香港中文大學(深圳) The Chinese University of Hong Kong, Shenzhen

MICHIGAN STATE UNIVERSITY

Fast Robust Principle Component Analysis using Gauss-Newton Iterations

William Chettleburgh[1] Zhishen Huang[2,1] Ming Yan[3,1]

[1]Michigan State University [2]Amazon [3]The Chinese University of Hong Kong, Shenzhen

Motivation

The low-rank structure appears in natural images, video/audio signal enhancement, RNA-sequencing, data denoising, social science data, etc.

Two ways to reconstruct the low-rank matrix data $\mathbf{L}$

- Decomposition-based methods: $\mathbf{L}_{m\times n} = \mathbf{X}_{m\times p}(\mathbf{Y}_{n\times p})^\top$
  - p is small
  - fast computation (no singular value decomposition (SVD))
  - requires the true rank p
- Low rank regularization $f(\mathbf{L})$ (e.g., nuclear norm $\|\mathbf{L}\|_*$)
  - requires SVD for a large $m \times n$ matrix (multiple times)
  - does not require the true rank

A combined approach proposed: computing SVD for a small $m \times p$ matrix (multiple times); requires only an upper bound of the true rank.

Basics of Proximal Operators

Proximal Operator

Given a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, the proximal of a function f is

$$\mathbf{prox}_f(\mathbf{M}) = \arg\min_{\mathbf{L}} f(\mathbf{L}) + \frac{1}{2}\|\mathbf{L}-\mathbf{M}\|_{\mathrm{F}}^2$$

Computation of $\mathbf{prox}_f(\mathbf{M})$

- Assume the objective f only depends on the singular values of the input matrix, i.e. $f(\mathbf{L}) = f(\mathbf{\Sigma_L})$, where $\mathbf{L} = \mathbf{U_L}\mathbf{\Sigma_L}\mathbf{V}_\mathbf{L}^\top$ is the singular value decomposition of $\mathbf{L}$.
- Let the input matrix $\mathbf{M}$ have the SVD $\mathbf{M} = \mathbf{U_M}\Sigma_\mathbf{M}\mathbf{V}_\mathbf{M}^\top$, then

$$\mathbf{prox}_f(\mathbf{M}) = \mathbf{U_M}\cdot\mathbf{\Sigma_L}\cdot\mathbf{V}_\mathbf{M}^\top$$

where $\mathbf{\Sigma_L} = \arg\min_{\mathbf{\Sigma_L}} f(\mathbf{\Sigma_L}) + \frac{1}{2}\|\mathbf{\Sigma_L}-\mathbf{\Sigma_M}\|_{\mathrm{F}}^2$

Preparation: Finding a Low-Rank Approximation to the Input of the Proximal Operator

Assuming the output of the proximal operator $\mathbf{L}$ has rank $\mathrm{rank}(\mathbf{L}) \le p$, we have its SVD

$$\begin{aligned}\mathbf{L} &= (\mathbf{U_L})_{m\times p}(\mathbf{\Sigma_L})_{p\times p}\big((\mathbf{V_L})_{n\times p}\big)^\top\\ &= \underbrace{(\mathbf{U_L})_{m\times p}(\mathbf{\Sigma_L})_{p\times p}\mathbf{O}_{p\times p}}_{\mathbf{X}_{m\times p}}\underbrace{(\mathbf{O}_{p\times p})^\top\big((\mathbf{V_L})_{n\times p}\big)^\top}_{(\mathbf{Y}_{n\times p})^\top}.\end{aligned}$$

- $(\mathbf{\Sigma_L})_{p\times p} \in \mathbb{R}^{p\times p}$ is computed from $\mathbf{\Sigma_M} \in \mathbb{R}^{m\times n}$.
- $(\mathbf{U_L})_{m\times p}$ is the first p columns of $(\mathbf{U_M})_{m\times m}$.
- $(\mathbf{V_L})_{n\times p}$ is the first p columns of $(\mathbf{V_M})_{n\times n}$.
- $\mathbf{O}_{p\times p}$ is an orthonormal matrix.
- Compare SVDs of $\mathbf{XX}^\top$ and $\mathbf{MM}^\top$:

$$\begin{aligned}\mathbf{XX}^\top &= (\mathbf{U_L})_{m\times p}(\mathbf{\Sigma_L})_{p\times p}^2((\mathbf{U_L})_{n\times p})^\top\\ \mathbf{MM}^\top &= (\mathbf{U_M})_{m\times m}(\mathbf{\Sigma_M})_{m\times n}((\mathbf{\Sigma_M})_{m\times n})^\top((\mathbf{U_M})_{m\times m})^\top\end{aligned}$$

- To find $\mathbf{L}$, we determine its decomposition components $\mathbf{X}$ and $\mathbf{Y}$.

Main Idea: Computing the proximal operator exactly with a small-sized matrix is fast. We find a low-rank approximation to the input matrix $\mathbf{M}$ through *the Gauss-Newton iteration*.

Gauss-Newton accelerated protocol for sequential $\mathrm{prox}_f(\mathrm{M})$

1. Gauss-Newton iteration [LWZ15, SSY21] to find $\widetilde{\mathbf{X}} = \arg\min_{\mathbf{X}\in\mathbb{R}^{m\times p}} \|\mathbf{XX}^\top - \mathbf{MM}^\top\|_{\mathrm{F}}^2$:

$$\widetilde{\mathbf{X}} \leftarrow \mathbf{MM}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} - \mathbf{X}\big((\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{MM}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} - \mathbf{I}\big)/2$$

2. $\mathbf{Y} = \mathbf{M}^\top\widetilde{\mathbf{X}}(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1} = (\mathbf{V_M})_{1:n,1:p}\mathbf{O}$ (by-product)
3. Compute $\mathbf{X} = \mathrm{prox}_f(\widetilde{\mathbf{X}}) = (\mathbf{U_M})_{1:m,1:p}\mathbf{\Sigma_L}\mathbf{O}$ (Note $\tilde{\mathbf{X}} \in \mathbb{R}^{m\times p}$) and output $\mathbf{XY}^\top$.

Application: Robust PCA Setup

Recover a low-rank component $\mathbf{L}$ and a sparse component $\mathbf{S}$ from a noisy data matrix $\mathbf{D} \in \mathbb{R}^{m\times n}$:

$$\min_{\mathbf{L},\mathbf{S}} \underbrace{\|\mathbf{L}\|_*}_{\text{low-rank}} + \underbrace{\beta\|\mathbf{S}\|_0}_{\text{sparse}} + \underbrace{\lambda\|\mathbf{D}-\mathbf{L}-\mathbf{S}\|_1}_{\text{robust noisy}},$$

where β and λ are two given constant parameters.

Robust PCA Algorithm [LR19]

Algorithm 1 Sparsity regularized principal component pursuit

```
while not converged do
    with an index set Φ, update L (can be accelerated by GN protocol) :
```

$$\mathbf{L} = \arg\min_{\mathbf{L}} \|\mathbf{L}\|_* + \lambda\|\mathcal{P}_\Phi(\mathbf{D}-\mathbf{L})\|_1$$

using ADMM. Here $\mathcal{P}_\Phi$ is the projection onto the index set.

fix $\mathbf{L}$, update $\mathbf{S}$ (has an analytical solution) :

$$\mathbf{S} = \arg\min_{\mathbf{S}} \beta\|\mathbf{S}\|_0 + \lambda\|\mathbf{D}-\mathbf{L}-\mathbf{S}\|_1$$

end while

ADMM has the *proximal* computation step $\mathbf{L}^{(k+1)} = \arg\min_{\mathbf{L}} \frac{1}{\theta}\|\mathbf{L}\|_* + \frac{1}{2}\|\mathbf{L}-\mathbf{M}\|_{\mathrm{F}}^2$, where $\mathbf{M}$ is an intermediate matrix. This is where the acceleration protocol kicks in.

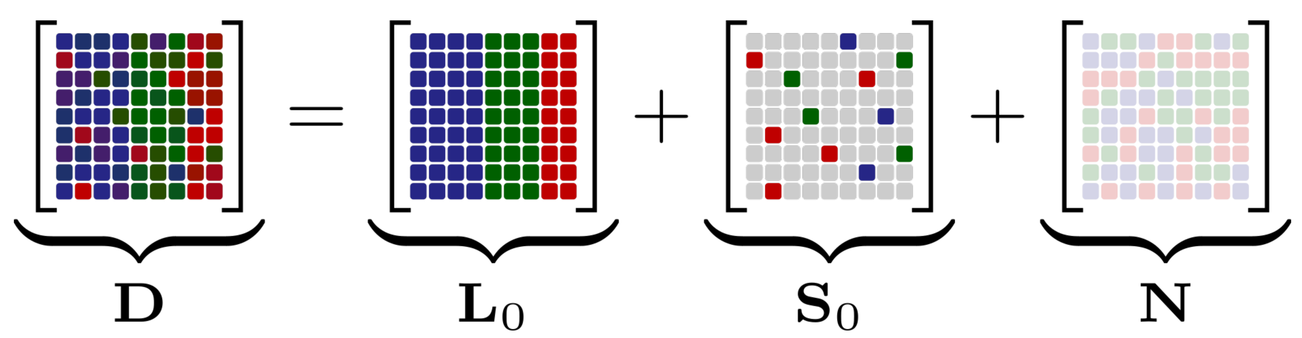
- The function $\frac{1}{\theta}\|\mathbf{L}\|_*$ only depends on the singular values of the input matrix $\mathbf{L}$.
- Only 2-3 Gauss-Newton iterations are sufficient for solving the dense component.

Synthetic Data Setup

$\mathbf{L}_0$: Low-rank component, formed by multiplying standard Gaussian matrices of size $n \times r$ and $r \times n$.

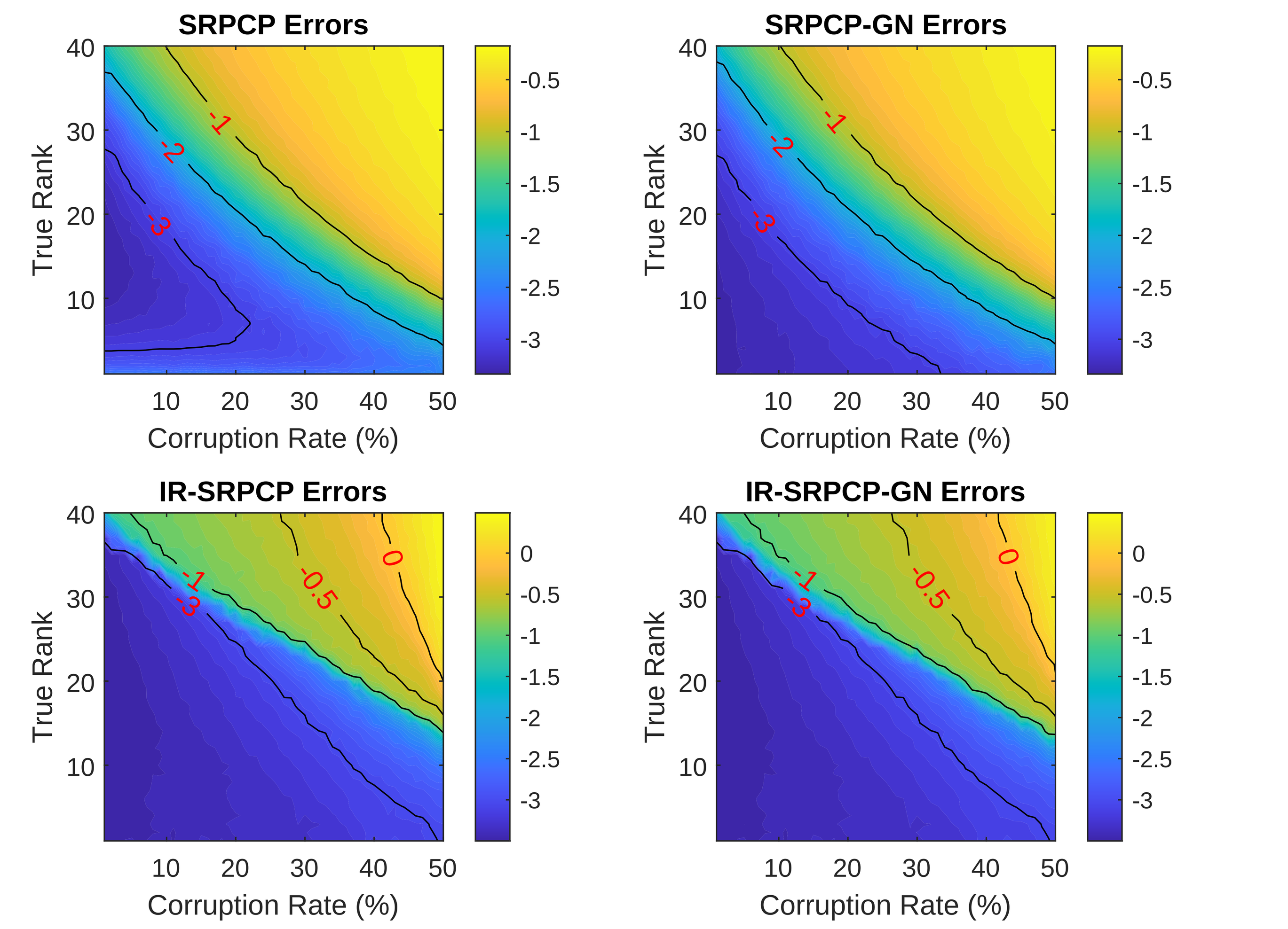
$\mathbf{S}_0$: Sparse matrix with ρn^2 outliers uniformly chosen in $[-100, 100]$.

$\mathbf{N}$: Gaussian noise with mean 0 and variance σ^2.

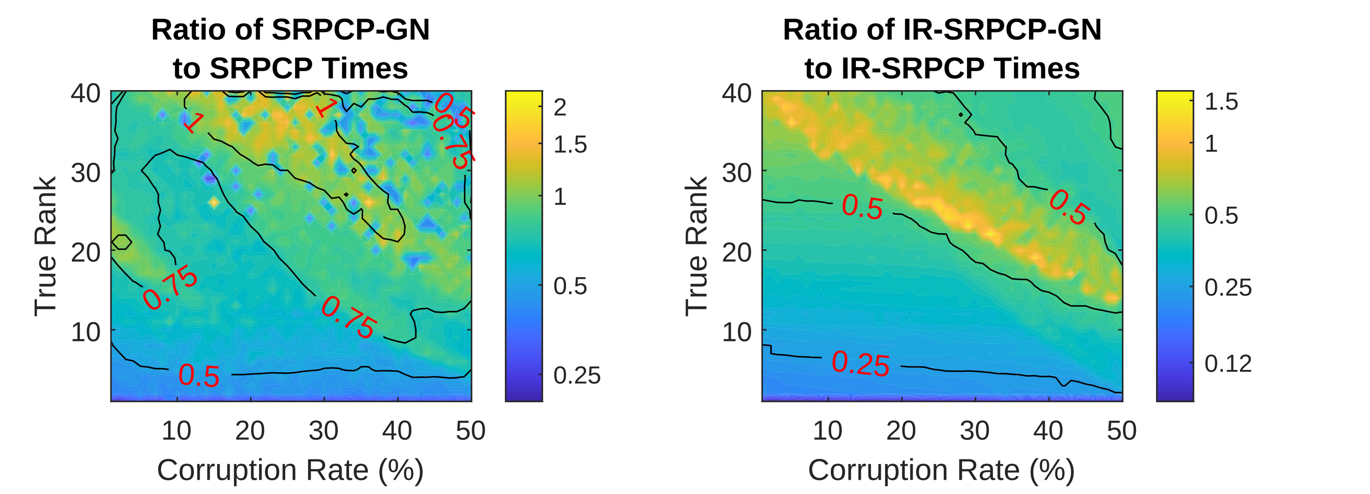


We average results over 10 repetitions for each combination of $r \in \{1:40\}$ and $\rho \in \{0.01:0.01:0.50\}$, with parameters $n = 100$, $\sigma = 0.1$, $\lambda = 0.1$, $\gamma = 40$ (see paper), $\beta = 2$, and $p = r + 5$ upper bound on the rank.

RPCA Error and Run-time Comparison



- Reconstruction error measured by $\frac{\|\hat{\mathbf{L}}-\mathbf{L}_0\|_{\mathrm{F}}}{\|\mathbf{L}_0\|_{\mathrm{F}}}$ and plotted on a log scale $2\log_{10}(\cdot)$.
- The proposed methods (right) have similar performance as the original (left).



- SRPCP-GN has a runtime that is only 76% as long as SRPCP on average, and it finishes before SRPCP in 84% of the tests.
- IR-SRPCP-GN takes 47% of the time that IR-SRPCP takes on average, and it finishes before IR-SRPCP 98% of the time.

Stability over Hyperparameters

With the Gauss-Newton acceleration protocol in the RPCA algorithm, the solver shows a wider area of stability with respect to the upper bound of the rank and hyperparameters (β and γ) (see paper).

References

[LR19] Jing Liu and Bhaskar D. Rao. Robust pca via ℓ_0-ℓ_1 regularization. *IEEE Transactions on Signal Processing*, 67(2):535–549, 2019.

[LWZ15] Xin Liu, Zaiwen Wen, and Yin Zhang. An efficient gauss–newton algorithm for symmetric low-rank product matrix approximations. *SIAM Journal on Optimization*, 25(3):1571–1608, 2015.

[SSY21] Ningyu Sha, Lei Shi, and Ming Yan. Fast algorithms for robust principal component analysis with an upper bound on the rank. *Inverse Problems and Imaging*, 15(1):109–128, 2021.

yanming@cuhk.edu.cn