Dropping convexity for faster semi-definite optimization

Srinadh Bhojanapalli, Anastasios Kyrillidis and Sujay Sanghavi

The University of Texas at Austin

[Semi-definite optimization]

We consider the following optimization problem:

(SDP) $$\min_{X \succeq 0} f(X)$$

Main assumptions:

- X is a symmetric, positive semi-definite in $\mathbb{R}^{n \times n}$.
- $f : \mathbb{S}^n_+ \to \mathbb{R}$ has Lipschitz continuous gradients:

$$\|\nabla f(X) - \nabla f(Y)\|_F \leq M \cdot \|X - Y\|_F, \quad X, Y \in \mathbb{S}^n_+$$

- f might also be (restricted) strongly convex:

$$f(Y) \geq f(X) + \langle \nabla f(X),\ Y - X \rangle + \tfrac{m}{2}\|Y - X\|_F^2, \quad X, Y \in \mathbb{S}^n_+$$

[Common practice: low rankness]

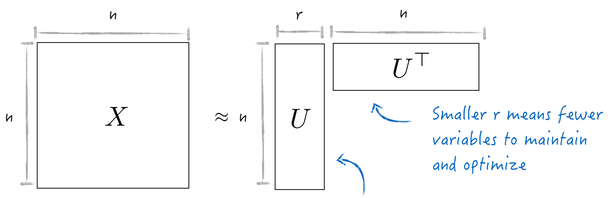
In modern machine learning tasks, one is interested in finding/ learning a **low rank** matrix (Occam's razor):

- Low-rankness models better the underlying task.
- Low-rankness results into better computational complexity (fewer variables to optimize).
- Low-rankness might prevent over-fitting in machine learning problems.

Examples: matrix completion, phase retrieval, quantum state tomography, etc.

[Low rankness + PSD constraint]

Any PSD matrix $X \in \mathbb{R}^{n \times n}$ of rank r can be written as $X = UU^\top$, *i.e.*,



(SDP) reduces to:

(Non-convex SDP) $$\min_{U \in \mathbb{R}^{n \times r}} g(U)$$
$$\text{where} \quad r \leq n$$

[Contributions]

- ***Factored Gradient Descent*** (FGD) algorithm: Non-convex!!!

$$U^+ = U - \eta \cdot \nabla f(UU^\top) \cdot U$$

- **Novel** step size selection η.
- (Local) convergence of FGD under common assumptions:
  - **Linear convergence** for smooth and strongly convex f.
  - **Sublinear** convergence for just smooth f.
- Global convergence after proper initialization.

[Factored gradient descent (FGD)]

[Algorithm: Factored gradient descent]

Input: function f, target rank r, # of iterations K.

- Compute X^0 (see below).
- Set $U \in \mathbb{R}^{n \times r}$ such that $X^0 = UU^\top$
- Set step size η.
- For $k = 0$ to $K - 1$
  - Compute $U^+ = U - \eta \cdot \nabla f(UU^\top) \cdot U$
  - Set $U = U^+$

Output: $X = UU^\top$

Step size selection

$$\eta = \frac{C}{M\|X^0\|_2 + \|\nabla f(X^0)\|_2}$$

for some constant $C > 0$.

Intuition for η selection.

- Let f be a separable function:

$$f(X) = \sum_{ij} f_{ij}(X_{ij})$$

- For $r = 1$, we have $X = uu^\top$.
- Define $f(uu^\top) \equiv g(u)$. Then:

$$\nabla g(u) = \nabla f(uu^\top) \cdot u$$

$$\nabla^2 g(u) = \mathrm{mat}(\mathrm{diag}(\nabla^2 f(uu^\top)) \cdot \mathrm{vec}(uu^\top) + \nabla f(uu^\top)$$

- Convex optimization suggests:

$$\eta < \frac{1}{\|\nabla^2 g(\cdot)\|_2} \propto \frac{1}{M\|X\|_2 + \|\nabla f(X)\|_2}$$

Initialization

- **Option #1**: run standard convex algorithms for a few iterations.

Let $X^+ = \mathcal{P}_+(X - \frac{1}{M}\nabla f(X))$ be the proj. gradient descent update. Then, $\|X^+ - X\|_F \leq \frac{c}{\kappa\sqrt{r}\tau(X_r)} \cdot \sigma_r(X)$ implies:

$$\textsc{Dist}(U_r, U_r^\star) \leq \frac{c'}{\tau(X_r^\star)}\sigma_r(U_r^\star), \quad \textit{for constants } c,\ c' > 0.$$

- **Option #2**: Set $X^0 := \frac{1}{\|\nabla f(0) - \nabla f(e_1 e_1^\top)\|_F}\mathcal{P}_+(-\nabla f(0))$. Then,

$$\textsc{Dist}(U_r^0, U_r^\star) \leq 4\sqrt{2}r\tau(X_r^\star) \cdot \sqrt{\kappa^2 - 2/\kappa + 1} \cdot \sigma_r(U_r^\star).$$

[Theoretical guarantees]

- Define the following *distance metric*:

$$\textsc{Dist}(U, V) := \min_{R : R \in \mathcal{O}} \|U - VR\|_F.$$

between matrices $U, V \in \mathbb{R}^{n \times r}$. Here, R is restricted such that $R^\top R = I_{r \times r}$.

FGD Assumptions #1

- FGD is initialized with a "good" starting point $X^0 = U^0(U^0)^\top$ such that:

$$\textsc{Dist}(U^0, U_r^\star) \leq \rho\sigma_r(U_r^\star) \quad \text{for} \quad \rho := \frac{1}{100}\frac{\sigma_r(X^\star)}{\sigma_1(X^\star)}$$
$$\textsc{Dist}(U^0, U_r^\star) \leq \rho'\sigma_r(U_r^\star) \quad \text{for} \quad \rho' := \frac{1}{100\kappa}\frac{\sigma_r(X^\star)}{\sigma_1(X^\star)}$$

smooth

strongly convex

$\frac{1}{k}$-convergence rate for **smooth** f

After k iterations, FGD algorithm finds a solution X^k such that:

$$f(X^k) - f(X_r^\star) \leq \frac{\frac{6}{\eta^\star} \cdot \textsc{Dist}(U^0, U_r^\star)^2}{k + \frac{6}{\eta^\star} \cdot \frac{\textsc{Dist}(U^0, U_r^\star)^2}{f(X^0) - f(X_r^\star)}}.$$

Here, we assume $X^\star = X_r^\star$ and define:

$$\eta^\star = \frac{\frac{1}{16}}{M\|X^\star\|_2 + \|\nabla f(X^\star)\|_2}$$

Linear rate under **strong convexity**

Per iteration, FGD algorithm satisfies the following iteration invariant:

$$\textsc{Dist}(U^+, U^\star)^2 \leq \alpha \cdot \textsc{Dist}(U, U^\star)^2.$$

where

$$\alpha = 1 - \frac{m\sigma_r(X^\star)}{64(M\|X^\star\|_2 + \|\nabla f(X^\star)\|_2)}$$

Furthermore:

$$\textsc{Dist}(U^+, U_r^\star) \leq \rho'\sigma_r(U_r^\star).$$

- For cases where $\mathrm{rank}(X^\star) > r$

FGD Assumptions #2

- Assuming

$$\|X^\star - X_r^\star\|_F \leq \frac{1}{100}\frac{\sigma_r(X^\star)}{\sigma_1(X^\star)}\sigma_r(X^\star)$$
$$\|X^\star - X_r^\star\|_F \leq \frac{1}{200\kappa^{1.5}}\frac{\sigma_r(X^\star)}{\sigma_1(X^\star)}\sigma_r(X^\star)$$

smooth

strongly convex

we further have

$$\textsc{Dist}(U^+, U_r^\star)^2 \leq \alpha \cdot \textsc{Dist}(U, U_r^\star)^2 + \beta \cdot \|X^\star - X_r^\star\|_F^2$$

for some (controllable) β.

[Related work]

Reference	Conv. rate	Initialization	Output rank
[Haz08]	$1/\epsilon$ (Smooth f)	$X^0 = 0$	$1/\epsilon$
[Lau12]	$1/\epsilon$ (Smooth f)	$X^0 = 0$	$1/\epsilon$
[CW15]	$1/\epsilon$ (Local Asm.)	Application dependent	r
[CW15]	$\log(1/\epsilon)$ (Local Asm.)	Application dependent	r
This work	$1/\epsilon$ (Smooth f)	SVD / top-r	r
This work	$\log(1/\epsilon)$ (Smooth, RSC f)	SVD / top-r	r

…and for the affine rank minimization problem:

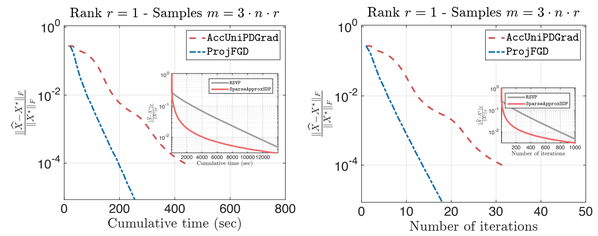
Reference	$\textsc{Dist}(U^+, U^\star)^2 \leq \alpha \cdot \textsc{Dist}(U, U^\star)^2$	$\textsc{Dist}(U^0, U^\star) \leq \cdots$
[JNS13]	$\alpha = \frac{1}{16}$ †	$\sqrt{6\delta} \cdot \frac{\sigma_r(X^\star)}{\sigma_1(X^\star)}\sigma_r(U_r^\star)$
[TBSR15]	$\alpha = 1 - \frac{\mathcal{C}_1}{\tau(U_r^\star)^4}$	$\frac{1}{4}\sigma_r(U_r^\star)$
[ZL15]	$\alpha = 1 - \frac{\mathcal{C}_2}{\tau(U_r^\star)\cdot r}$	$\sqrt{\frac{3}{16}} \cdot \sigma_r(U_r^\star)$
[CW15]	$\alpha = 1 - \frac{\mathcal{C}_3}{\tau(U_r^\star)^{10}}$	$(1 - \tau) \cdot \sigma_r(U_r^\star)$
This work	$\alpha = 1 - \frac{\mathcal{C}_4}{\tau(U_r^\star)^2}$	$\frac{1}{100} \cdot \frac{\sigma_r(X^\star)}{\sigma_1(X^\star)}\sigma_r(U_r^\star)$

[In practice…]

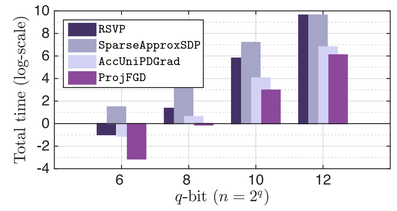
- **Quantum state tomography:**

$$\begin{array}{ll}\underset{X \in \mathbb{R}^{n \times n}}{\text{minimize}} & f(X) \\ \text{subject to} & \mathrm{rank}(X) = r \\ & X \succeq 0, \textsc{Tr}(X) \leq 1\end{array} \quad \rightarrow \quad \begin{array}{ll}\underset{U \in \mathbb{R}^{n \times r}}{\text{minimize}} & f(UU^\top) \\ \text{subject to} & \|U\|_F^2 \leq 1.\end{array}$$

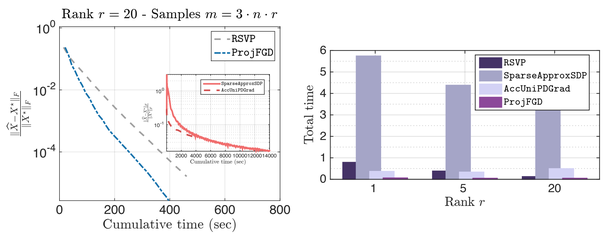
- **Pure** state density matrix recovery (r = 1)



- Faster convergence to a vicinity around $X^\star$:



- **Almost pure** state density matrix recovery (r > 1; here, r = {5, 20})



[References]

- Srinadh Bhojanapalli, Anastasios Kyrillidis and, Sujay Sanghavi, "*Dropping convexity for faster semi-definite optimization*", Arxiv preprint, submitted to STOC.