# Provable Acceleration of Nesterov's Accelerated Gradient for Rectangular Matrix Factorization and Linear Neural Networks

Zhenghao Xu [1] Yuqing Wang [2] Tuo Zhao [1]
Rachel Ward [3] Molei Tao [1]

[1]Georgia Tech [2]UC Berkeley [3]UT Austin

NeurIPS, 2024

# Convergence in classical smooth convex optimization

$$f^\star := f(x^\star) = \min_{x \in \mathbb{R}^d} f(x) > -\infty$$

First-order methods: only use gradient information $\nabla f(x)$.

- Gradient Descent (GD), Nesterov's Accelerated Gradient (NAG), etc.
- Widely used in machine learning.

# Convergence in classical smooth convex optimization

$$f^\star := f(x^\star) = \min_{x \in \mathbb{R}^d} f(x) > -\infty$$

First-order methods: only use gradient information $\nabla f(x)$.

- Gradient Descent (GD), Nesterov's Accelerated Gradient (NAG), etc.
- Widely used in machine learning.

Global convergence theory:

- For $L$-smooth $\mu$-(quasi) strongly convex function, GD converges in $O(\frac{L}{\mu} \log \frac{1}{\epsilon})$ iterations.
- $L$-smooth: $\|\nabla \ell(x) - \nabla \ell(y)\| \leq L\|x - y\|$.
- $\mu$-quasi strongly convex: $f^\star \geq f(x) + \langle \nabla f(x), x^\star - x \rangle + \frac{\mu}{2}\|x - x^\star\|^2$.

# Convergence in classical smooth convex optimization

$$f^\star := f(x^\star) = \min_{x \in \mathbb{R}^d} \; f(x) > -\infty$$

First-order methods: only use gradient information $\nabla f(x)$.

- Gradient Descent (GD), Nesterov's Accelerated Gradient (NAG), etc.
- Widely used in machine learning.

Global convergence theory:

- For $L$-smooth $\mu$-(quasi) strongly convex function, GD converges in $O(\frac{L}{\mu} \log \frac{1}{\epsilon})$ iterations.
- $L$-smooth: $\|\nabla \ell(x) - \nabla \ell(y)\| \leq L\|x - y\|$.
- $\mu$-quasi strongly convex: $f^\star \geq f(x) + \langle \nabla f(x), x^\star - x \rangle + \frac{\mu}{2}\|x - x^\star\|^2$.

Acceleration:

- NAG can accelerate the rate to $O(\sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon})$.
- Dependence on condition number $\frac{L}{\mu}$ is largely improved.

# Matrix factorization: nonconvex and nonsmooth

In ML, objective function can be nonconvex and nonsmooth.
Example: matrix factorization.

$$\min_{X \in \mathbb{R}^{m \times d}, Y \in \mathbb{R}^{n \times d}} f(X, Y) = \frac{1}{2}\|A - XY^\top\|_F^2,$$

- Low-rank: $\mathrm{rank}(A) = r \ll \min(m, n)$, $\kappa = \frac{\sigma_1(A)}{\sigma_r(A)}$.
- Overparameterization: $d \geq r$.
- Nonconvex and not globally smooth: scaling and rotation w/o regularization.

# Matrix factorization: nonconvex and nonsmooth

In ML, objective function can be nonconvex and nonsmooth.
Example: matrix factorization.

$$\min_{X\in\mathbb{R}^{m\times d}, Y\in\mathbb{R}^{n\times d}} f(X, Y) = \frac{1}{2}\|A - XY^\top\|_F^2,$$

- Low-rank: $\text{rank}(A) = r \ll \min(m, n)$, $\kappa = \frac{\sigma_1(A)}{\sigma_r(A)}$.
- Overparameterization: $d \geq r$.
- Nonconvex and not globally smooth: scaling and rotation w/o regularization.

Existing convergence rates:

- GD (Ye and Du'21): $O(d^4(m+n)^2\kappa^4 \log \frac{1}{\epsilon})$.
- GD (Jiang et al'23): $O(\kappa^3 \log \frac{1}{\epsilon})$.
- AltGD (Ward and Kolda'23): $O(d^2(d-r+1)^{-2}\kappa^2 \log \frac{1}{\epsilon})$.

# Matrix factorization: nonconvex and nonsmooth

In ML, objective function can be nonconvex and nonsmooth.
Example: matrix factorization.

$$\min_{X\in\mathbb{R}^{m\times d}, Y\in\mathbb{R}^{n\times d}} f(X,Y) = \frac{1}{2}\|A - XY^\top\|_F^2,$$

- Low-rank: $\text{rank}(A) = r \ll \min(m,n)$, $\kappa = \frac{\sigma_1(A)}{\sigma_r(A)}$.
- Overparameterization: $d \geq r$.
- Nonconvex and not globally smooth: scaling and rotation w/o regularization.

Existing convergence rates:

- GD (Ye and Du'21): $O(d^4(m+n)^2\kappa^4 \log\frac{1}{\epsilon})$.
- GD (Jiang et al'23): $O(\kappa^3 \log\frac{1}{\epsilon})$.
- AltGD (Ward and Kolda'23): $O(d^2(d-r+1)^{-2}\kappa^2 \log\frac{1}{\epsilon})$.

Can we rigorously show GD achieves the same rate as AltGD?

# Matrix factorization: nonconvex and nonsmooth

In ML, objective function can be nonconvex and nonsmooth.
Example: matrix factorization.

$$\min_{X \in \mathbb{R}^{m \times d}, Y \in \mathbb{R}^{n \times d}} f(X, Y) = \frac{1}{2} \|A - XY^\top\|_F^2,$$

- Low-rank: $\text{rank}(A) = r \ll \min(m, n)$, $\kappa = \frac{\sigma_1(A)}{\sigma_r(A)}$.
- Overparameterization: $d \geq r$.
- Nonconvex and not globally smooth: scaling and rotation w/o regularization.

Existing convergence rates:

- GD (Ye and Du'21): $O(d^4(m+n)^2\kappa^4 \log \frac{1}{\epsilon})$.
- GD (Jiang et al'23): $O(\kappa^3 \log \frac{1}{\epsilon})$.
- AltGD (Ward and Kolda'23): $O(d^2(d-r+1)^{-2}\kappa^2 \log \frac{1}{\epsilon})$.

Can we rigorously show GD achieves the same rate as AltGD?
Can NAG provably accelerate matrix factorization?

# Initialization and algorithms

Unbalanced initialization: $X_0 = cA\Phi$, $Y_0 = 0$, where $c > 0$ is large, $\Phi \in \mathbb{R}^{n \times d}$ is a Gaussian random matrix.

# Initialization and algorithms

Unbalanced initialization: $X_0 = cA\Phi$, $Y_0 = 0$, where $c > 0$ is large, $\Phi \in \mathbb{R}^{n \times d}$ is a Gaussian random matrix.

- Can be relaxed to small nonzero Gaussian $Y_0$ as in Ward and Kolda'23.

# Initialization and algorithms

Unbalanced initialization: $X_0 = cA\Phi$, $Y_0 = 0$, where $c > 0$ is large, $\Phi \in \mathbb{R}^{n \times d}$ is a Gaussian random matrix.

- Can be relaxed to small nonzero Gaussian $Y_0$ as in Ward and Kolda'23.
- Important for provable guarantee:
  - Unbalance: scale of $X_0$ much greater than $Y_0$, dominating the dynamics.
  - Sketching: $X_0$ keeps the column space of $A$.

# Initialization and algorithms

Unbalanced initialization: $X_0 = cA\Phi$, $Y_0 = 0$, where $c > 0$ is large, $\Phi \in \mathbb{R}^{n \times d}$ is a Gaussian random matrix.

- Can be relaxed to small nonzero Gaussian $Y_0$ as in Ward and Kolda'23.
- Important for provable guarantee:
  - Unbalance: scale of $X_0$ much greater than $Y_0$, dominating the dynamics.
  - Sketching: $X_0$ keeps the column space of $A$.

Denote residual $R_t = X_t Y_t^\top - A$. GD with step size $\eta$:

$$\begin{cases} X_{t+1} = X_t - \eta R_t Y_t, \\ Y_{t+1} = Y_t - \eta R_t^\top X_t. \end{cases}$$

# Initialization and algorithms

Unbalanced initialization: $X_0 = cA\Phi$, $Y_0 = 0$, where $c > 0$ is large, $\Phi \in \mathbb{R}^{n\times d}$ is a Gaussian random matrix.

- Can be relaxed to small nonzero Gaussian $Y_0$ as in Ward and Kolda'23.
- Important for provable guarantee:
  - Unbalance: scale of $X_0$ much greater than $Y_0$, dominating the dynamics.
  - Sketching: $X_0$ keeps the column space of $A$.

Denote residual $R_t = X_t Y_t^\top - A$. GD with step size $\eta$:

$$\begin{cases} X_{t+1} = X_t - \eta R_t Y_t, \\ Y_{t+1} = Y_t - \eta R_t^\top X_t. \end{cases}$$

NAG with step size $\eta$ and momentum $\beta$:

$$\begin{cases} X_{t+1} = (1+\beta)(X_t - \eta R_t Y_t) - \beta(X_{t-1} - \eta R_{t-1} Y_{t-1}), \\ Y_{t+1} = (1+\beta)(Y_t - \eta R_t^\top X_t) - \beta(Y_{t-1} - \eta R_{t-1}^\top X_{t-1}). \end{cases}$$

# Main results

## Theorem 1 (GD, informal)

*Set $c$ to be a large constant, then with probability at least $1 - e^{-\Theta(d-r+1)}$, GD finds $\|R_T\|_F \leq \epsilon \|A\|_F$ in $T = O\left(d^2(d-r+1)^{-2}\kappa^2 \cdot \log \frac{1}{\epsilon}\right)$ iterations.*

- GD has the same rate as AltGD.

# Main results

## Theorem 1 (GD, informal)

*Set c to be a large constant, then with probability at least* $1 - e^{-\Theta(d-r+1)}$*, GD finds* $\|R_T\|_F \leq \epsilon \|A\|_F$ *in* $T = O\left(d^2(d-r+1)^{-2}\kappa^2 \cdot \log \frac{1}{\epsilon}\right)$ *iterations.*

- GD has the same rate as AltGD.

## Theorem 2 (NAG, informal)

*Set c to be a large constant, then with probability at least* $1 - e^{-\Theta(d-r+1)}$*, NAG finds* $\|R_T\|_F \leq \epsilon \|A\|_F$ *in* $T = O\left(d(d-r+1)^{-1}\kappa \cdot \log \frac{1}{\epsilon}\right)$ *iterations.*

- NAG provably accelerates convergence rate.
- Overparameterization helps convergence.

# Extension to linear neural networks

$$\min_{X\in\mathbb{R}^{m\times d}, Y\in\mathbb{R}^{n\times d}} f(X,Y) = \frac{1}{2}\|L - XY^\top D\|_F^2.$$

- Data matrix: $D \in \mathbb{R}^{n\times N}$, $\text{rank}(D) = \bar{r}$, $\kappa = \frac{\sigma_1(D)}{\sigma_{\bar{r}}(D)}$.
- Label matrix: $L \in \mathbb{R}^{m\times N}$, $\text{rank}(L) = r$. Assume $L = AD$, $\kappa(A) = O(1)$.

# Extension to linear neural networks

$$\min_{X \in \mathbb{R}^{m \times d}, Y \in \mathbb{R}^{n \times d}} f(X, Y) = \frac{1}{2} \|L - XY^\top D\|_F^2.$$

- Data matrix: $D \in \mathbb{R}^{n \times N}$, $\text{rank}(D) = \bar{r}$, $\kappa = \frac{\sigma_1(D)}{\sigma_{\bar{r}}(D)}$.
- Label matrix: $L \in \mathbb{R}^{m \times N}$, $\text{rank}(L) = r$. Assume $L = AD$, $\kappa(A) = O(1)$.

Initialization:

1. $d \geq r - 1 + \Omega(\log \frac{1}{\delta})$, $X_0 = cL\Phi$, $Y_0 = 0$, $\delta \in (0, 1)$.
2. $d \geq m - 1 + \Omega(\log \frac{1}{\delta})$, $X_0 = c\Phi$, $Y_0 = 0$, $\delta \in (0, 1)$.

# Extension to linear neural networks

$$\min_{X\in\mathbb{R}^{m\times d}, Y\in\mathbb{R}^{n\times d}} f(X,Y) = \frac{1}{2}\|L - XY^\top D\|_F^2.$$

- Data matrix: $D \in \mathbb{R}^{n\times N}$, $\mathrm{rank}(D) = \bar{r}$, $\kappa = \frac{\sigma_1(D)}{\sigma_{\bar{r}}(D)}$.
- Label matrix: $L \in \mathbb{R}^{m\times N}$, $\mathrm{rank}(L) = r$. Assume $L = AD$, $\kappa(A) = O(1)$.

Initialization:

1. $d \geq r - 1 + \Omega(\log\frac{1}{\delta})$, $X_0 = cL\Phi$, $Y_0 = 0$, $\delta \in (0,1)$.
2. $d \geq m - 1 + \Omega(\log\frac{1}{\delta})$, $X_0 = c\Phi$, $Y_0 = 0$, $\delta \in (0,1)$.

- Previous width: $\Omega\left(poly(\kappa)\cdot\bar{r}\cdot(m + \log\frac{1}{\delta})\right)$ (Du'19, Wang'21, Liu'22)

# Extension to linear neural networks

$$\min_{X\in\mathbb{R}^{m\times d}, Y\in\mathbb{R}^{n\times d}} \ f(X,Y) = \frac{1}{2}\|L - XY^\top D\|_F^2.$$

- Data matrix: $D \in \mathbb{R}^{n\times N}$, $\text{rank}(D) = \bar{r}$, $\kappa = \frac{\sigma_1(D)}{\sigma_{\bar{r}}(D)}$.
- Label matrix: $L \in \mathbb{R}^{m\times N}$, $\text{rank}(L) = r$. Assume $L = AD$, $\kappa(A) = O(1)$.

Initialization:

1. $d \geq r - 1 + \Omega(\log\frac{1}{\delta})$, $X_0 = cL\Phi$, $Y_0 = 0$, $\delta \in (0,1)$.
2. $d \geq m - 1 + \Omega(\log\frac{1}{\delta})$, $X_0 = c\Phi$, $Y_0 = 0$, $\delta \in (0,1)$.

- Previous width: $\Omega\left(poly(\kappa)\cdot\bar{r}\cdot(m + \log\frac{1}{\delta})\right)$ (Du'19, Wang'21, Liu'22)

## Theorem 3 (LNN, informal)

*With probability at least $1-\delta$, NAG with init 1 and 2 converge in $T_1 = O\left(\frac{d}{d-r+1}\kappa^2\log\frac{1}{\epsilon}\right)$ and $T_2 = O\left(\frac{d}{d-m+1}\kappa\log\frac{1}{\epsilon}\right)$ iterations.*

- Accelerated rate with less width: $\approx$ rank or dimension.

# Numerical experiments

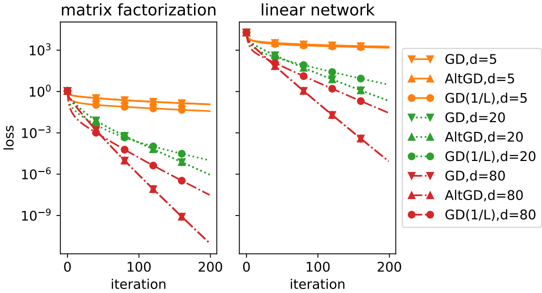


Figure: Experiment 1: AltGD and GD (upper/lower triangle) performance similar with the same step size, while changing step size (round) will change the convergence rate.

# Numerical experiments

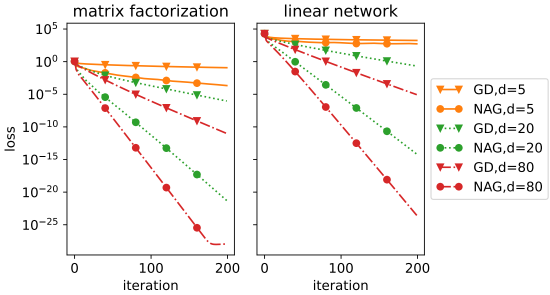


Figure: Experiment 2: NAG (round) converges much faster than GD (triangle) across different overparameterization levels.

# Numerical experiments

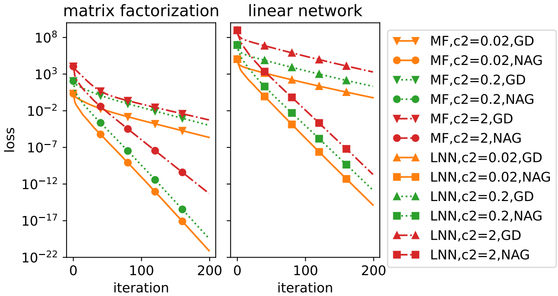


Figure: Experiment 3: Initialize $Y_0 = c_2\Phi_2$ with small $c_2$. As long as unbalanced, $c_2 \leq O(c)$, the rate (slope) will be roughly the same.

# Conclusion

- We show the convergence rate of Gradient Descent as a baseline for matrix factorization under unbalanced initialization. Such initialization is crucial for our analysis.
- Nesterov's Accelerated Gradient can provably accelerate the convergence rate for matrix factorization, despite its nonconvexity, nonsmoothness, and overparameterization.
- Extending the analysis to linear neural networks largely improves the minimum width requirement.

# Thank You!

Link: arxiv.org/abs/2410.09640

Email: zhenghaoxu@gatech.edu