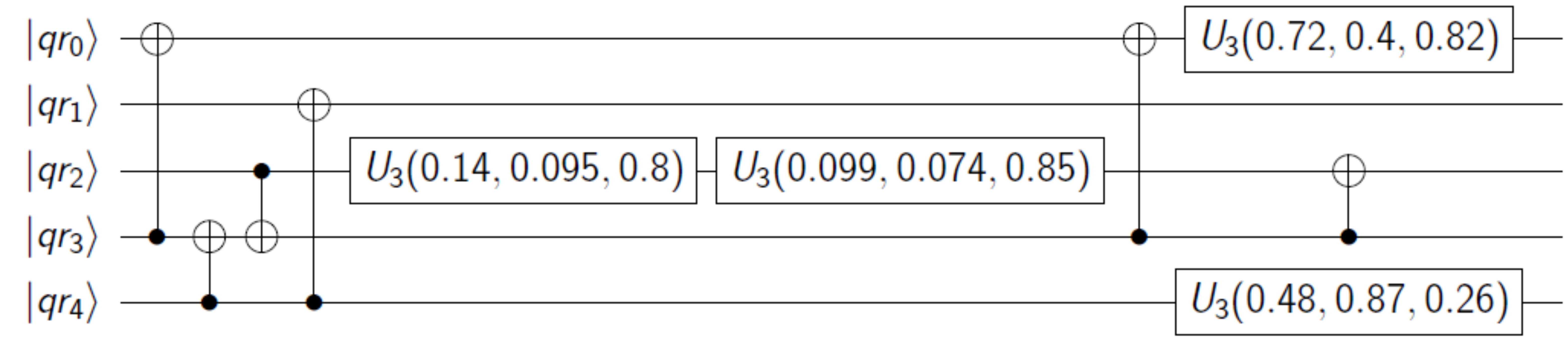
$$\text{s.t.} \quad \boxed{x \in \mathcal{C}}$$

..over non−convex constraints

– We will focus on the cases of (structured) sparsity and **low−rankness**

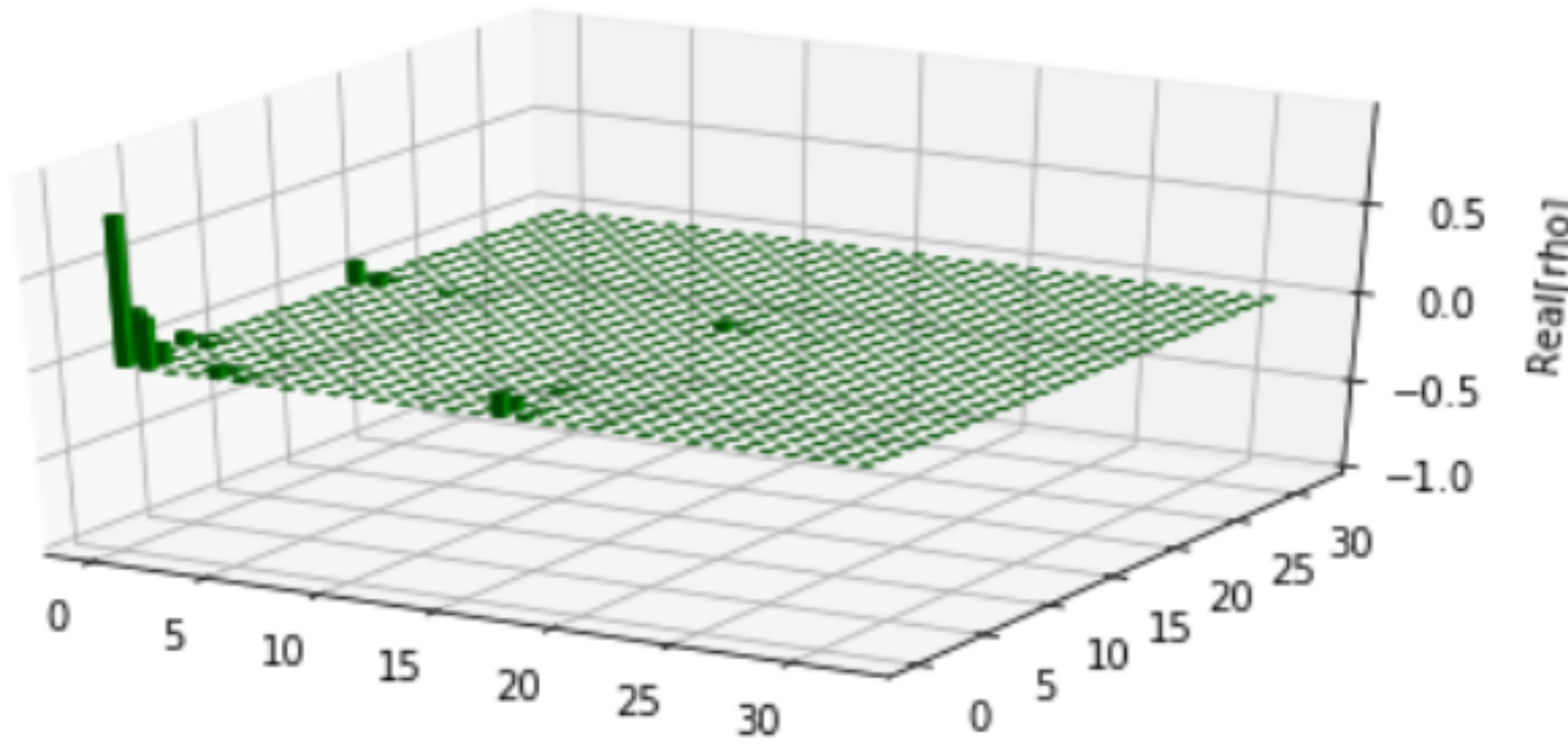
(But I open to other alternatives as we proceed)

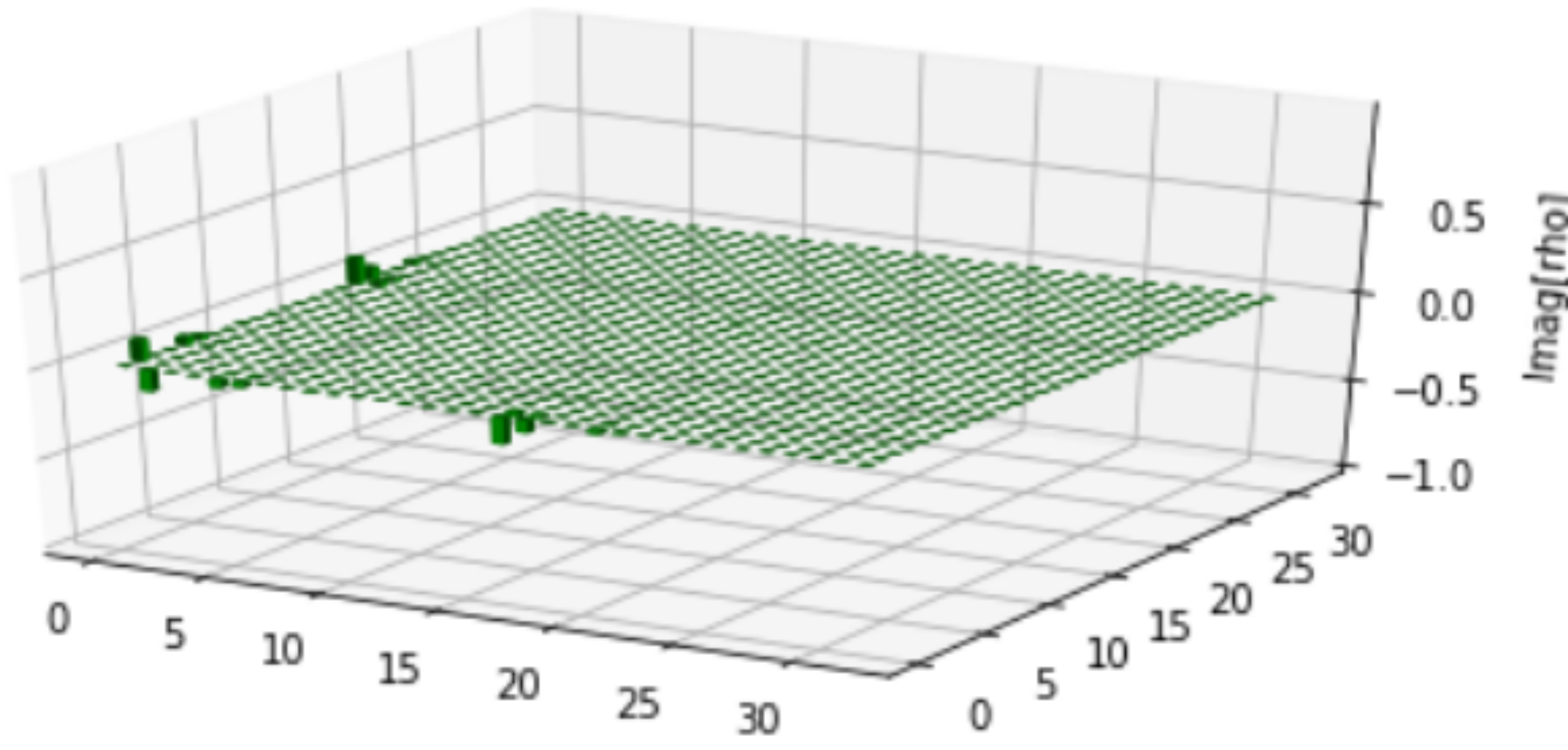
Problem setting via an application



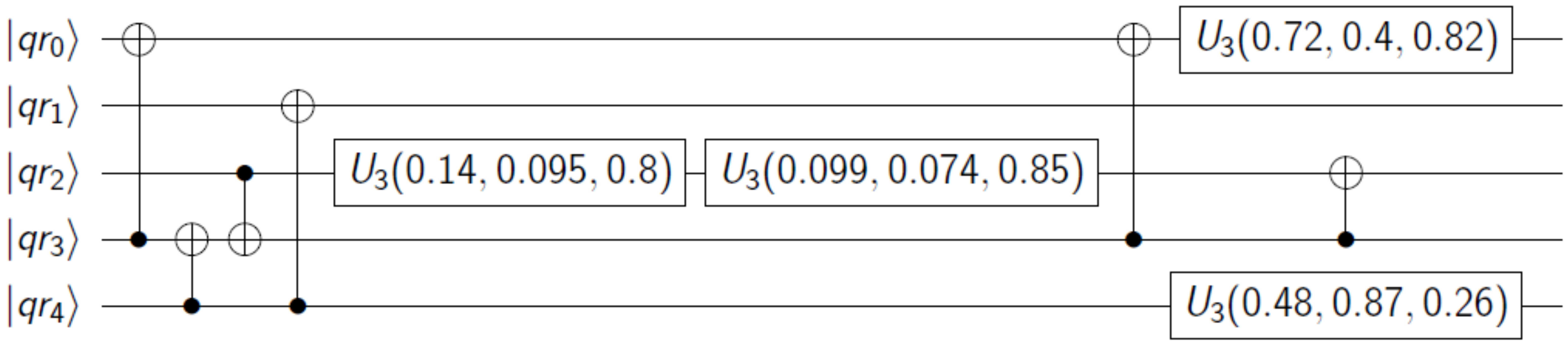

```
OPENQASM 2.0;
include "qelib1.inc";
qreg qr[5];
creg cr[5];
cx qr[3],qr[0];
cx qr[4],qr[3];
cx qr[2],qr[3];
cx qr[4],qr[1];
u3(0.139745784966679,0.0948307634768559,0.799402574081021) qr[2];
u3(0.0987633446591477,0.0737424336287251,0.850473826259255) qr[2];
cx qr[3],qr[0];
cx qr[3],qr[2];
u3(0.477009776552717,0.865309927771640,0.260492310391959) qr[4];
u3(0.719704686403954,0.398823542224269,0.824844977148233) qr[0];
```

```
fidelity: 0.997607
```



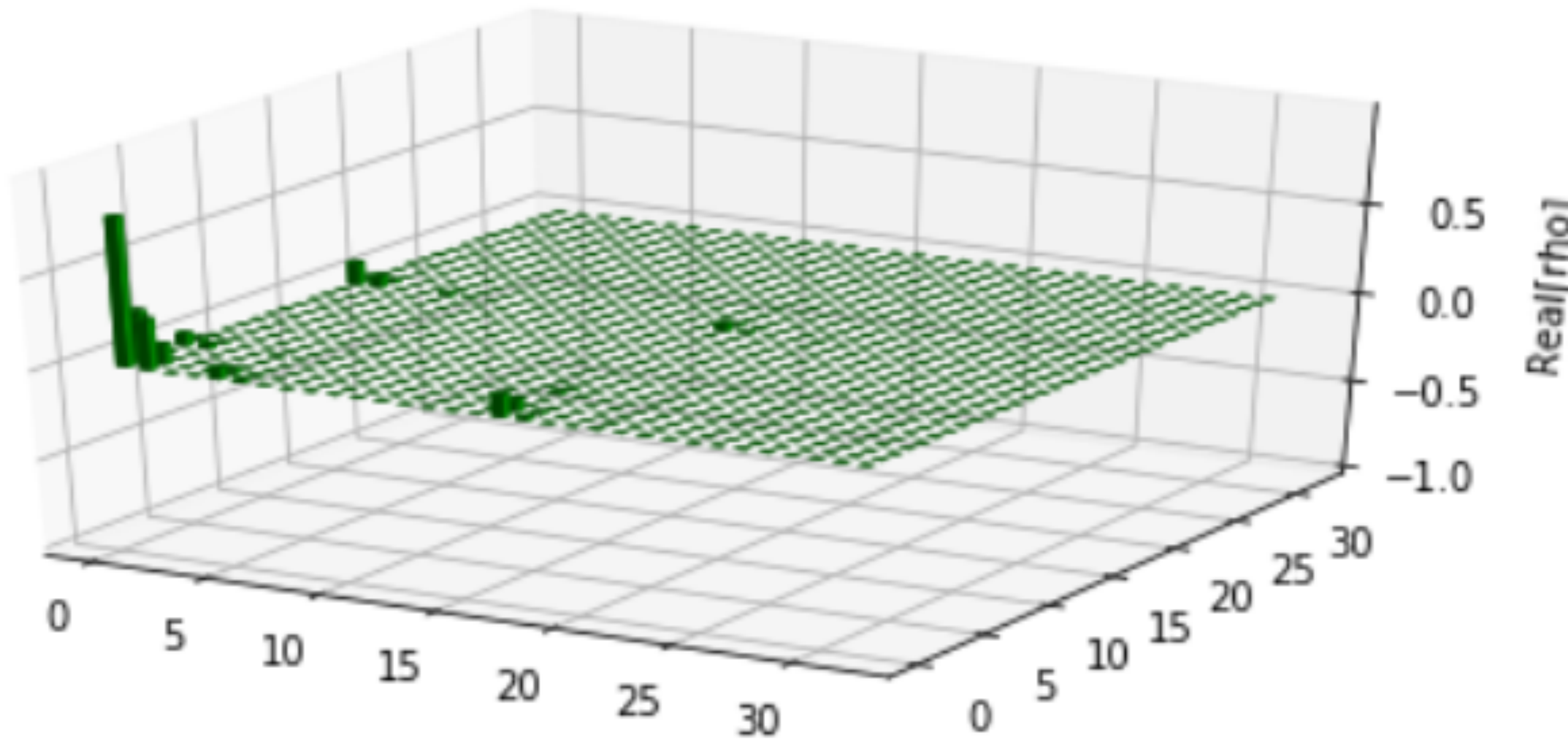


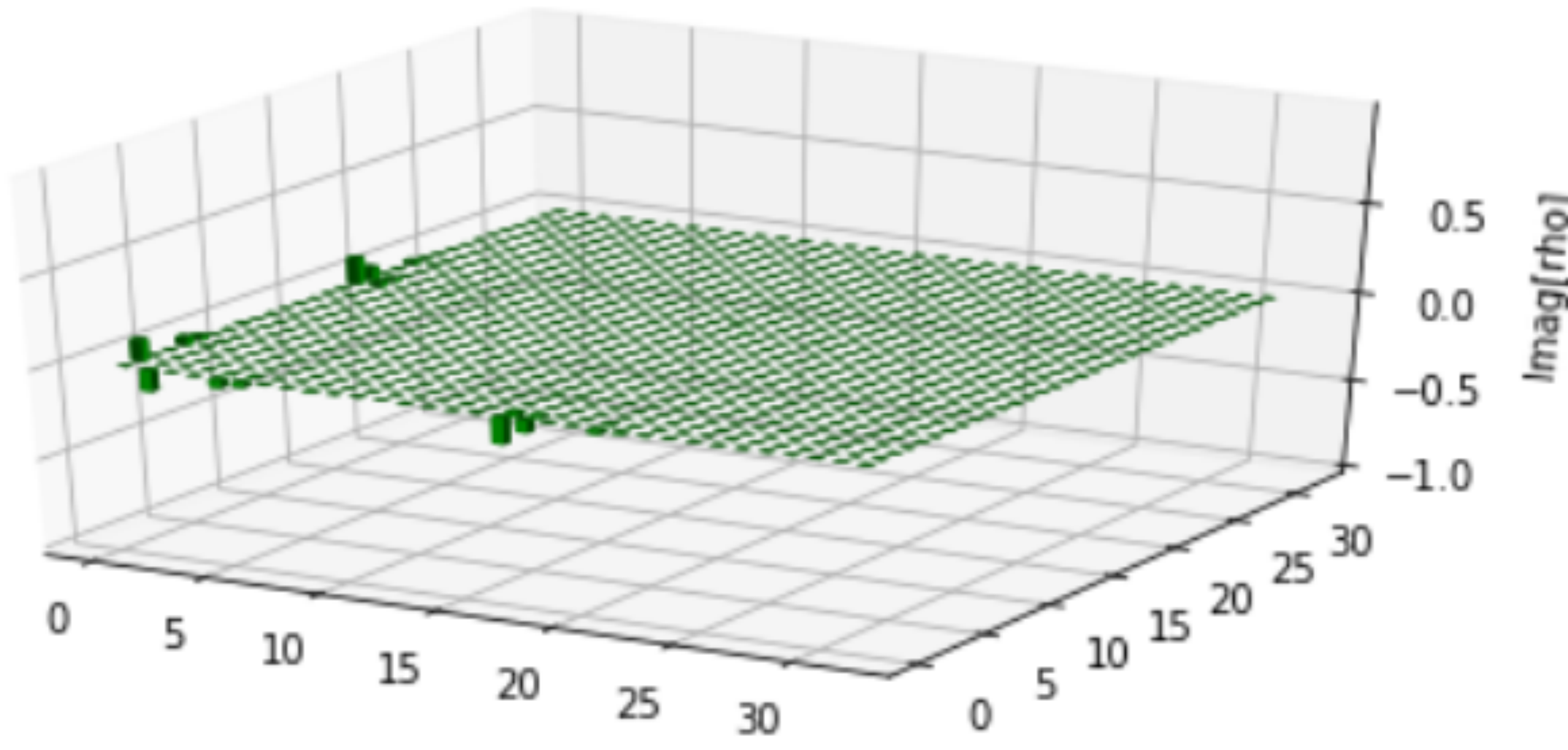
Problem setting via an application




```
OPENQASM 2.0;
include "qelib1.inc";
qreg qr[5];
creg cr[5];
cx qr[3],qr[0];
cx qr[4],qr[3];
cx qr[2],qr[3];
cx qr[4],qr[1];
u3(0.139745784966679,0.0948307634768559,0.799402574081021) qr[2];
u3(0.0987633446591477,0.0737424336287251,0.850473826259255) qr[2];
cx qr[3],qr[0];
cx qr[3],qr[2];
u3(0.477009776552717,0.865309927771640,0.260492310391959) qr[4];
u3(0.719704686403954,0.398823542224269,0.824844977148233) qr[0];
```

fidelity: 0.997607





– Goal: Validate the system is in the expected.. state, the computations are completed ..as expected

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \mathrm{Tr}(A_i X^\star) + w_i$

  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise

- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**

- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\mathrm{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \mathrm{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\mathrm{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
  - Some background:

1. Quantum computers can be described by their state they are in

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$

  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise

- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**

- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p,\ X^\star \succeq 0$

  - Some background:

1. Quantum computers can be described by their state they are in
2. The state of a quantum computer with q qubits is described by the **density matrix** in $\mathbb{C}^{2^q \times 2^q}$

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
  - Some background:

1. Quantum computers can be described by their state they are in
2. The state of a quantum computer with q qubits is described by the **density matrix** in $\mathbb{C}^{2^q \times 2^q}$
3. **An algorithm is a sequence of operations that transform the state of the quantum computer;** the final state is the answer to our question

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
  - Some background:

1. Quantum computers can be described by their state they are in
2. The state of a quantum computer with q qubits is described by the **density matrix** in $\mathbb{C}^{2^q \times 2^q}$
3. **An algorithm is a sequence of operations that transform the state of the quantum computer;** the final state is the answer to our question
4. A quantum computer is a **non-deterministic machine**: we don't know the final state, unless we measure it (this is where Schroedinger's cat come into the picture :))

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
  - Some background:

1. Quantum computers can be described by their state they are in
2. The state of a quantum computer with q qubits is described by the **density matrix** in $\mathbb{C}^{2^q \times 2^q}$
3. **An algorithm is a sequence of operations that transform the state of the quantum computer;** the final state is the answer to our question
4. A quantum computer is a **non-deterministic machine**: we don't know the final state, unless we measure it (this is where Schroedinger's cat come into the picture :))
5. But if we perform the steps "correctly", w.h.p. we measure the anticipated state

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
    - $A_i \in \mathbb{R}^{p \times p}$: features
    - $y_i \in \mathbb{R}$: responses
    - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
    - Some background:

6. Current implementations of quantum computers are more prototypes, rather not commercial

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
  - Some background:

6. Current implementations of quantum computers are more prototypes, rather not commercial
7. **We need verification tools to verify that quantum computers behave as anticipated**

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
  - Some background:

6. Current implementations of quantum computers are more prototypes, rather not commercial
7. **We need verification tools to verify that quantum computers behave as anticipated**
8. Quantum state tomography is one of such procedures: we can repeat the measurement many times, we keep the data, and we try to inverse the procedure to get the density matrix

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
  - Some background:

6. Current implementations of quantum computers are more prototypes, rather not commercial
7. **We need verification tools to verify that quantum computers behave as anticipated**
8. Quantum state tomography is one of such procedures: we can repeat the measurement many times, we keep the data, and we try to inverse the procedure to get the density matrix
9. Classical quantum state tomography is like solving linear equations; if we have a $O(4^q)$ object to recover, we need that many measurements

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
  - Some background:

6. Current implementations of quantum computers are more prototypes, rather not commercial
7. **We need verification tools to verify that quantum computers behave as anticipated**
8. Quantum state tomography is one of such procedures: we can repeat the measurement many times, we keep the data, and we try to inverse the procedure to get the density matrix
9. Classical quantum state tomography is like solving linear equations; if we have a $O(4^q)$ object to recover, we need that many measurements
10. When q = 20 or even 50, do the math

Quantum state tomography

(Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \mathrm{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\mathrm{rank}(X^\star) = r \ll p,\ X^\star \succeq 0$
  - Some background:

11. Why assume that the state is low-rank? These are called **pure** states - can be considered as a first step before going into more **mixed** states.

Quantum state tomography

(Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
  - Some background:

11. Why assume that the state is low-rank? These are called **pure** states – can be considered as a first step before going into more **mixed** states.

12. Theoretically, we can assume rank-1 constructed density matrices; noise + other Phenomena increases the rank in practice

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \mathrm{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\mathrm{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
- How do we measure?

Quantum state tomography

(Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
- How do we measure?

(Pauli operators)

1. Select: $A_i = \sigma_{i_1} \otimes \sigma_{i_2} \otimes \cdots \otimes \sigma_{i_q}$, where

$$\sigma_I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$

  - $A_i \in \mathbb{R}^{p \times p}$: features – $y_i \in \mathbb{R}$: responses – $w_i \in \mathbb{R}$: additive noise

- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**

- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$

- How do we measure? (Pauli operators)

1. Select: $A_i = \sigma_{i_1} \otimes \sigma_{i_2} \otimes \cdots \otimes \sigma_{i_q}$, where

$$\sigma_I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

2. Applying it to the system is equivalent (for the moment) with

$$y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$$

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p,\ X^\star \succeq 0$
- How do we solve for $X^\star \in \mathbb{R}^{p \times p}$, **without any prior information**?

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$

  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise

- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**

- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$

- How do we solve for $X^\star \in \mathbb{R}^{p \times p}$, **without any prior information**?

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2$$

$$\text{s.t.} \quad X \succeq 0, \; \text{Tr}(X) \leq 1$$

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
- How do we solve for $X^\star \in \mathbb{R}^{p \times p}$, **without any prior information**?

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle A_i, X \rangle\right)^2$$

$$\text{s.t.} \quad X \succeq 0, \ \text{Tr}(X) \leq 1$$

- X has $O(4^q)$ parameters
- This means that we need that many measurements

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \text{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\text{rank}(X^\star) = r \ll p,\ X^\star \succeq 0$
- What if we assume $X^\star \in \mathbb{R}^{p \times p}$, is of **low rank**?

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle A_i, X \rangle \right)^2$$

$$\text{s.t.} \quad X \succeq 0, \text{Tr}(X) \leq 1, \text{rank}(X) \leq r$$

Quantum state tomography (Much easier than it sounds like..)

- Generative model: $y_i = \langle A_i, X^\star \rangle + w_i = \mathrm{Tr}(A_i X^\star) + w_i$
  - $A_i \in \mathbb{R}^{p \times p}$: features
  - $y_i \in \mathbb{R}$: responses
  - $w_i \in \mathbb{R}$: additive noise
- **Disclaimer: quantum state tomography operates on complex numbers here, for simplicity, we assume real numbers**
- Generative prior: $X^\star \in \mathbb{R}^{p \times p}$ is rank-r and PSD: $\mathrm{rank}(X^\star) = r \ll p$, $X^\star \succeq 0$
- What if we assume $X^\star \in \mathbb{R}^{p \times p}$, is of **low rank**?

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle A_i, X \rangle \right)^2$$

$$\text{s.t.} \quad X \succeq 0, \mathrm{Tr}(X) \leq 1, \mathrm{rank}(X) \leq r$$

- X has $O(2^q r)$ parameters
- If rank is small compared to ambient dimension, then there is hope

Quantum state tomography (Much easier than it sounds like..)

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2$$

$$\text{s.t.} \quad X \succeq 0, \mathrm{Tr}(X) \leq 1, \mathrm{rank}(X) \leq r$$

Quantum state tomography (Much easier than it sounds like..)

$$\begin{aligned} \min_{X \in \mathbb{R}^{p \times p}} \quad & \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2 \\ \text{s.t.} \quad & X \succeq 0, \mathrm{Tr}(X) \leq 1, \mathrm{rank}(X) \leq r \end{aligned}$$

– Can we recover $X^\star \in \mathbb{R}^{p \times p}$ from limited set of measurements?

RIP for Pauli operators

$$(1 - \delta) \|X\|_F^2 \leq \|\mathcal{A}(X)\|_2^2 \leq (1 + \delta) \|X\|_F^2, \quad \forall \text{ rank-}r \ X \in \mathbb{R}^{p \times p}$$

$$[\mathcal{A}(X)]_i = \mathrm{Tr}(A_i, X)$$

(RIP also holds for (sub−)Gaussian matrices, Fourier, etc.)

– Similar to the sparsity case, RIP leads to convergence for various algos

Matrix sensing

(without the trace and PSD constraints)

$$
\begin{aligned}
\min_{X \in \mathbb{R}^{p \times p}} \quad & \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle A_i, X \rangle\right)^2 \\
\text{s.t.} \quad & \operatorname{rank}(X) \leq r
\end{aligned}
$$

Matrix sensing

(without the trace and PSD constraints)

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2$$
$$\text{s.t.} \quad \text{rank}(X) \leq r$$

Nuclear norm min.

– Solution #1: **convexification** + proj. gradient descent

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2$$
$$\text{s.t.} \quad \|X\|_* \leq \lambda$$

$\longrightarrow$

$$X_{t+1} = \Pi_{\|\cdot\|_* \leq \lambda} (X_t - \eta \nabla f(X_t))$$

(Pros & Cons?)

Matrix sensing

(without the trace and PSD constraints)

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle A_i, X \rangle \right)^2$$
$$\text{s.t.} \quad \text{rank}(X) \leq r$$

Nuclear norm min.

– Solution #1: **convexification** + proj. gradient descent

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle A_i, X \rangle \right)^2$$
$$\text{s.t.} \quad \|X\|_* \leq \lambda$$

$\longrightarrow$ $X_{t+1} = \Pi_{\|\cdot\|_* \leq \lambda} \left(X_t - \eta \nabla f(X_t) \right)$

(Pros & Cons?)

– Definition of the **nuclear norm**: $\|X\|_* = \sum_{i=1}^{p} \sigma_i(X)$

(Requires full SVD for its calculation)

Matrix sensing

(without the trace and PSD constraints)

$$\begin{aligned} \min_{X \in \mathbb{R}^{p \times p}} \quad & \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2 \\ \text{s.t.} \quad & \text{rank}(X) \leq r \end{aligned}$$

Matrix sensing

(without the trace and PSD constraints)

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2$$

$$\text{s.t.} \quad \text{rank}(X) \leq r$$

Hard-thresholding

– Solution #2: **keep the rank−constraint** + proj. gradient descent (Non−convex)

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2 \quad \text{s.t.} \quad \text{rank}(X) \leq r \quad \longrightarrow \quad X_{t+1} = \Pi_{\text{rank}(X) \leq r} \left(X_t - \eta \nabla f(X_t) \right)$$

(Pros & Cons?)

Matrix sensing

(without the trace and PSD constraints)

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2}\sum_{i=1}^{n} \left(y_i - \langle A_i, X \rangle\right)^2$$
$$\text{s.t.} \quad \text{rank}(X) \leq r$$

Hard-thresholding

– Solution #2: **keep the rank-constraint** + proj. gradient descent (Non-convex)

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2}\sum_{i=1}^{n} \left(y_i - \langle A_i, X \rangle\right)^2 \quad \longrightarrow \quad X_{t+1} = \Pi_{\text{rank}(X) \leq r}\left(X_t - \eta \nabla f(X_t)\right)$$
$$\text{s.t.} \quad \text{rank}(X) \leq r$$

(Pros & Cons?)

– Definition of the projection onto low-rank matrices

$$\widehat{X} \in \min_{X} \quad \frac{1}{2}\|X - Y\|_F^2$$
$$\text{s.t.} \quad \text{rank}(X) \leq r$$

(Requires truncated SVD for its calculation)

But before we proceed..

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2$$

$$\text{s.t.} \quad \text{rank}(X) \leq r$$

– Some questions:

But before we proceed..

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2 \quad \text{s.t.} \quad \text{rank}(X) \leq r$$

- Some questions:
  - Q: ``How easy it is to solve rank-constrained problems?``
  - A: ``Low-rankness makes problems exponentially hard to solve``

(This assumes the most general case)

But before we proceed..

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle A_i, X \rangle\right)^2$$
$$\text{s.t.} \quad \text{rank}(X) \leq r$$

- Some questions:
  - Q: "How easy it is to solve rank-constrained problems?"
  - A: "Low-rankness makes problems exponentially hard to solve"
    (This assumes the most general case)
  - Q: "But isn't the problem underdetermined?"
  - A: "Yes, without any constraints, the problem has infinite solutions"

But before we proceed..

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2$$
$$\text{s.t.} \quad \text{rank}(X) \leq r$$

- Some questions:
  - Q: ``How easy it is to solve rank-constrained problems?``
  - A: ``Low-rankness makes problems exponentially hard to solve``

    (This assumes the most general case)
  - Q: ``But isn't the problem underdetermined?``
  - A: ``Yes, without any constraints, the problem has infinite solutions``
  - Q: ``Why then do we have hopes solving this problem?``
  - A: ``Similar to sparsity, under assumptions on average this problem can be solved in polynomial time``

Iterative hard thresholding (IHT)

(It is just projected gradient descent on low-rank constraints)

Iterative hard thresholding (IHT)

(It is just projected gradient descent on low−rank constraints)

– Matrix IHT:

$$X_{t+1} = H_r\left(X_t - \eta \nabla f(X_t)\right)$$

where $$H_r(Z) \in \min_{X \in \mathbb{R}^{p \times p}} \|X - Z\|_F^2 \quad \text{s.t.} \quad \text{rank}(X) \leq r$$

Iterative hard thresholding (IHT)

(It is just projected gradient descent on low−rank constraints)

– Matrix IHT:

$$X_{t+1} = H_r\left(X_t - \eta \nabla f(X_t)\right)$$

where $$H_r(Z) \in \min_{X \in \mathbb{R}^{p \times p}} \quad \|X - Z\|_F^2 \quad \text{s.t.} \quad \text{rank}(X) \leq r$$

(Have we seen this before?)

Iterative hard thresholding (IHT)

(It is just projected gradient descent on low-rank constraints)

– Matrix IHT:

$$X_{t+1} = H_r\left(X_t - \eta \nabla f(X_t)\right)$$

where $$H_r(Z) \in \min_{X \in \mathbb{R}^{p \times p}} \|X - Z\|_F^2 \quad \text{s.t.} \quad \text{rank}(X) \leq r$$

(Have we seen this before?)

(If yes, how we solve it?)

Iterative hard thresholding (IHT)

(It is just projected gradient descent on low−rank constraints)

- Matrix IHT:

$$X_{t+1} = H_r\left(X_t - \eta \nabla f(X_t)\right)$$

where $$H_r(Z) \in \min_{X \in \mathbb{R}^{p \times p}} \|X - Z\|_F^2 \quad \text{s.t.} \quad \text{rank}(X) \leq r$$

(Have we seen this before?)

(If yes, how we solve it?)

- Now, imagine yourself implementing this.. What are the hyper−parameters?

Iterative hard thresholding (IHT)

(It is just projected gradient descent on low−rank constraints)

- Matrix IHT:

$$X_{t+1} = H_r\left(X_t - \eta \nabla f(X_t)\right)$$

where $H_r(Z) \in \min_{X \in \mathbb{R}^{p \times p}} \|X - Z\|_F^2$

s.t. $\text{rank}(X) \leq r$

(Have we seen this before?)

(If yes, how we solve it?)

- Now, imagine yourself implementing this.. What are the hyper−parameters?
  - "How do we set the step size?"
  - "How do we select the initial point? (it is non−convex after all)"
  - "What if we don't know the sparsity level?"
  - "Are there any other tricks we can pull−off?"

Iterative hard thresholding (IHT)

(It is just projected gradient descent on low−rank constraints)

- Matrix IHT:

$$X_{t+1} = H_r\left(X_t - \eta \nabla f(X_t)\right)$$

where $H_r(Z) \in \min_{X \in \mathbb{R}^{p \times p}} \quad \|X - Z\|_F^2$

$\text{s.t.} \quad \text{rank}(X) \leq r$

(Have we seen this before?)

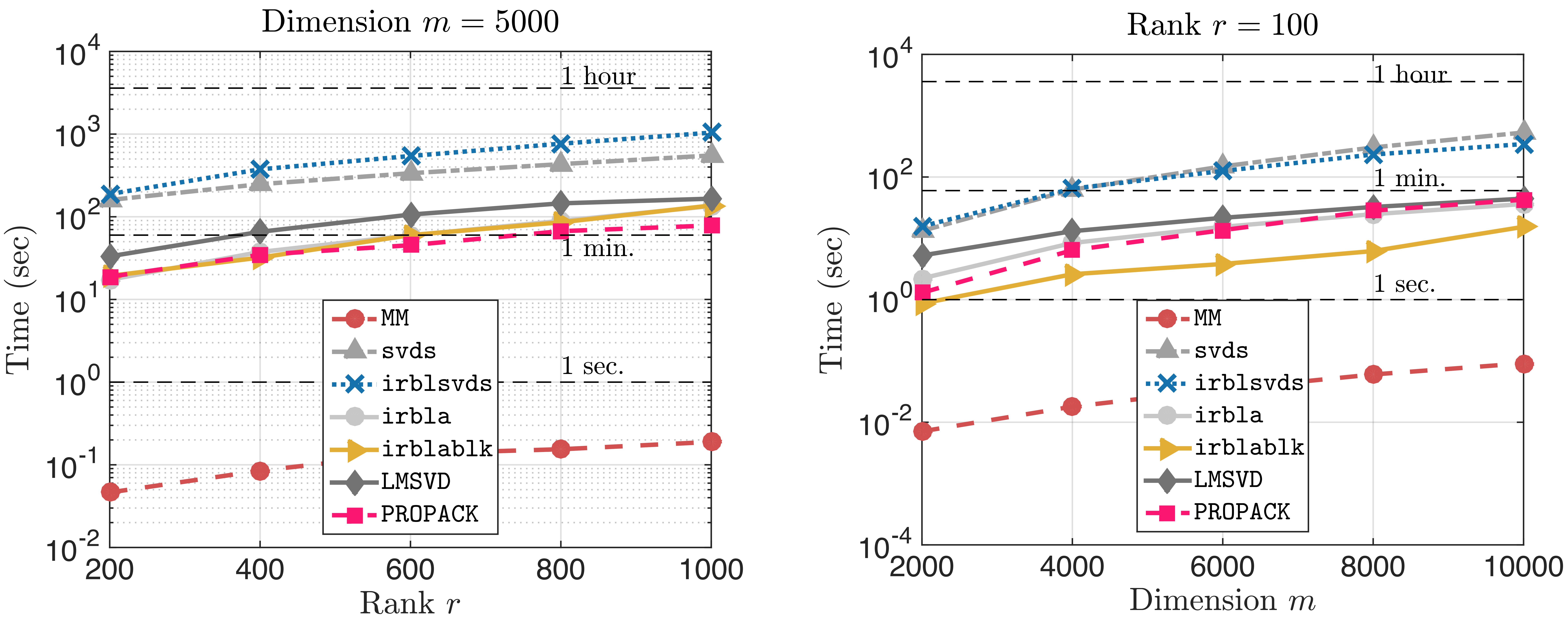
(If yes, how we solve it?)

- Now, imagine yourself implementing this.. What are the hyper−parameters?
  - ``How do we set the step size?``
  - ``How do we select the initial point? (it is non−convex after all)``
  - ``What if we don't know the sparsity level?``
  - ``Are there any other tricks we can pull−off?`` (Answer: see previous Chapter)

Convexification vs. hard−thresholding in practice

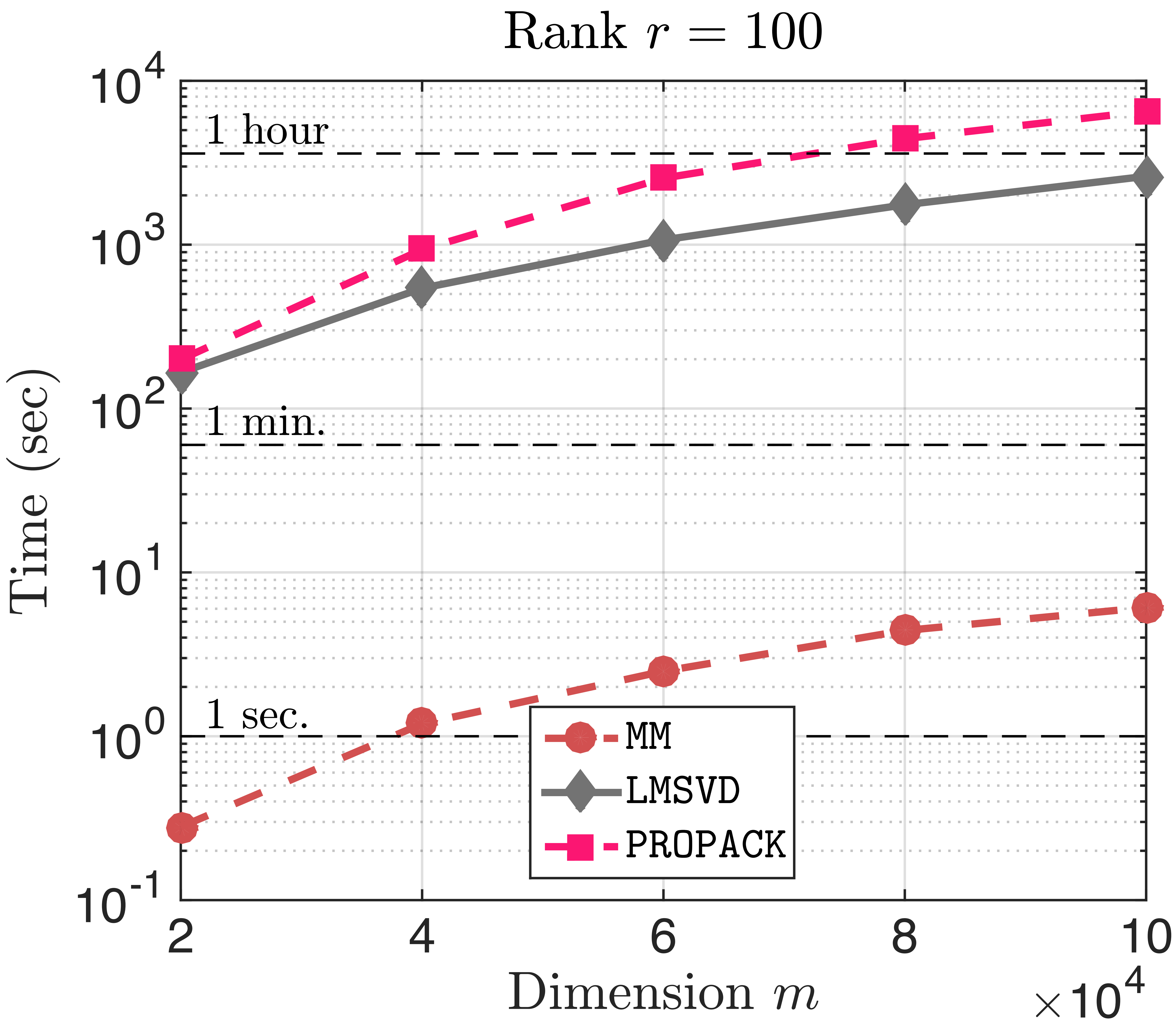
Demo

The price of SVD



$$\texttt{SVD}(X) \text{ vs. } X \cdot U, \text{ where } X \in \mathbb{R}^{m \times m}, \ U \in \mathbb{R}^{m \times r}$$

The price of SVD



$\texttt{SVD}(X)$ vs. $X \cdot U$, where $X \in \mathbb{R}^{m \times m}$, $U \in \mathbb{R}^{m \times r}$

$$X = UV^\top$$

Non-PSD

$X \in \mathbb{R}^{n \times p}$

$U \in \mathbb{R}^{n \times r}$

$V \in \mathbb{R}^{p \times r}$

PSD

$X \in \mathbb{R}^{n \times n}$

$U = V \in \mathbb{R}^{n \times r}$

First consider a simpler objective: Rank-1 PCA

Whiteboard

First consider a simpler objective: Rank−1 PCA

- Some properties of the proof:
  - Initialization does matter: e.g., for PCA there are initializations that do not lead to convergence

(More to come later on)

First consider a simpler objective: Rank-1 PCA

- Some properties of the proof:
  - Initialization does matter: e.g., for PCA there are initializations that do not lead to convergence

    (More to come later on)

  - After proper initialization, one can prove convergence to global minimum. Despite this, such convergence results are called **local convergence guarantees**

First consider a simpler objective: Rank−1 PCA

- Some properties of the proof:
  - Initialization does matter: e.g., for PCA there are initializations that do not lead to convergence

    (More to come later on)

  - After proper initialization, one can prove convergence to global minimum. Despite this, such convergence results are called **local convergence guarantees**

  - Often the theory dictates how to set the step size, in order to obtain convergence. For some cases it is a range of values, in other cases we just rely on a specific step size.

Back to matrix sensing

$$
\begin{aligned}
\min_{X \in \mathbb{R}^{p \times p}} \quad & \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2 \\
\text{s.t.} \quad & \operatorname{rank}(X) \leq r
\end{aligned}
$$

Back to matrix sensing

$$\min_{X \in \mathbb{R}^{p \times p}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle A_i, X \rangle \right)^2$$

s.t. ~~$\operatorname{rank}(X) \leq r$~~

$$X = UV^\top$$

Back to matrix sensing

$$\min_{U \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{p \times r}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle A_i, UV^\top \rangle \right)^2$$

Back to matrix sensing

Non-convex!

$$\min_{U \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{p \times r}} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle A_i, UV^\top \rangle \right)^2$$

Back to matrix sensing

Non-convex!

$$\min_{U \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{p \times r}} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle A_i, UV^\top \rangle\right)^2$$

No constraints!

Back to matrix sensing

Non-convex!

$$\min_{U \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{p \times r}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle A_i, UV^\top \rangle \right)^2$$

No constraints!

- Key differences with PCA:
  - Number of observations less than number of parameters
  - Mapping is identity, but satisfies a restricted isometry property

The same story holds for more general functions

$$\min_{\substack{X \in \mathbb{R}^{m \times n} \\ \text{rank}(X) \leq r}} f(X)$$

The same story holds for more general functions

$$\min_{\substack{X \in \mathbb{R}^{m \times n} \\ \text{rank}(X) \leq r}} f(X)$$

$$X = UV^\top$$

The same story holds for more general functions

$$\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$$

The same story holds for more general functions

Non-convex!

No constraints!

$$\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$$

The same story holds for more general functions

No constraints!

Non-convex!

$$\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$$

- Key differences with matrix sensing:
  - Restricted isometry might be substituted by restricted strong cvx/smoothness
  - Restricted strong convexity might not hold

How would we solve this problem?

$$U_{\boldsymbol{i+1}} = U_i - \eta \nabla f(U_i {V_i}^\top) \cdot V_i$$

$$V_{\boldsymbol{i+1}} = V_i - \eta \nabla f(U_i {V_i}^\top)^\top \cdot U_i$$

How would we solve this problem?

$$U_{i+1} = U_i - \eta \overbrace{\nabla f(U_i V_i^\top) \cdot V_i}^{\text{Gradient of f w.r.t. U}}$$

$$V_{i+1} = V_i - \eta \underbrace{\nabla f(U_i V_i^\top)^\top \cdot U_i}_{\text{Gradient of f w.r.t. V}}$$

How would we solve this problem?

$$\boldsymbol{U_{i+1}} = U_i - \eta \nabla f(U_i {V_i}^\top) \cdot V_i$$

Select initial point

$$\boldsymbol{V_{i+1}} = V_i - \eta \nabla f(U_i {V_i}^\top)^\top \cdot U_i$$

How would we solve this problem?

$$U_{\boldsymbol{i+1}} = U_i - \eta \nabla f(U_i {V_i}^\top) \cdot V_i$$

Select initial point

Select step size

$$V_{\boldsymbol{i+1}} = V_i - \eta \nabla f(U_i {V_i}^\top)^\top \cdot U_i$$

How would we solve this problem?

Do gradient step

$$U_{i+1} = U_i - \eta \nabla f(U_i V_i^\top) \cdot V_i$$

Select initial point

Select step size

$$V_{i+1} = V_i - \eta \nabla f(U_i V_i^\top)^\top \cdot U_i$$

Do gradient step

How would we solve this problem?

$$U_{i+1} = U_i - \eta \nabla f(U_i {V_i}^\top) \cdot V_i$$

$$V_{i+1} = V_i - \eta \nabla f(U_i {V_i}^\top)^\top \cdot U_i$$

How would we solve this problem?

$$U_{i+1} = U_i - \eta \nabla f(U_i V_i^\top) \cdot V_i$$

$$V_{i+1} = V_i - \eta \nabla f(U_i V_i^\top)^\top \cdot U_i$$

How would we solve this problem?

$$U_{i+1} = U_i - \eta \nabla f(U_i V_i^\top) \cdot V_i$$

$$V_{i+1} = V_i - \eta \nabla f(U_i V_i^\top)^\top \cdot U_i$$

How would we solve this problem?

$$U_{i+1} V_{i+1}^{\top} = \text{rank-}r \text{ matrix}$$

How would we solve this problem?

- We solve:

$$\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$$

via:

$$U_{i+1} = U_i - \eta \nabla f(U_i V_i^\top) \cdot V_i^\top$$

$$V_{i+1} = V_i - \eta \nabla f(U_i V_i^\top)^\top \cdot U_i$$

How would we solve this problem?

- We solve:

$$\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$$

via:

$$U_{i+1} = U_i - \eta \nabla f(U_i V_i^\top) \cdot V_i^\top$$

$$V_{i+1} = V_i - \eta \nabla f(U_i V_i^\top)^\top \cdot U_i$$

Does $X \mapsto UV^\top$ introduce new global and local minima?

How would we solve this problem?

- We solve:

$$\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$$

via:

$$U_{i+1} = U_i - \eta \nabla f(U_i {V_i}^\top) \cdot {V_i}^\top$$

$$V_{i+1} = V_i - \eta \nabla f(U_i {V_i}^\top)^\top \cdot U_i$$

Does $X \mapsto UV^\top$ introduce new global and local minima?

Does initialization play key role?

How would we solve this problem?

- We solve:

$$\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$$

via:

$$U_{i+1} = U_i - \eta \nabla f(U_i V_i^\top) \cdot V_i^\top$$

$$V_{i+1} = V_i - \eta \nabla f(U_i V_i^\top)^\top \cdot U_i$$

Does $X \mapsto UV^\top$ introduce new global and local minima?

Does initialization play key role?

What about (local) convergence under assumptions on f?

How would we solve this problem?

- We solve:

$$\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$$

via:

$$U_{i+1} = U_i - \eta \nabla f(U_i V_i^\top) \cdot V_i^\top$$
$$V_{i+1} = V_i - \eta \nabla f(U_i V_i^\top)^\top \cdot U_i$$

Does $X \mapsto UV^\top$ introduce new global and local minima?

Does initialization play key role?

What about (local) convergence under assumptions on f?

How to initialize in practice (U_0, V_0) ?

Non−uniqueness of global minima

– Factors at $X^\star$ are not unique

Non−uniqueness of global minima

- Factors at $X^\star$ are not unique

$$X^\star = U^\star V^{\star\top} = U^\star R \cdot R^\top V^{\star\top} = \widehat{U}^\star \widehat{V}^{\star\top}$$

for all R such that $RR^\top = I$

Non−uniqueness of global minima

- Factors at $X^\star$ are not unique

$$X^\star = U^\star V^{\star\top} = U^\star R \cdot R^\top V^{\star\top} = \widehat{U}^\star \widehat{V}^{\star\top}$$

for all R such that $RR^\top = I$

- Example:

$$f(X) = \frac{1}{2} \cdot \|y - \texttt{vec}(A \cdot X)\|_2^2$$

where $X^\star = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ Unique! (r=1)

Non−uniqueness of global minima

- Factors at $X^\star$ are not unique

$$X^\star = U^\star V^{\star\top} = U^\star R \cdot R^\top V^{\star\top} = \widehat{U}^\star \widehat{V}^{\star\top}$$

for all R such that $RR^\top = I$

- Example:

$$f(X) = \tfrac{1}{2} \cdot \|y - \texttt{vec}(A \cdot X)\|_2^2$$

where $X^\star = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ Unique! (r=1)

In this case

$$U^\star = [1 \;\; 1]^\top \text{ or } [-1 \;\; -1]^\top$$

Non−uniqueness of global minima

– Factors at $X^\star$ are not unique

$$X^\star = U^\star V^{\star\top} = U^\star R \cdot R^\top V^{\star\top} = \widehat{U}^\star \widehat{V}^{\star\top}$$

for all R such that $RR^\top = I$

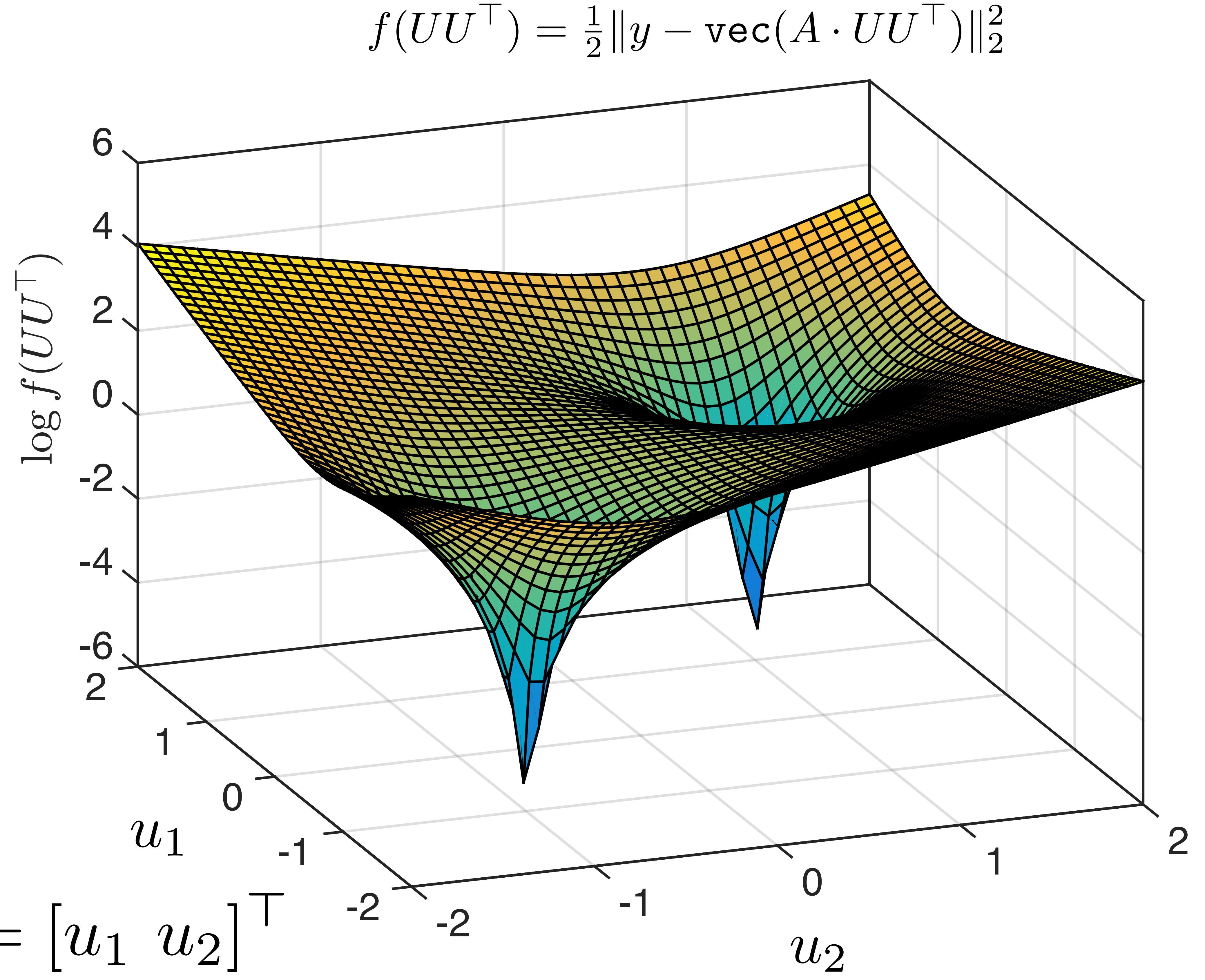
– Example:

$$f(X) = \tfrac{1}{2} \cdot \|y - \texttt{vec}(A \cdot X)\|_2^2$$

where $X^\star = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ Unique! (r=1)

In this case

$$U^\star = [1 \;\; 1]^\top \text{ or } [-1 \;\; -1]^\top$$



$U = [u_1 \;\; u_2]^\top$

Non−uniqueness of global minima

- Factors at $X^\star$ are not unique

$$X^\star = U^\star V^{\star\top} = U^\star R \cdot R^\top V^{\star\top} = \widehat{U}^\star \widehat{V}^{\star\top}$$

for all R such that $RR^\top = I$

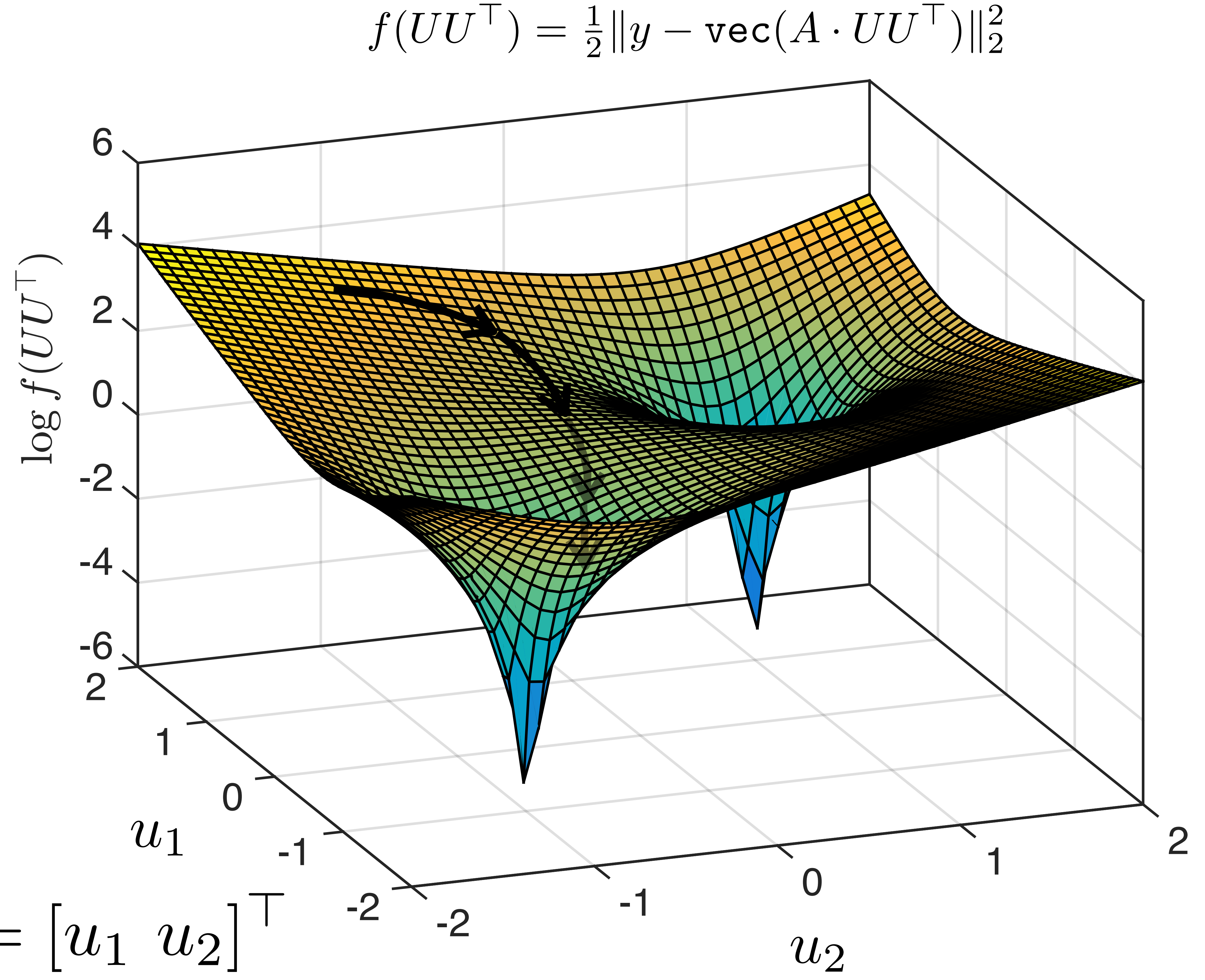
- Example:

$$f(X) = \frac{1}{2} \cdot \|y - \texttt{vec}(A \cdot X)\|_2^2$$

where $X^\star = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ Unique! (r=1)

In this case

$$U^\star = [1 \;\; 1]^\top \text{ or } [-1 \;\; -1]^\top$$



Non−uniqueness of global minima

- Factors at $X^\star$ are not unique

$$X^\star = U^\star V^{\star\top} = U^\star R \cdot R^\top V^{\star\top} = \widehat{U}^\star \widehat{V}^{\star\top}$$

for all R such that $RR^\top = I$

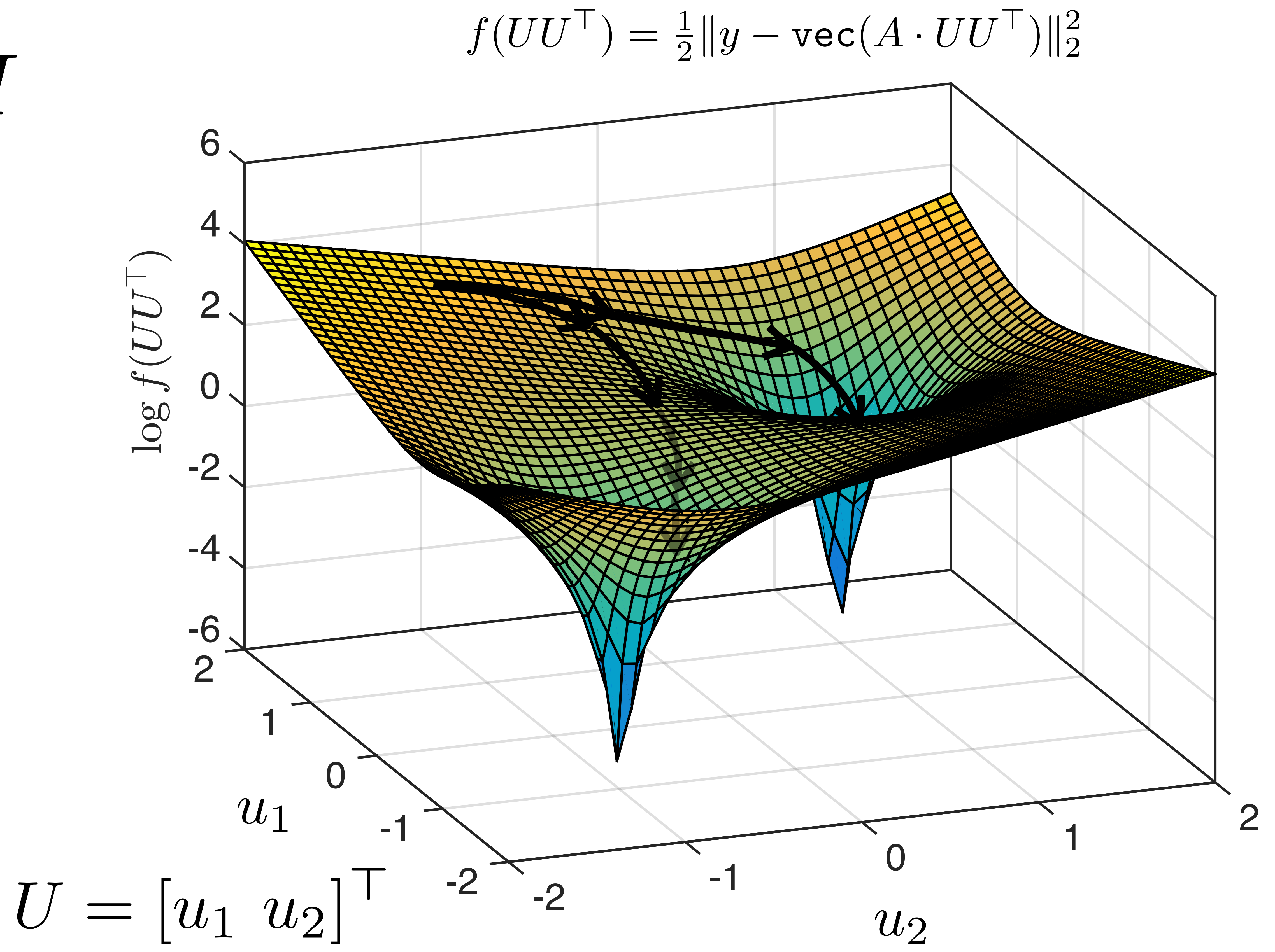
- Example:

$$f(X) = \frac{1}{2} \cdot \|y - \texttt{vec}(A \cdot X)\|_2^2$$

where $X^\star = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ Unique! (r=1)

In this case

$$U^\star = [1 \;\; 1]^\top \text{ or } [-1 \;\; -1]^\top$$



Non−uniqueness of global minima

– Factors at $X^\star$ are not unique

$$X^\star = U^\star V^{\star\top} = U^\star R \cdot R^\top V^{\star\top} = \widehat{U}^\star \widehat{V}^{\star\top}$$

for all R such that $RR^\top = I$

– Example:

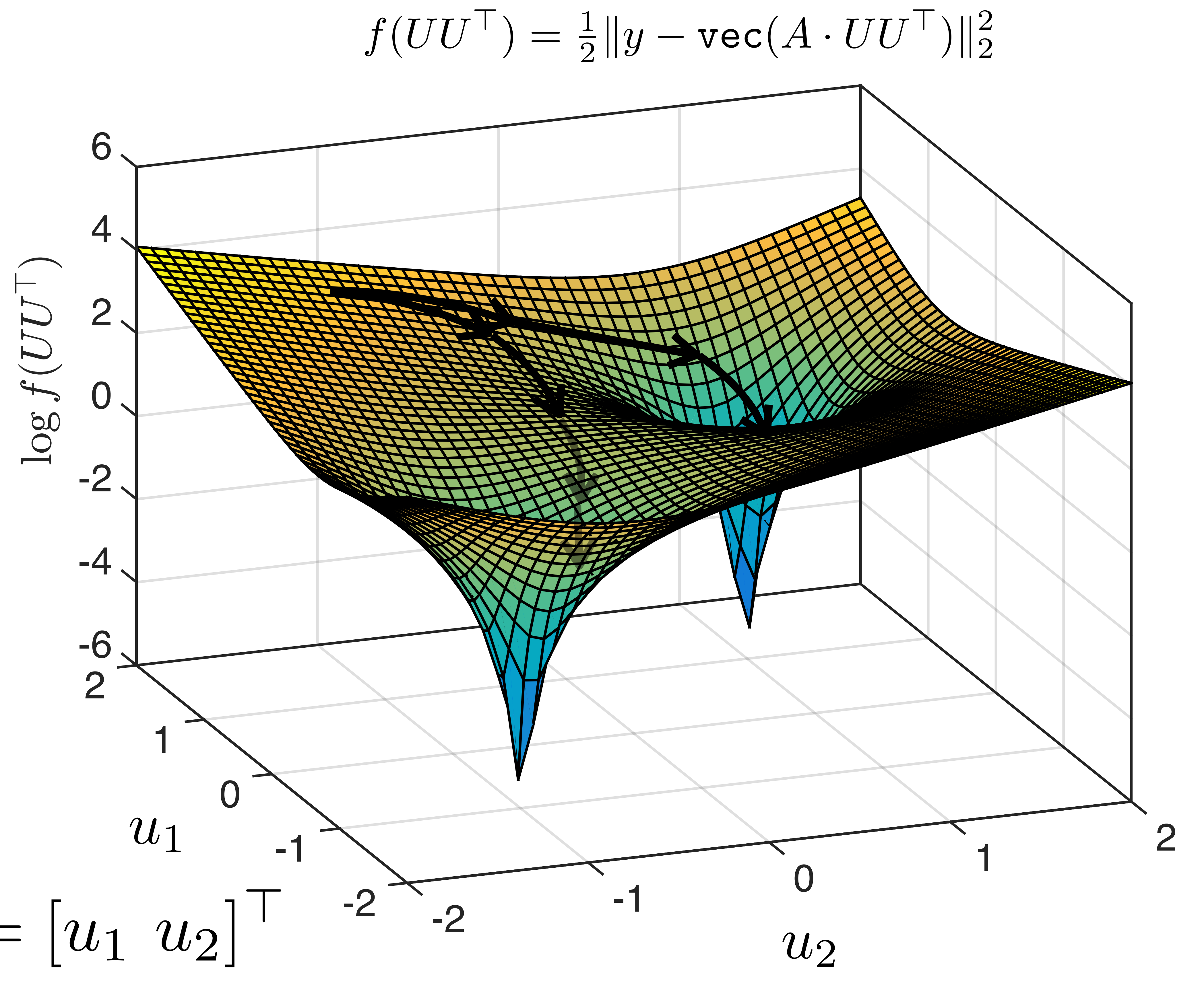
$$f(X) = \tfrac{1}{2} \cdot \|y - \texttt{vec}(A \cdot X)\|_2^2$$

where $X^\star = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ Unique! (r=1)

In this case

$$U^\star = [1 \;\; 1]^\top \text{ or } [-1 \;\; -1]^\top$$

- $X \mapsto UV^\top$ "ruins" convexity
- Even local convergence results are important



$U = [u_1 \;\; u_2]^\top$

What about local minima?

What about local minima?

- Factorization might also introduce local minima

What about local minima?

- Factorization might also introduce local minima
- Example: **Weighted low-rank approximation**

$$f(uu^\top) = \sum_{ij} W_{ij} \cdot (X^\star_{ij} - u_i u_j)^2 \quad \text{where} \quad X^\star = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad W = \begin{bmatrix} 100 & 1 \\ 1 & 100 \end{bmatrix}$$

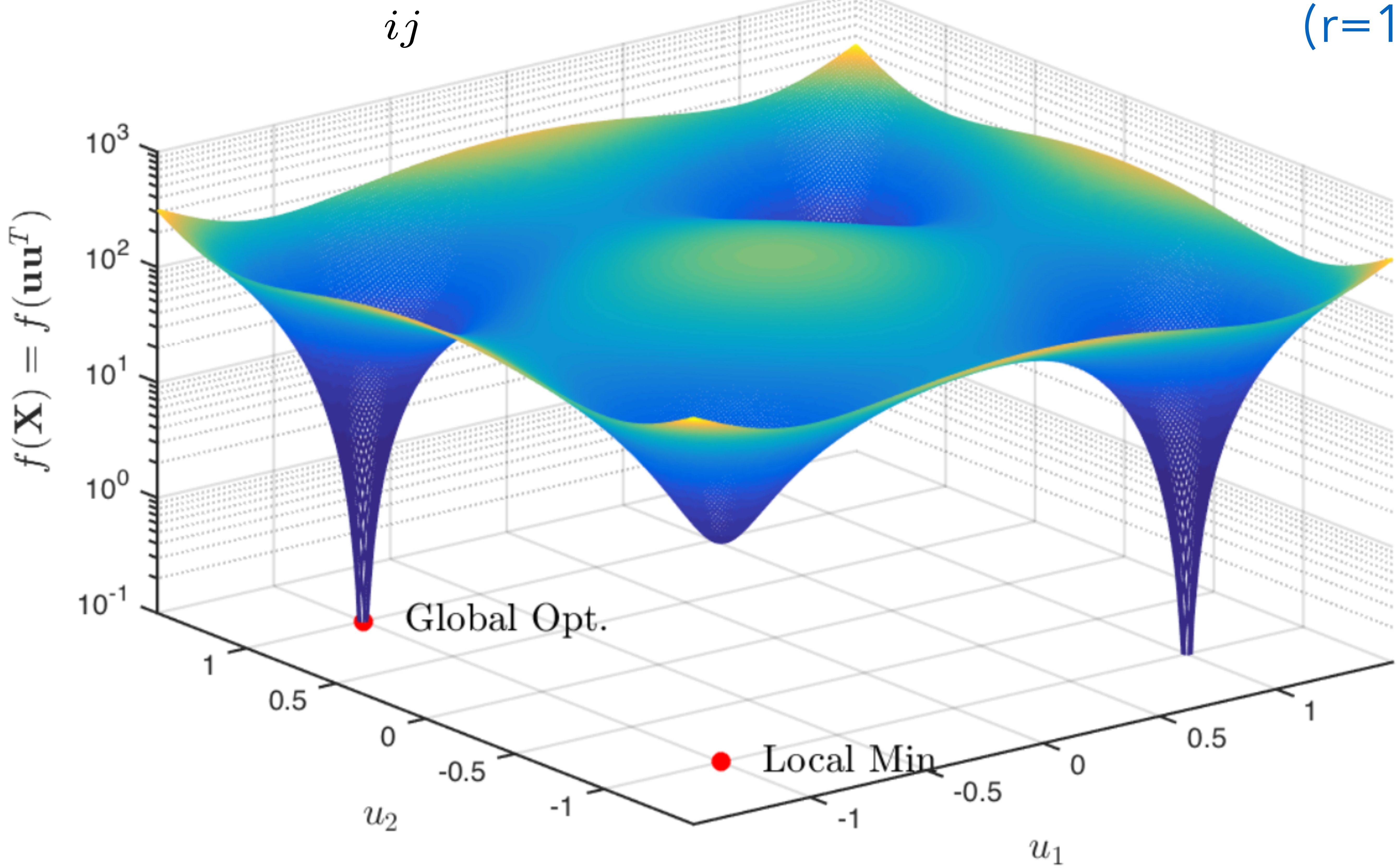
(r=1)

What about local minima?

- Factorization might also introduce local minima
- Example: **Weighted low-rank approximation**

$$f(uu^\top) = \sum_{ij} W_{ij} \cdot (X^\star_{ij} - u_i u_j)^2 \quad \text{where} \quad X^\star = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad W = \begin{bmatrix} 100 & 1 \\ 1 & 100 \end{bmatrix}$$

(r=1)

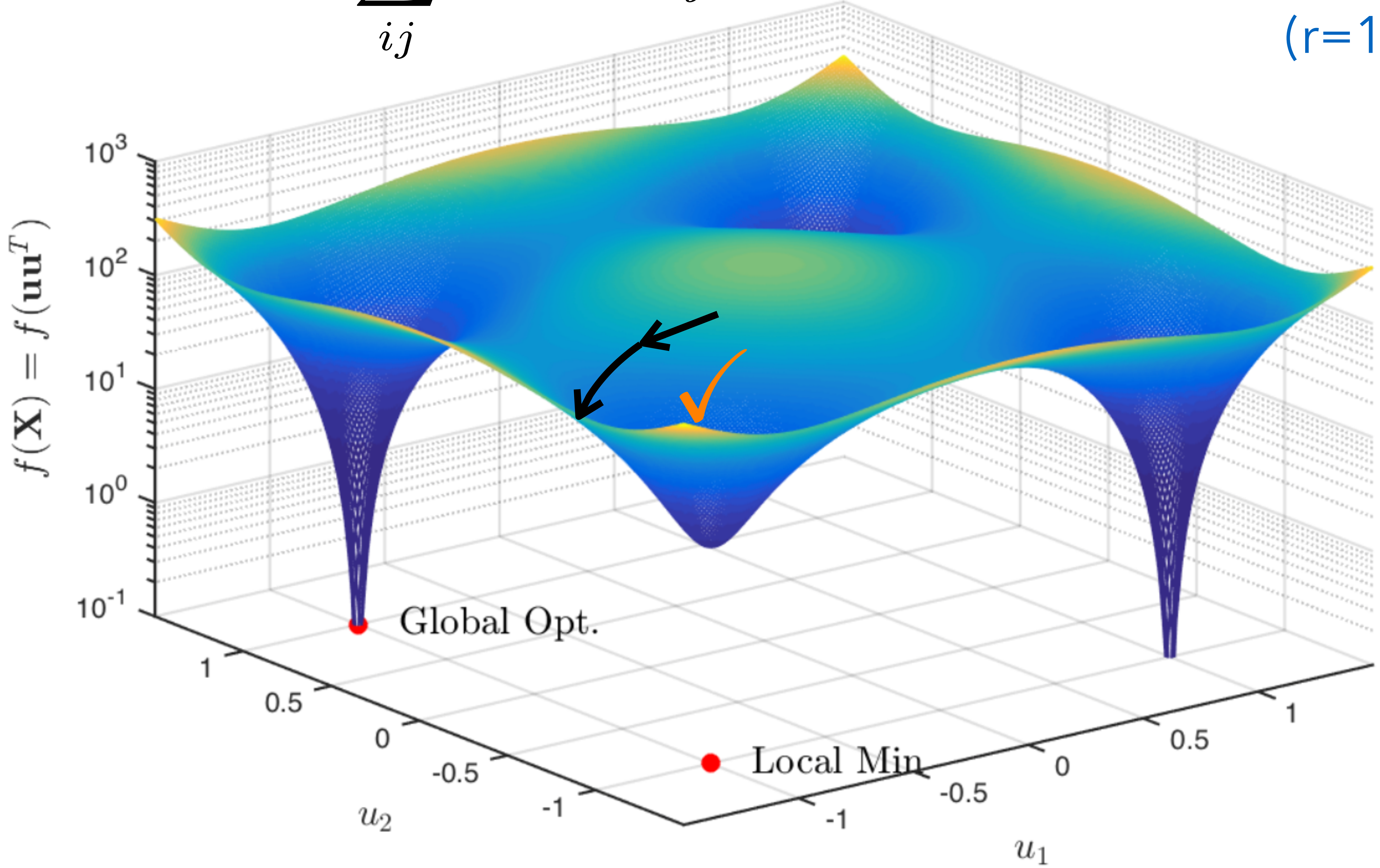


What about local minima?

- Factorization might also introduce local minima
- Example: **Weighted low-rank approximation**

$$f(uu^\top) = \sum_{ij} W_{ij} \cdot (X^\star_{ij} - u_i u_j)^2 \quad \text{where} \quad X^\star = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad W = \begin{bmatrix} 100 & 1 \\ 1 & 100 \end{bmatrix}$$

(r=1)

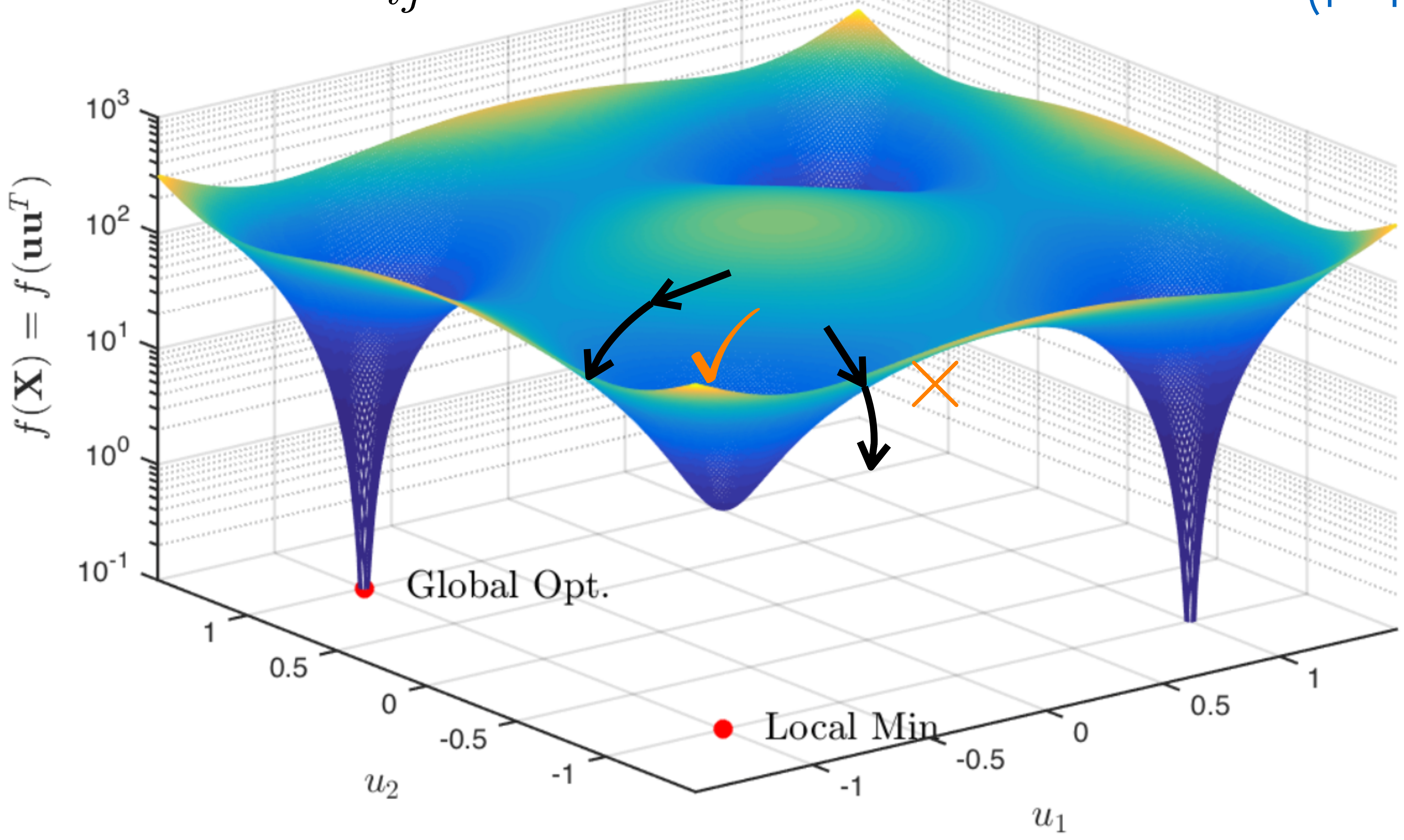


What about local minima?

- Factorization might also introduce local minima
- Example: **Weighted low-rank approximation**

$$f(uu^\top) = \sum_{ij} W_{ij} \cdot (X^\star_{ij} - u_i u_j)^2 \quad \text{where} \quad X^\star = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad W = \begin{bmatrix} 100 & 1 \\ 1 & 100 \end{bmatrix}$$

(r=1)

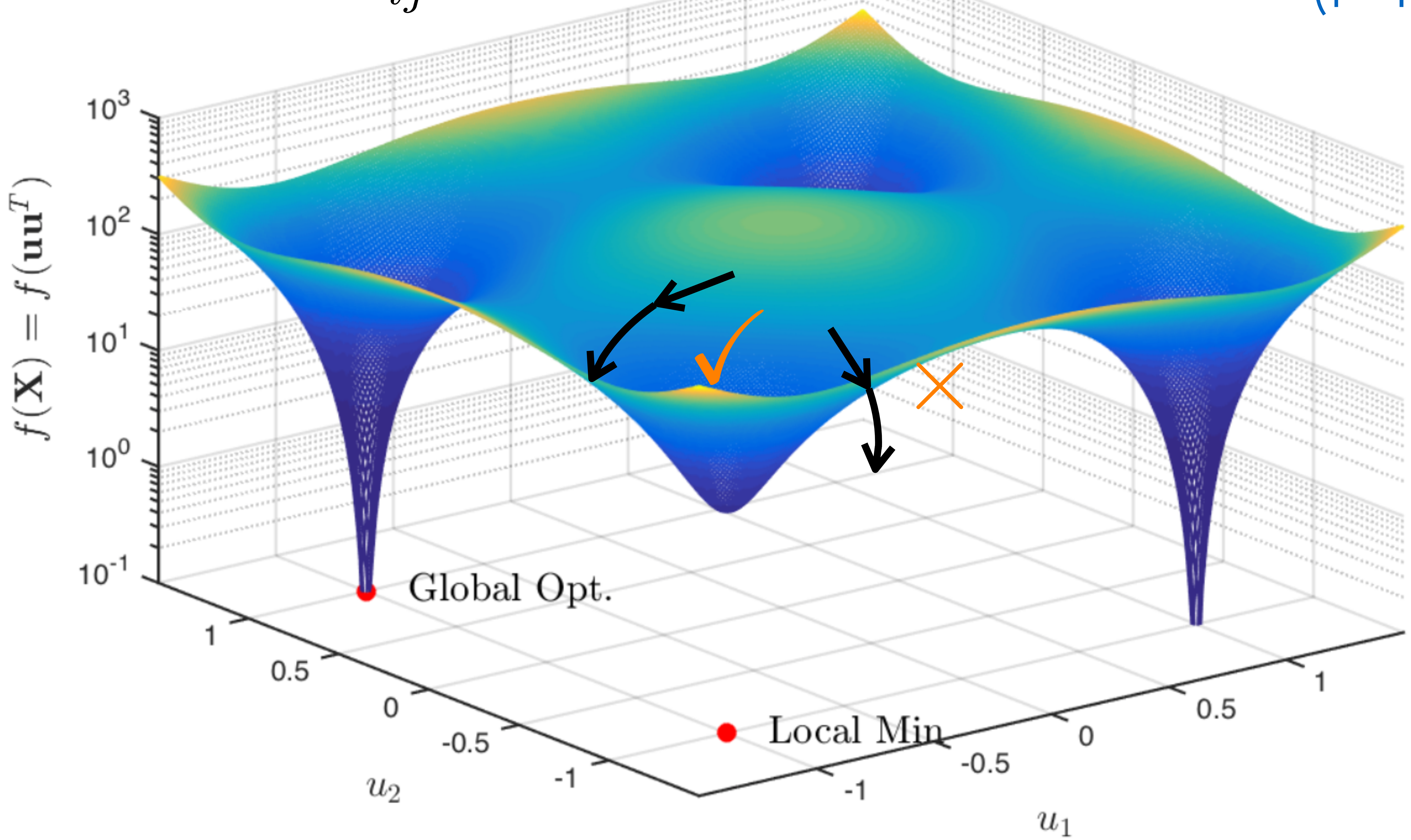


What about local minima?

- Factorization might also introduce local minima
- Example: **Weighted low-rank approximation**

$$f(uu^\top) = \sum_{ij} W_{ij} \cdot (X^\star_{ij} - u_i u_j)^2 \quad \text{where} \quad X^\star = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad W = \begin{bmatrix} 100 & 1 \\ 1 & 100 \end{bmatrix}$$

(r=1)



- Even simple objectives can be hard to handle
- Proper initialization is key

Nevertheless, can we hope for some guarantees?

norm: abuse of notation to indicate a general class of distance functions

- General recipe

$$
\begin{aligned}
\|x_{t+1} - x^\star\|_\sharp^2 &= \|x_t - \eta \nabla f(x_t) - x^\star\|_\sharp^2 \\
&= \|x_t - x^\star\|_\sharp^2 - 2\eta \langle \nabla f(x_t), x_t - x^\star \rangle + \eta^2 \|\nabla f(x_t)\|_\sharp^2
\end{aligned}
$$

Nevertheless, can we hope for some guarantees?

norm: abuse of notation to indicate a general class of distance functions

- General recipe

$$\|x_{t+1} - x^\star\|_\sharp^2 = \|x_t - \eta \nabla f(x_t) - x^\star\|_\sharp^2$$
$$= \underbrace{\|x_t - x^\star\|_\sharp^2}_{} - 2\eta \langle \nabla f(x_t), x_t - x^\star \rangle + \eta^2 \|\nabla f(x_t)\|_\sharp^2$$

(This term dictates the distance from previous iteration)

Nevertheless, can we hope for some guarantees?

norm: abuse of notation to indicate a general class of distance functions

- General recipe

$$\|x_{t+1} - x^\star\|_\sharp^2 = \|x_t - \eta\nabla f(x_t) - x^\star\|_\sharp^2$$
$$= \underbrace{\|x_t - x^\star\|_\sharp^2}_{} - 2\eta \underbrace{\langle \nabla f(x_t), x_t - x^\star \rangle}_{} + \eta^2 \|\nabla f(x_t)\|_\sharp^2$$

(This term dictates the distance from previous iteration)

(If we can bound this term to cancel this term)

Nevertheless, can we hope for some guarantees?

norm: abuse of notation to indicate a general class of distance functions

- General recipe

$$\|x_{t+1} - x^\star\|_\sharp^2 = \|x_t - \eta\nabla f(x_t) - x^\star\|_\sharp^2$$
$$= \underbrace{\|x_t - x^\star\|_\sharp^2}_{\text{(This term dictates the distance from previous iteration)}} - 2\eta \underbrace{\langle \nabla f(x_t), x_t - x^\star \rangle}_{\text{(If we can bound this term to cancel this term)}} + \eta^2 \|\nabla f(x_t)\|_\sharp^2$$

(This term dictates the distance from previous iteration)

(If we can bound this term to cancel this term)

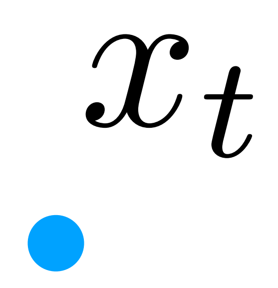
- Where can we actively intervene? **By choosing appropriate step size!**

Nevertheless, can we hope for some guarantees?

- What is the geometric intuition of $\langle \nabla f(x_t), x_t - x^\star \rangle$?

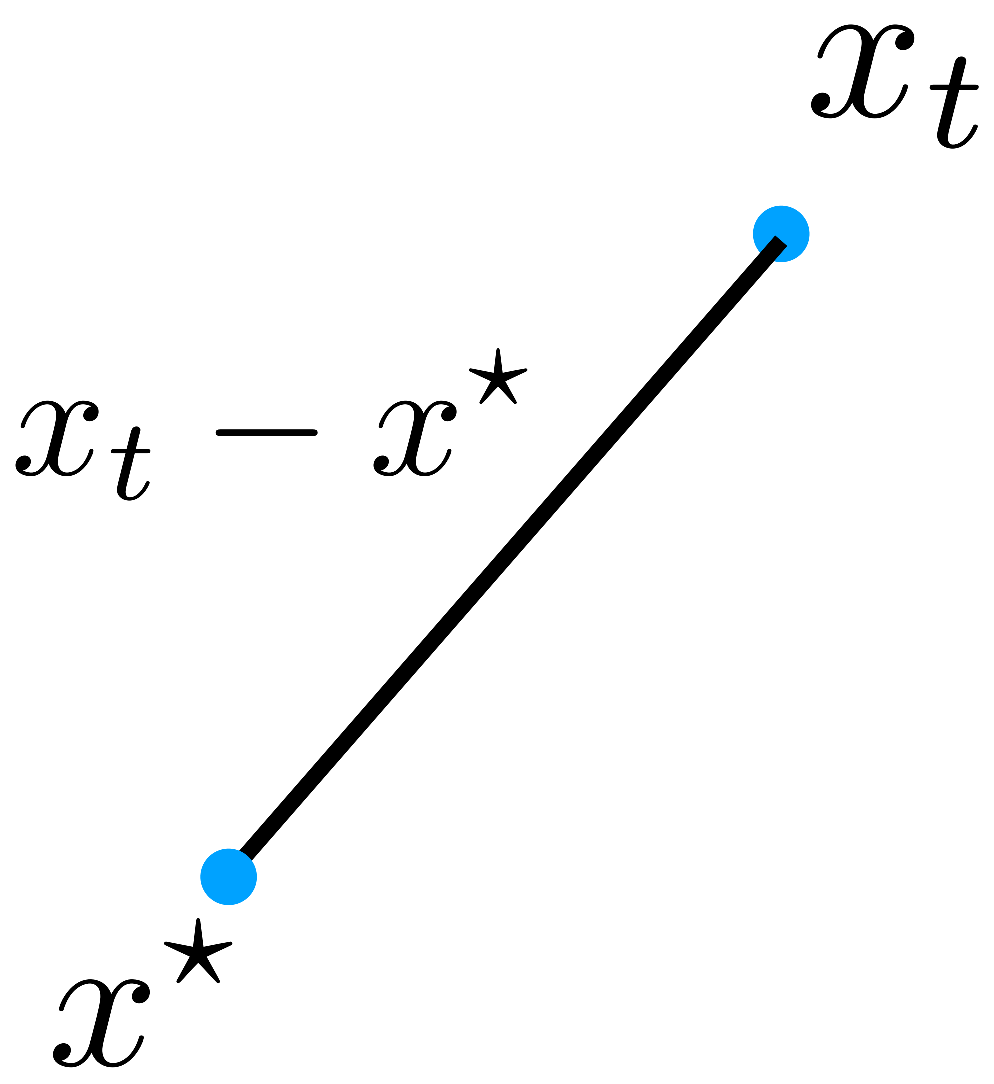
Nevertheless, can we hope for some guarantees?

- What is the geometric intuition of $\langle \nabla f(x_t), x_t - x^\star \rangle$?



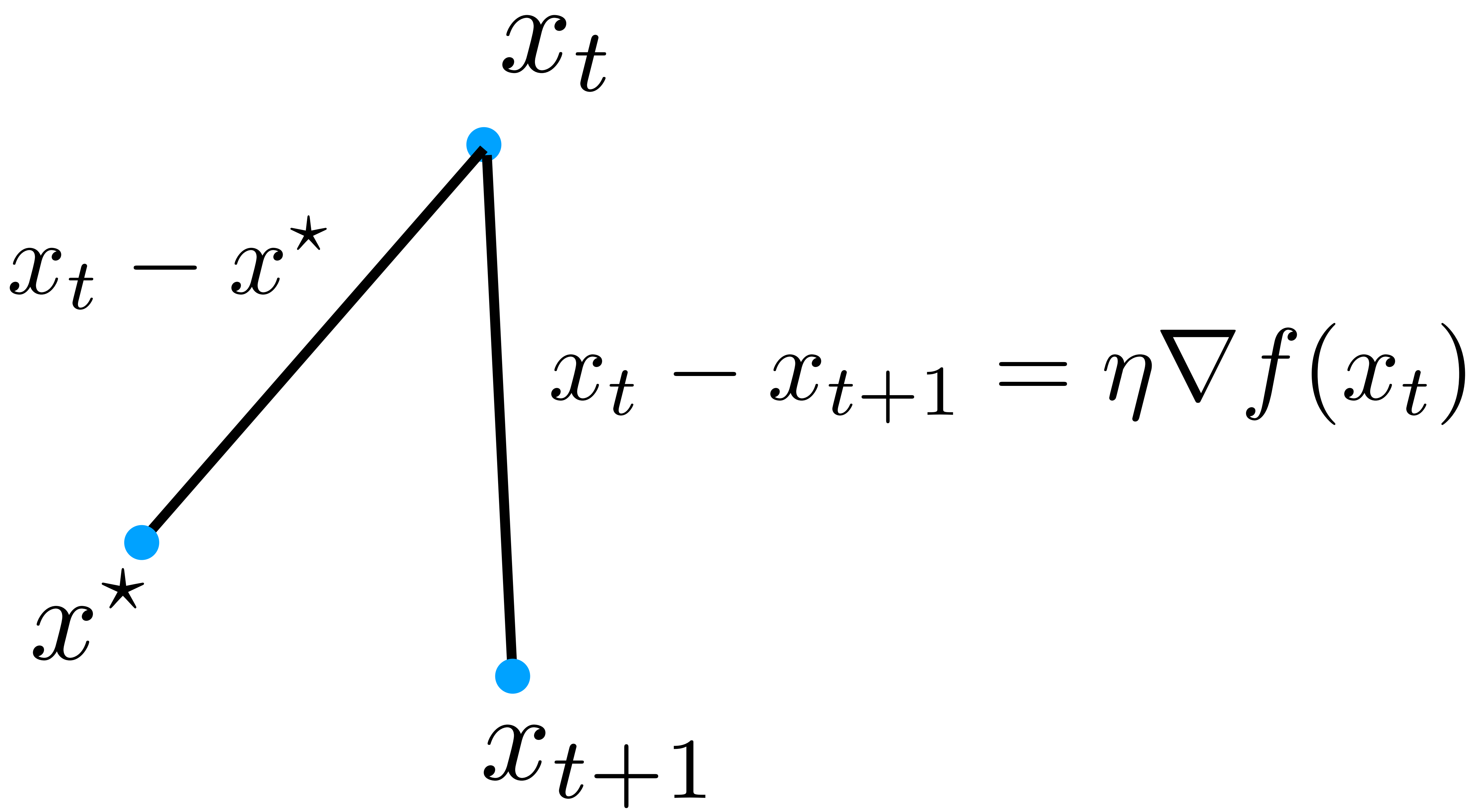
Nevertheless, can we hope for some guarantees?

– What is the geometric intuition of $\langle \nabla f(x_t), x_t - x^\star \rangle$?



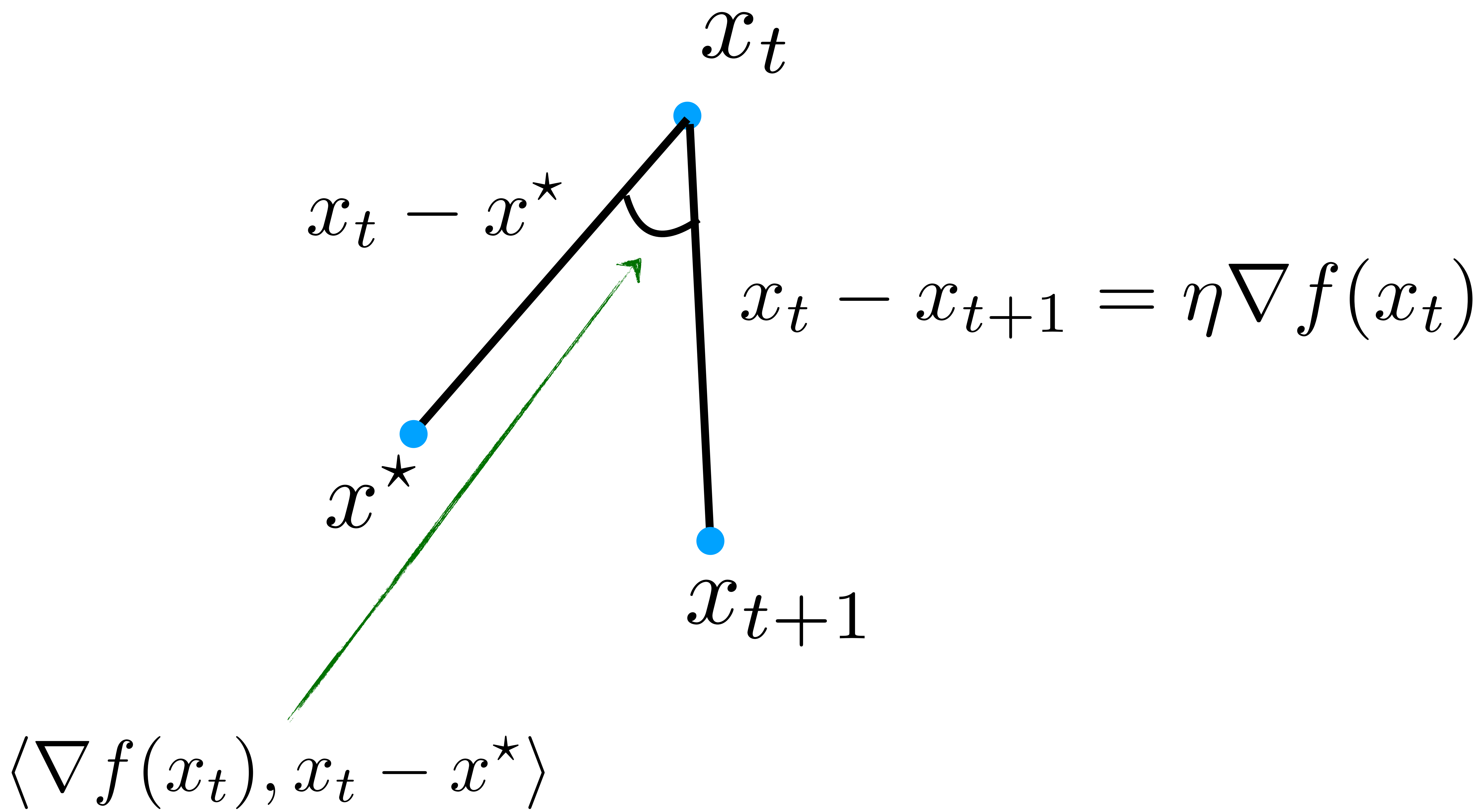
Nevertheless, can we hope for some guarantees?

– What is the geometric intuition of $\langle \nabla f(x_t), x_t - x^\star \rangle$?



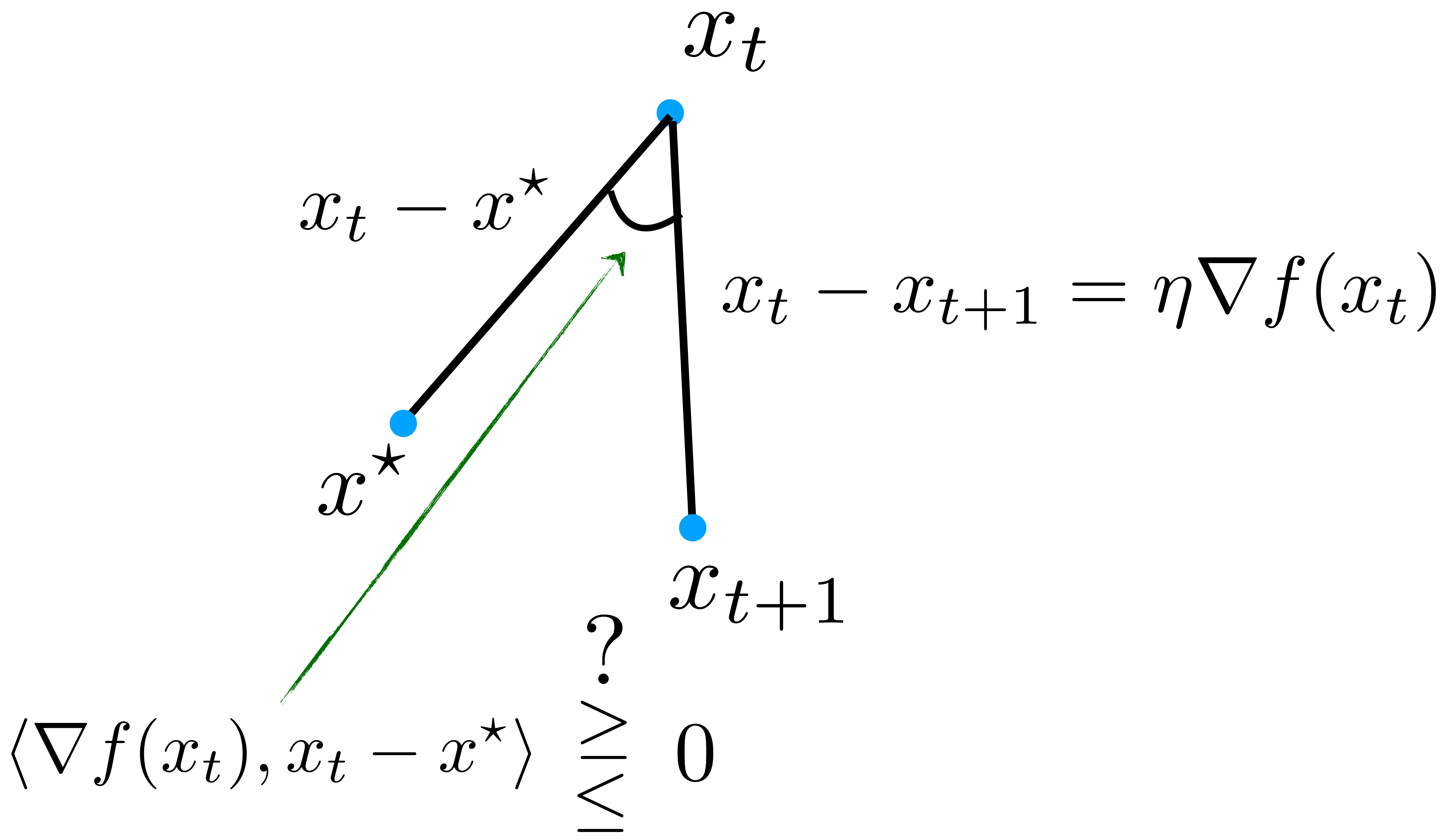
Nevertheless, can we hope for some guarantees?

– What is the geometric intuition of $\langle \nabla f(x_t), x_t - x^\star \rangle$?



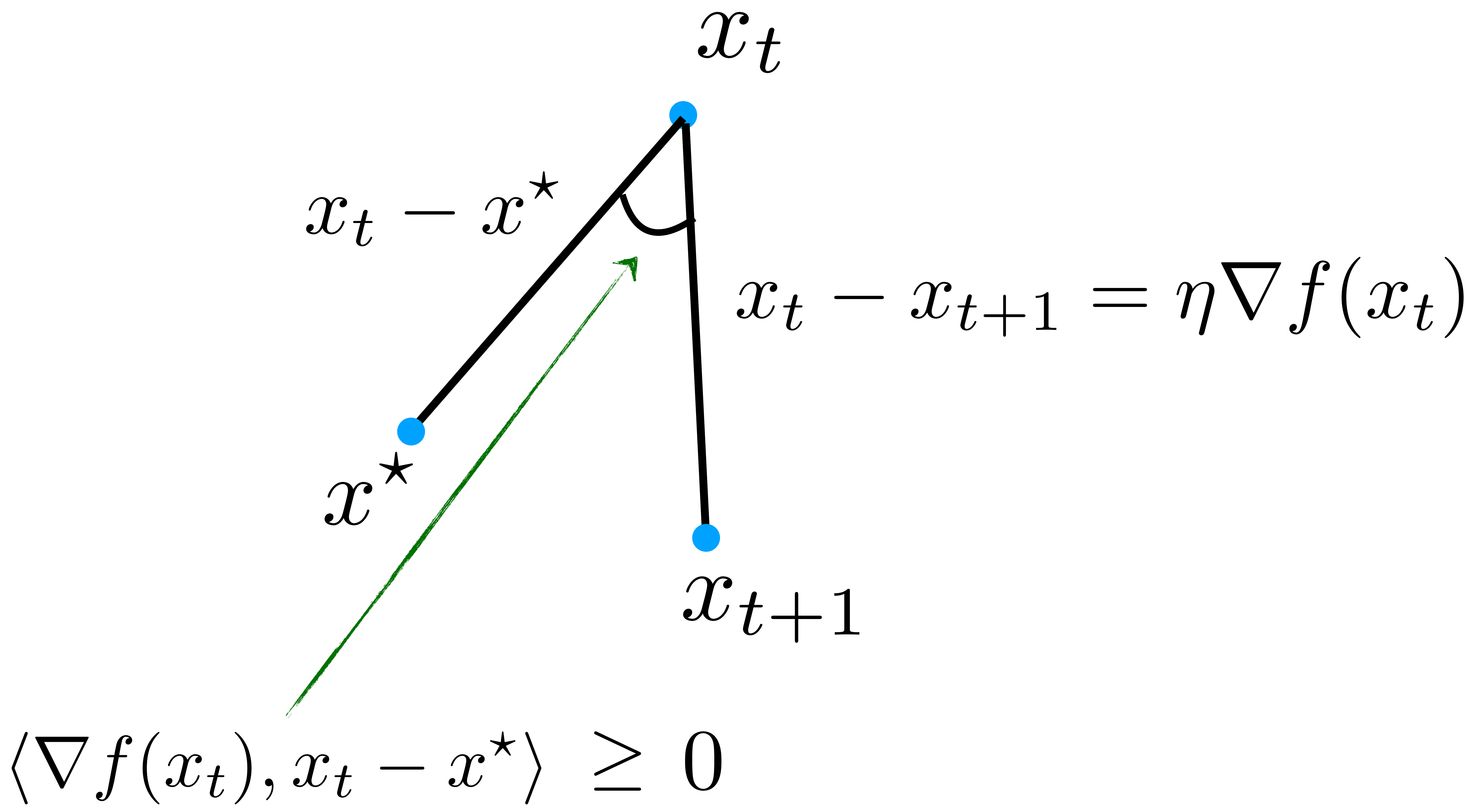
Nevertheless, can we hope for some guarantees?

- What is the geometric intuition of $\langle \nabla f(x_t), x_t - x^\star \rangle$?



Nevertheless, can we hope for some guarantees?

- What is the geometric intuition of $\langle \nabla f(x_t), x_t - x^\star \rangle$?



Regulatory condition

– Reminder:

$$\begin{aligned}\|x_{t+1} - x^\star\|_\sharp^2 &= \|x_t - \eta \nabla f(x_t) - x^\star\|_\sharp^2 \\ &= \|x_t - x^\star\|_\sharp^2 - 2\eta \left\langle \nabla f(x_t), x_t - x^\star \right\rangle + \eta^2 \|\nabla f(x_t)\|_\sharp^2\end{aligned}$$

Regulatory condition

- Reminder:

$$\begin{aligned}\|x_{t+1} - x^\star\|_\sharp^2 &= \|x_t - \eta \nabla f(x_t) - x^\star\|_\sharp^2 \\ &= \|x_t - x^\star\|_\sharp^2 - 2\eta \langle \nabla f(x_t), x_t - x^\star \rangle + \eta^2 \|\nabla f(x_t)\|_\sharp^2\end{aligned}$$

- We would like:

$$\langle \nabla f(x_t), x_t - x^\star \rangle \geq \alpha \|x_t - x^\star\|_\sharp^2 + \beta \|\nabla f(x_t)\|_\sharp^2$$

Regulatory condition

- Reminder:

$$\|x_{t+1} - x^\star\|_\sharp^2 = \|x_t - \eta \nabla f(x_t) - x^\star\|_\sharp^2$$
$$= \|x_t - x^\star\|_\sharp^2 - 2\eta \langle \nabla f(x_t), x_t - x^\star \rangle + \eta^2 \|\nabla f(x_t)\|_\sharp^2$$

- We would like:

$$\langle \nabla f(x_t), x_t - x^\star \rangle \geq \alpha \|x_t - x^\star\|_\sharp^2 + \beta \|\nabla f(x_t)\|_\sharp^2$$

for sufficient $\alpha, \beta \geq 0$ such that

$$\|x_t - x^\star\|_\sharp^2 - 2\eta \langle \nabla f(x_t), x_t - x^\star \rangle + \eta^2 \|\nabla f(x_t)\|_\sharp^2$$
$$\leq \|x_t - x^\star\|_\sharp^2 - c\alpha\eta \|x_t - x^\star\|_\sharp^2 - \left(c\eta\beta - \eta^2\right) \|\nabla f(x_t)\|_\sharp^2$$

c is problem dependent

Why should we hope for such a condition to hold?

Why should we hope for such a condition to hold?

- We know from convex analysis that

  "For smooth and strongly convex functions:" $\forall x, y$

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \frac{\mu L}{\mu + L} \|x - y\|_2^2 + \frac{1}{\mu + L} \|\nabla f(x) - \nabla f(y)\|_2^2$$

Why should we hope for such a condition to hold?

- We know from convex analysis that

  "For smooth and strongly convex functions:" $\forall x, y$

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \frac{\mu L}{\mu + L} \|x - y\|_2^2 + \frac{1}{\mu + L} \|\nabla f(x) - \nabla f(y)\|_2^2$$

- Set $y = x^\star$ and since $\nabla f(x^\star) = 0$

$$\langle \nabla f(x), x - x^\star \rangle \geq \frac{\mu L}{\mu + L} \|x - x^\star\|_2^2 + \frac{1}{\mu + L} \|\nabla f(x)\|_2^2$$

  and compare with

$$\langle \nabla f(x_t), x_t - x^\star \rangle \geq \alpha \|x_t - x^\star\|_\sharp^2 + \beta \|\nabla f(x_t)\|_\sharp^2$$

Local convergence guarantees for $UU^\top$

(The UV' case is left for you to study and explore)

– Define **distance function**:

$$\textsc{Dist}(U,\ U^\star R) := \min_R \|U - U^\star R\|_F$$

(This is the $\|\cdot\|_\sharp$ distance for this problem)

Local convergence guarantees for $UU^\top$

(The UV' case is left for you to study and explore)

- Define **distance function**:

$$\textsc{Dist}(U,\ U^\star R) := \min_R \|U - U^\star R\|_F$$

(This is the $\|\cdot\|_\sharp$ distance for this problem)

- **Local convergence**: we assume we start from a sufficiently good initial point

Whiteboard

Main result: Local convergence guarantees

- f is convex and differentiable

$$U_{i+1} = U_i - \eta \nabla f(U_i {V_i}^\top) \cdot {V_i}^\top$$

$$V_{i+1} = V_i - \eta \nabla f(U_i {V_i}^\top)^\top \cdot U_i$$

THEOREM: LOCAL CONVERGENCE

If f is a "nice" function and (U_i, V_i) are ***sufficiently*** close to $(U^\star, V^\star)$, then **non-convex** alternating gradient descent **i)** converges to $(U^\star, V^\star)$, and **ii)** achieves the same convergence guarantees with convex optimization:

Main result: Local convergence guarantees

- f is convex and differentiable

$$U_{i+1} = U_i - \eta \nabla f(U_i V_i^\top) \cdot V_i^\top$$

$$V_{i+1} = V_i - \eta \nabla f(U_i V_i^\top)^\top \cdot U_i$$

THEOREM: LOCAL CONVERGENCE

If f is a "nice" function and (U_i, V_i) are ***sufficiently*** close to $(U^\star, V^\star)$, then **non-convex** alternating gradient descent **i)** converges to $(U^\star, V^\star)$, and **ii)** achieves the same convergence guarantees with convex optimization:

i.e., in $O\left(1/\varepsilon\right)$ or $O\left(\log 1/\varepsilon\right)$ iter., we have $f(\widehat{U}\widehat{V}^\top) - f(U^\star V^{\star\top}) \leq \varepsilon$

(just smooth) *(strongly convex)*

Main result: Local convergence guarantees

- f is convex and differentiable

$$U_{i+1} = U_i - \eta \nabla f(U_i V_i^\top) \cdot V_i^\top$$

$$V_{i+1} = V_i - \eta \nabla f(U_i V_i^\top)^\top \cdot U_i$$

THEOREM: LOCAL CONVERGENCE

If f is a "nice" function and (U_i, V_i) are ***sufficiently*** close to $(U^\star, V^\star)$, then **non-convex** alternating gradient descent **i)** converges to $(U^\star, V^\star)$, and **ii)** achieves the same convergence guarantees with convex optimization:

i.e., in $O\left(1/\varepsilon\right)$ or $O\left(\log 1/\varepsilon\right)$ iter., we have $f(\widehat{U}\widehat{V}^\top) - f(U^\star V^{\star\top}) \leq \varepsilon$

(just smooth) *(strongly convex)*

Impact in practice: Theory…

- **…provides insights for step size selection, proper initialization,**
- **…covers cases where we do not know the rank parameter a priori,**
- **…provides statistical guarantees for specific f.**

Our proof strategy

Show how the algorithm behaves *locally*

i.e., if we are sufficiently close to the optimal point.

Our proof strategy

Show how the algorithm behaves *locally*

i.e., if we are sufficiently close to the optimal point.

↓

Provide proper initialization

i.e., how to gets close to points where we know our algorithm behaves well

Our proof strategy

Show how the algorithm behaves *locally*

i.e., if we are sufficiently close to the optimal point.

↓

Provide proper initialization

i.e., how to gets close to points where we know our algorithm behaves well

↓

Convergence to global minimum for non-convex optimization!

Main result: Proper initialization and global convergence

Main result: Proper initialization and global convergence

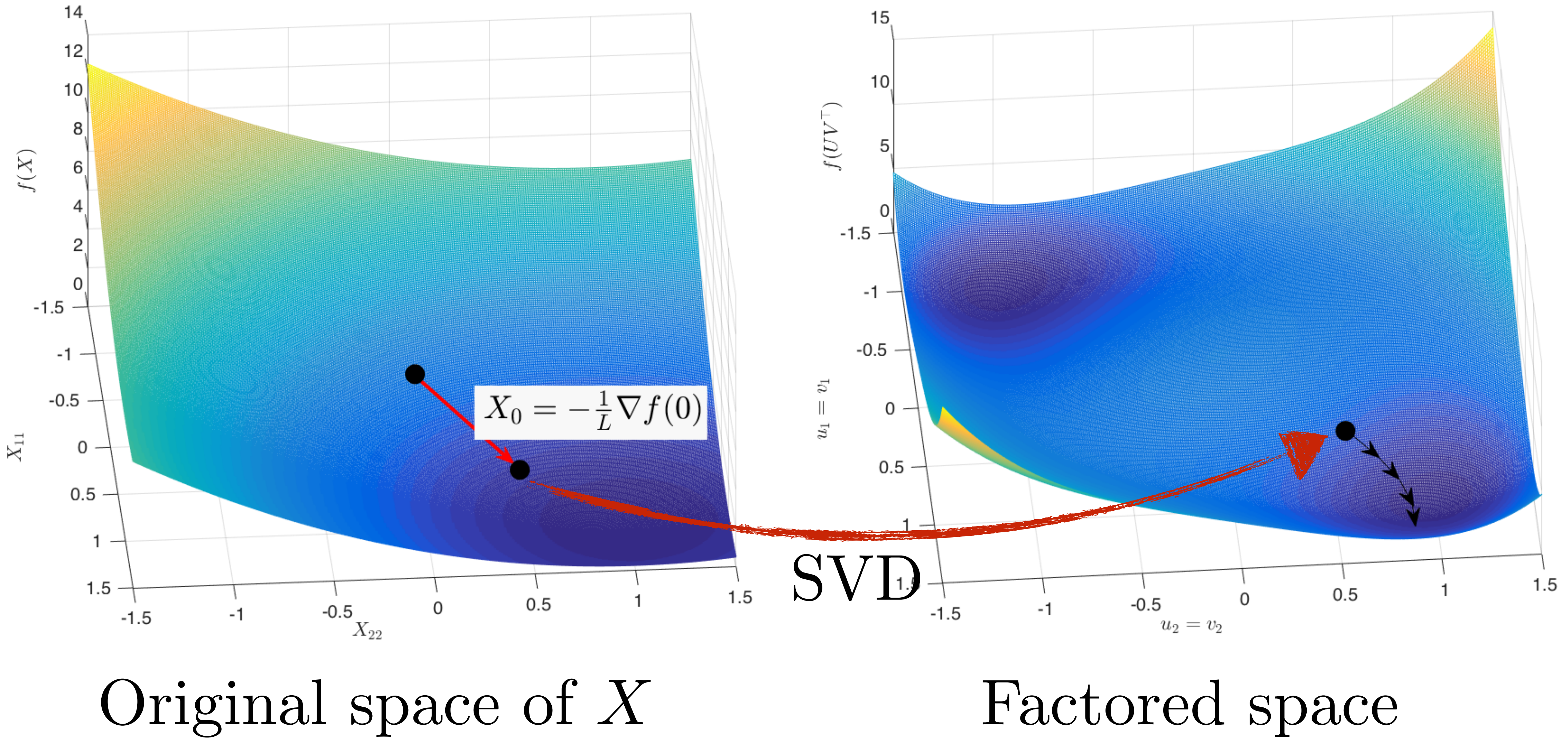
Goal: Initialize such that (U_0, V_0) is sufficiently close to $(U^\star, V^\star)$

Main result: Proper initialization and global convergence

Goal: Initialize such that (U_0, V_0) is sufficiently close to $(U^\star, V^\star)$

- **Proposed initialization:**
  - Compute $X_0 \propto -\nabla f(0)$
  - Perform one SVD calculation:

$$X_0 = U_0 V_0^\top$$

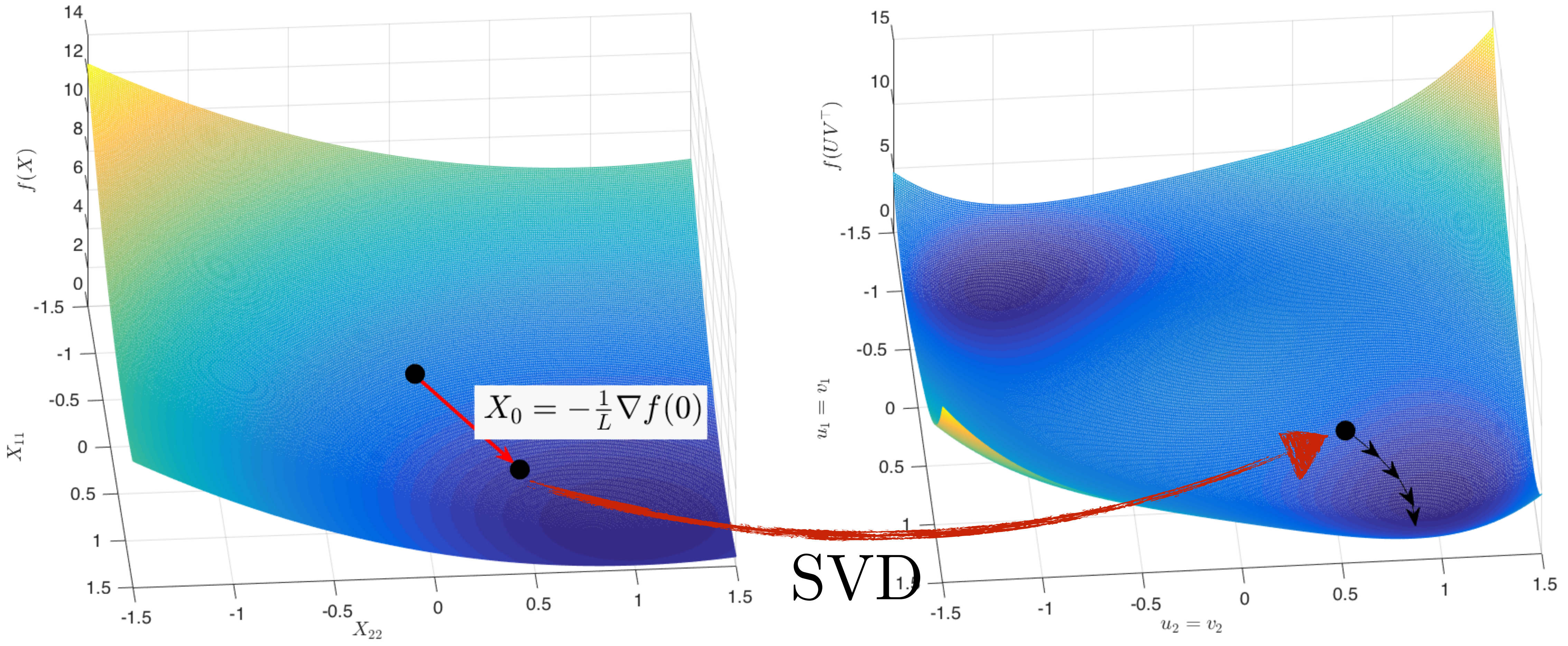


Main result: Proper initialization and global convergence

Goal: Initialize such that (U_0, V_0) is sufficiently close to $(U^\star, V^\star)$

- **Proposed initialization:**
  - Compute $X_0 \propto -\nabla f(0)$
  - Perform one SVD calculation:

$$X_0 = U_0 V_0^\top$$



Original space of X

Factored space

THEOREM: GLOBAL CONVERGENCE

If the function f is "well-conditioned" , then non-convex alternating gradient descent converges to the global optimum / optima.

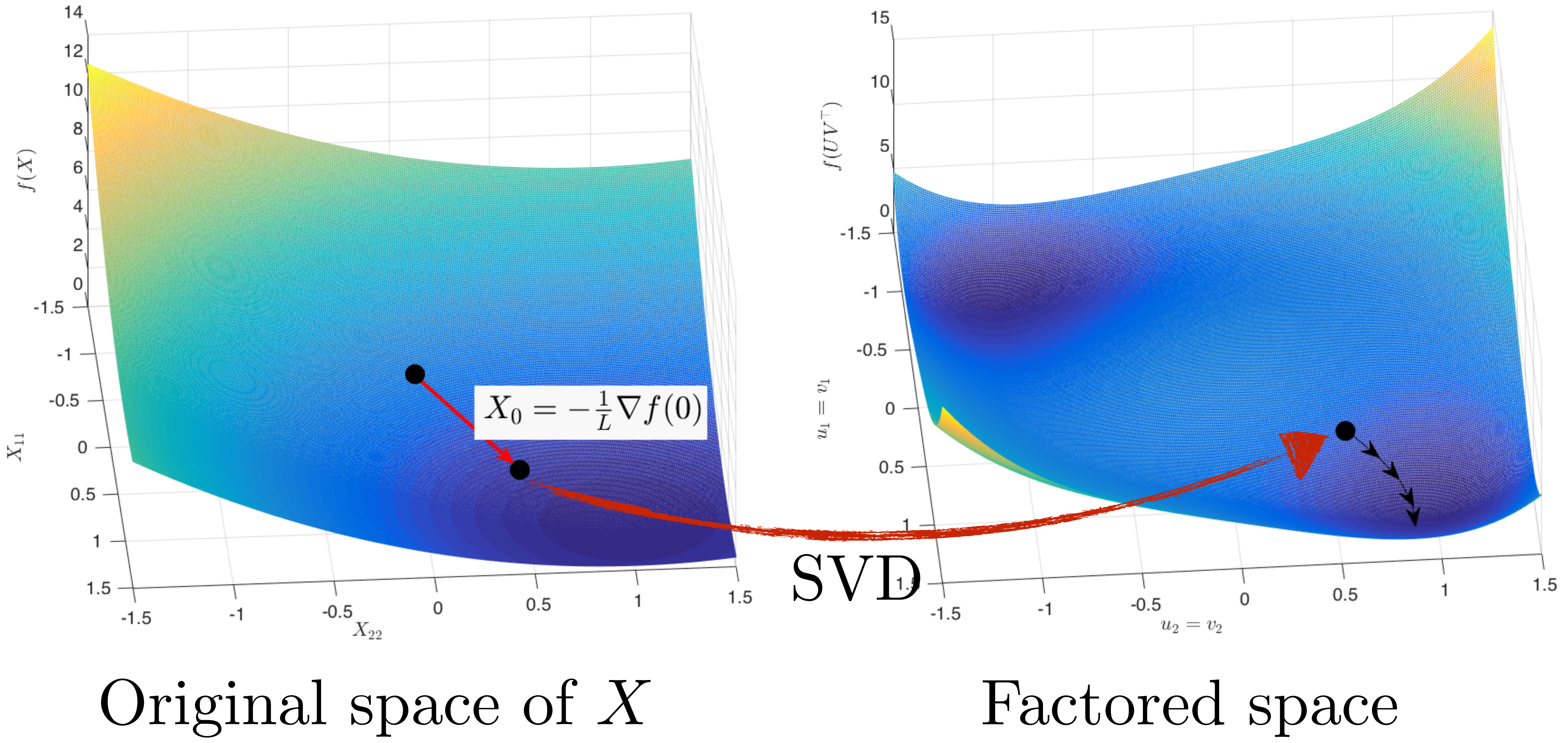
Condition number: ratio of smoothness over strong convexity parameters

Main result: Proper initialization and global convergence

Goal: Initialize such that (U_0, V_0) is sufficiently close to $(U^\star, V^\star)$

- **Proposed initialization:**
  - Compute $X_0 \propto -\nabla f(0)$
  - Perform one SVD calculation:

$$X_0 = U_0 V_0^\top$$



PRACTICAL IMPACT

One SVD *(non-convex)* **vs.** **SVD per iteration!** *(convex)*

Practical aspects of optimizing $\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$

.. by using $(U_{t+1}, V_{t+1}) = (U_t, V_t) - \eta(\nabla f(U_t V_t^\top) V_t, \nabla f(U_t V_t)^\top U_t)$

Practical aspects of optimizing $\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$

.. by using $(U_{t+1}, V_{t+1}) = (U_t, V_t) - \eta(\nabla f(U_t V_t^\top) V_t, \nabla f(U_t V_t)^\top U_t)$

– There are initializations that come with some convergence guarantees

$$(U_0, V_0) = \texttt{SVD}\left(-\nabla f(0_{n \times p})\right)$$

(Often called spectral method for initialization)

..the guarantees are weak, but often it works in practice!

Practical aspects of optimizing $\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$

.. by using $(U_{t+1}, V_{t+1}) = (U_t, V_t) - \eta(\nabla f(U_t V_t^\top) V_t, \nabla f(U_t V_t)^\top U_t)$

– There are initializations that come with some convergence guarantees

$$(U_0, V_0) = \texttt{SVD}\left(-\nabla f(0_{n \times p})\right)$$

(Often called spectral method for initialization)

..the guarantees are weak, but often it works in practice!

– What about random initialization?

Practical aspects of optimizing $\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$

.. by using $(U_{t+1}, V_{t+1}) = (U_t, V_t) - \eta(\nabla f(U_t V_t^\top) V_t, \nabla f(U_t V_t)^\top U_t)$

– There are initializations that come with some convergence guarantees

$$(U_0, V_0) = \texttt{SVD}\left(-\nabla f(0_{n \times p})\right)$$

(Often called spectral method for initialization)

..the guarantees are weak, but often it works in practice!

– What about random initialization?

– Constant step size vs. adaptive step size (Open question for specific f)

Practical aspects of optimizing $\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$

.. by using $(U_{t+1}, V_{t+1}) = (U_t, V_t) - \eta(\nabla f(U_t V_t^\top) V_t, \nabla f(U_t V_t)^\top U_t)$

– What if we don't know the exact rank? (Open question)

Practical aspects of optimizing $\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} f(UV^\top)$

.. by using $(U_{t+1}, V_{t+1}) = (U_t, V_t) - \eta(\nabla f(U_t V_t^\top) V_t, \nabla f(U_t V_t)^\top U_t)$

- What if we don't know the exact rank? (Open question)

Demo

Conclusion

- This lecture considers **low–rank model selection** in Data Science applications
- While there are rigorous and efficient methods also in the convex domain we followed the **non–convex path,** beyond hard thresholding methods
- We discussed some global convergence guarantees (under proper initialization assumptions) and discussed about some open questions

Next lecture

- We will focus on the landscape of non–convex functions, starting from simple cases (such as low–rankness), and moving towards more generic scenaria