

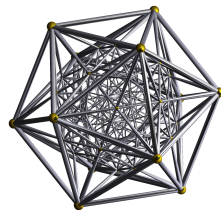
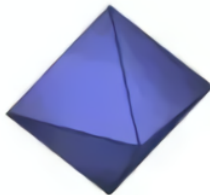
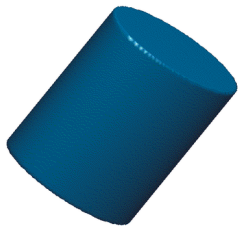
Computational Principles for High-dim Data Analysis

(Lecture Eleven)

Yi Ma and Jiantao Jiao

EECS Department, UC Berkeley

October 5, 2021



Decomposing Low-Rank and Sparse Matrices (Principal Component Pursuit)

- 1 Problem and Motivating Example
- 2 Principal Component Pursuit

"The whole is greater than the sum of the parts."
– Aristotle, *Metaphysics*

Problem Formulation: Mixture of Sparse and Low-Rank

Given a large data matrix $\mathbf{Y} \in \mathbb{R}^{n_1 \times n_2}$ which is a superposition of two unknown matrices:

$$\mathbf{Y} = \mathbf{L}_o + \mathbf{S}_o, \quad (1)$$

where

- $\mathbf{L}_o \in \mathbb{R}^{n_1 \times n_2}$ is a low-rank matrix;
- $\mathbf{S}_o \in \mathbb{R}^{n_1 \times n_2}$ is a sparse matrix.

Problem: Can we hope to efficiently recover both \mathbf{L}_o and \mathbf{S}_o ?

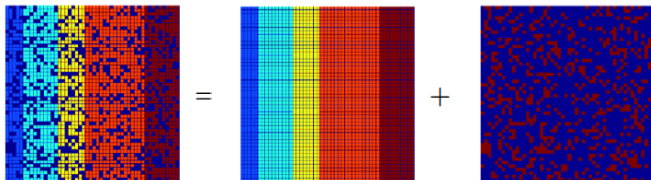
Compare this with the classic model:

$$\mathbf{Y} = \mathbf{L}_o + \mathbf{Z}_o, \quad (2)$$

where \mathbf{Z}_o is dense but small, say Gaussian, noise?

PCA versus Robust PCA.

Complexity of Low-Rank Sparse Decomposition



Definition (Matrix Rigidity)

The *rigidity* of a matrix M (relative to rank r matrices) is defined to be:

$$R_M(r) \doteq \min\{\|S\|_0 : \text{rank}(M + S) \leq r\}, \quad (3)$$

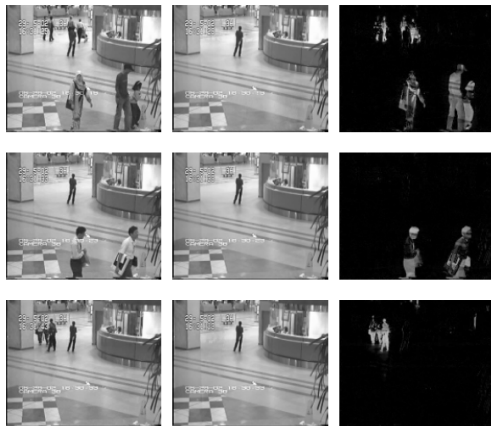
the smallest # of entries modified in order to change M rank r .

Computing matrix rigidity is NP-Hard¹, so is decomposition.

¹On the complexity of matrix rank and rigidity. Meena Mahajan and Jayalal Sarma M.N., 2007

Examples of Low-Rank Sparse Decomposition

Example. A sequence of video frames can be modeled as a static background (low-rank) and moving foreground (sparse).



(a) Original frames

(b) Low-rank \tilde{L} (c) Sparse \hat{S}

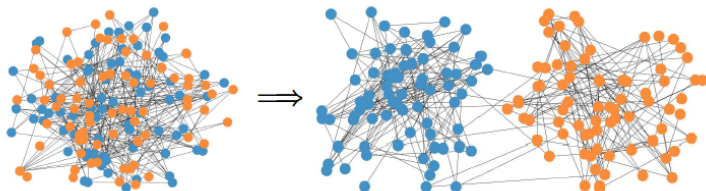
Examples of Low-Rank Sparse Decomposition

Example. A set of face images of the same person under different lightings can be modeled as a low-dimensional, 3 ~ 9D (see Chapter 14), subspace and sparse occlusions and corruptions (specularities).



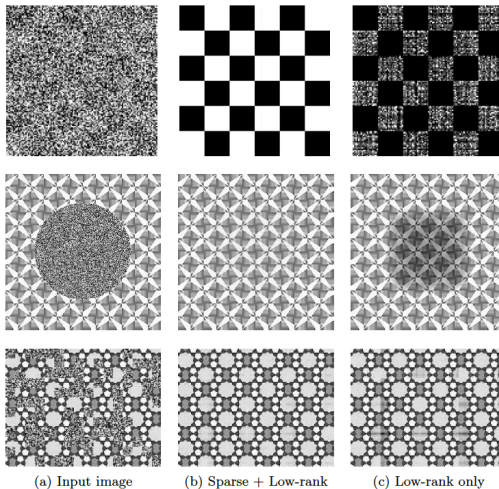
Examples of Low-Rank Sparse Decomposition

Example. Finding communities in a large social networks. Each community can be modeled as a clique of the social graph \mathcal{G} , hence a rank-1 block in the connectivity matrix \mathbf{M} . Hence \mathbf{M} is a low-rank matrix and some sparse connections across communities.



Examples of Low-Rank Sparse Decomposition

Example. Structured regular texture recovery (Chapter 15).



and many more...

Convex Relaxation: Principal Component Pursuit

Optimization problem:

$$\text{minimize } \text{rank}(\mathbf{L}) + \lambda \|\mathbf{S}\|_0 \quad \text{subject to } \mathbf{L} + \mathbf{S} = \mathbf{Y}, \quad (4)$$

which is intractable. Consider **convex relaxation**:

$$\|\mathbf{S}\|_0 = \#\{S_{ij} \neq 0\} \rightarrow \|\mathbf{S}\|_1 = \sum_{ij} |S_{ij}| \quad (\ell^1 \text{ norm}). \quad (5)$$

$$\text{rank}(\mathbf{L}) = \#\{\sigma_i(\mathbf{L}) \neq 0\} \rightarrow \|\mathbf{L}\|_* = \sum_i \sigma_i(\mathbf{L}) \quad (\text{nuclear norm}) \quad (6)$$

Principal Component Pursuit (PCP):

$$\text{minimize } \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \quad \text{subject to } \mathbf{L} + \mathbf{S} = \mathbf{Y}. \quad (7)$$

Alternating Directions Method of Multipliers (ADMM)

Augmented Lagrangian

$$\mathcal{L}_\mu(\mathbf{L}, \mathbf{S}, \mathbf{\Lambda}) = \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \langle \mathbf{\Lambda}, \mathbf{L} + \mathbf{S} - \mathbf{Y} \rangle + \frac{\mu}{2}\|\mathbf{L} + \mathbf{S} - \mathbf{Y}\|_F^2 \quad (8)$$

Instead of

$$(\mathbf{L}_{k+1}, \mathbf{S}_{k+1}) = \arg \min_{\mathbf{L}, \mathbf{S}} \mathcal{L}_\mu(\mathbf{L}, \mathbf{S}, \mathbf{\Lambda}_k), \quad (9)$$

we realize

$$\arg \min_{\mathbf{S}} \mathcal{L}_\mu(\mathbf{L}, \mathbf{S}, \mathbf{\Lambda}) = \mathcal{S}_{\lambda/\mu}(\mathbf{Y} - \mathbf{L} - \mu^{-1}\mathbf{\Lambda}) \quad (10)$$

$$\arg \min_{\mathbf{L}} \mathcal{L}_\mu(\mathbf{L}, \mathbf{S}, \mathbf{\Lambda}) = \mathcal{D}_{1/\mu}(\mathbf{Y} - \mathbf{S} - \mu^{-1}\mathbf{\Lambda}) \quad (11)$$

Soft-thresholding operators

Recall

$$\mathcal{S}_\tau(x) = \text{sgn}(x) \max(|x| - \tau, 0) \quad (12)$$

For matrix $M = U\Sigma V^*$, we define the singular value thresholding operator

$$\mathcal{D}_\tau(M) = U\mathcal{S}_\tau(\Sigma)V^*. \quad (13)$$

Dominating computation is $\mathcal{D}_{1/\mu}$, can speed up using partial SVD.

Algorithm: Alternating Direction Minimization

- 1: **initialize:** $\mathbf{S}_0 = \mathbf{\Lambda}_0 = 0, \mu > 0$.
- 2: **while** not converged **do**
- 3: compute $\mathbf{L}_{k+1} = \mathcal{D}_{1/\mu}(\mathbf{Y} - \mathbf{S}_k - \mu^{-1}\mathbf{\Lambda}_k)$
- 4: compute $\mathbf{S}_{k+1} = \mathcal{S}_{\lambda/\mu}(\mathbf{Y} - \mathbf{L}_{k+1} - \mu^{-1}\mathbf{\Lambda}_k)$
- 5: compute $\mathbf{\Lambda}_{k+1} = \mathbf{\Lambda}_k + \mu(\mathbf{L}_{k+1} + \mathbf{S}_{k+1} - \mathbf{Y})$.
- 6: **end while**

Algorithm: A Little Lesson from History

Comparison from chronological development of algorithms for solving the PCP problem: **the older the algorithm, the more efficient!**

GOOD NEWS: Scalable first-order gradient-descent algorithms:

- Proximal Gradient [Osher, Mao, Dong, Yin '09, Wright et. al.'09, Cai et. al.'09].
- Accelerated Proximal Gradient [Nesterov '83, Beck and Teboulle '09]:
- Augmented Lagrange Multiplier [Hestenes '69, Powell '69]:
- Alternating Direction Method of Multipliers [Gabay and Mercier '76].

For a 1000x1000 matrix of rank 50, with 10% (100,000) entries randomly corrupted: $\min \|A\|_* + \lambda \|E\|_1 \text{ subj } A + E = D.$

Algorithms	Accuracy	Rank	$\ E\ _0$	# iterations	time (sec)
IT	5.99e-006	50	101,268	8,550	119,370.3
DUAL	8.65e-006	50	100,024	822	1,855.4
APG	5.85e-006	50	100,347	134	1,468.9
APG _p	5.91e-006	50	100,347	134	82.7
EALM _p	2.07e-007	50	100,014	34	37.5
IALM _p	3.83e-007	50	99,996	23	11.8

**10,000
times
speedup!**



Empirical success rate

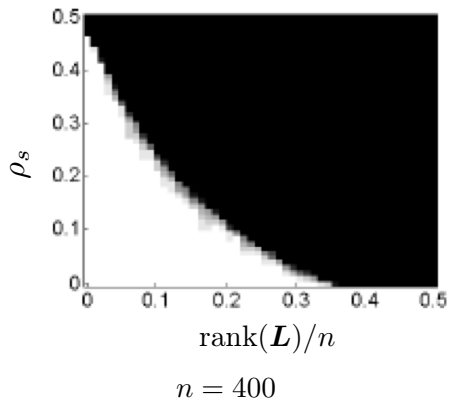


Fig. credit: Candès, Li, Ma, Wright '11

When is decomposition possible?

Identifiability issue: a matrix might be simultaneously low-rank and sparse!

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{\text{sparse and low-rank}} \quad \text{vs.} \quad \underbrace{\begin{bmatrix} 1 & 0 & 1 & \cdots & 1 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{\text{sparse but not low-rank}}$$

Nonzero entries of sparse component need to be spread out

— This lecture: assume locations of the nonzero entries are random

When is decomposition possible?

Identifiability issue: a matrix might be simultaneously low-rank and sparse!

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & & \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}}_{\text{low-rank and dense}} \quad \text{vs.} \quad \underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{\text{low-rank but sparse}}$$

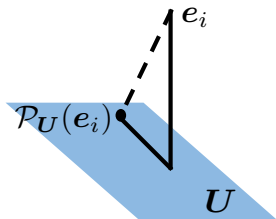
The low-rank component needs to be incoherent

Low-rank component: incoherence

Definition

Incoherence parameter μ_1 of $\mathbf{L}_o = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ is the smallest quantity s.t.

$$\max_i \|\mathbf{e}_i^* \mathbf{U}\|_2^2 \leq \frac{\mu_1 r}{n} \quad \text{and} \quad \max_i \|\mathbf{e}_i^* \mathbf{V}\|_2^2 \leq \frac{\mu_1 r}{n}$$



Low-rank component: joint coherence

Definition (Joint coherence)

Joint coherence parameter μ_2 of $\mathbf{L}_o = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ is the smallest quantity s.t.

$$\|\mathbf{UV}^*\|_\infty \leq \sqrt{\frac{\mu_2 r}{n^2}}$$

This prevents \mathbf{UV}^* from being too peaky

- $\mu_1 \leq \mu_2 \leq \mu_1^2 r$, since

$$|(\mathbf{UV}^*)_{ij}| = |\mathbf{e}_i^\top \mathbf{UV}_j^*| \leq \|\mathbf{e}_i^\top \mathbf{U}\|_2 \cdot \|\mathbf{V}_j^*\|_2 \leq \frac{\mu_1 r}{n}$$

$$\|\mathbf{UV}^*\|_\infty^2 \geq \frac{\|\mathbf{UV}^* \mathbf{e}_j\|_F^2}{n} = \frac{\|\mathbf{V}_j^*\|_2^2}{n} = \frac{\mu_1 r}{n^2} \quad (\text{suppose } \|\mathbf{V}_j^*\|_2^2 = \frac{\mu_1 r}{n})$$

In the book we have set $\mu_1 = \mu_2 = \nu$.

Theoretical guarantee

Theorem (Candès, Li, Ma, Wright '11)

- $\text{rank}(\mathbf{L}) \lesssim \frac{n}{\max\{\mu_1, \mu_2\} \log^2 n}$;
- *Nonzero entries of \mathbf{S} are randomly located, and $\|\mathbf{S}\|_0 \leq \rho_s n^2$ for some constant $\rho_s > 0$ (e.g. $\rho_s = 0.2$).*

Then PCP with $\lambda = 1/\sqrt{n}$ is exact with high prob.

- $\text{rank}(\mathbf{L})$ can be quite high (up to $n/\text{polylog}(n)$)
- Parameter free: $\lambda = 1/\sqrt{n}$
- Ability to correct gross error: $\|\mathbf{S}\|_0 \asymp n^2$
- Sparse component \mathbf{S} can have arbitrary magnitudes / signs!

Geometry

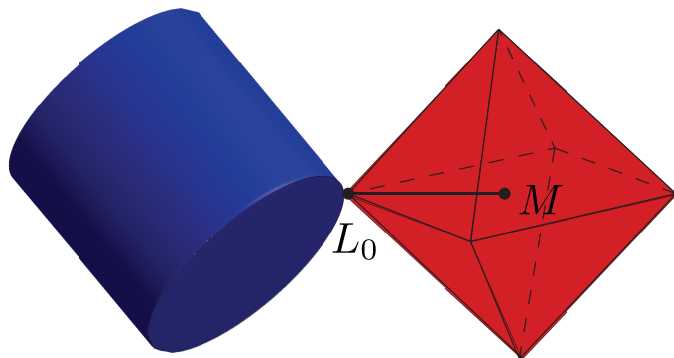


Fig. credit: Candès '14

Dense error correction

Theorem (Ganesh, Wright, Li, Candès, Ma '10, Chen, Jalali, Sanghavi, Caramanis '13)

- $\text{rank}(\mathbf{L}) \lesssim \frac{n}{\max\{\mu_1, \mu_2\} \log^2 n}$;
- *Nonzero entries of \mathbf{S} are randomly located, have random sign, and $\|\mathbf{S}\|_0 = \rho_s n^2$.*

Then PCP with $\lambda \asymp \sqrt{\frac{1-\rho_s}{\rho_s n}}$ succeeds with high prob., provided that

$$\underbrace{1 - \rho_s}_{\text{non-corruption rate}} \gtrsim \sqrt{\frac{\max\{\mu_1, \mu_2\} r \text{polylog}(n)}{n}}$$

- When additive corruptions have random signs, PCP works even when *a dominant fraction* of the entries are corrupted

Is joint coherence needed?

- Matrix completion: does not need μ_2
- Robust PCA: so far we need μ_2

Question: Can we recover \mathbf{L} with rank up to $\frac{n}{\mu_1 \text{polylog}(n)}$ (rather than $\frac{n}{\max\{\mu_1, \mu_2\} \text{polylog}(n)}$)?

Answer: No

Planted clique problem

Setup: a graph \mathcal{G} of n nodes generated as follows

1. connect each pair of nodes independently with prob. 0.5
2. pick n_0 nodes and make them a clique (fully connected)

Goal: find the hidden clique from \mathcal{G}

Information theoretically, one can recover the clique if $n_0 > 2 \log_2 n$

Conjecture on computational barrier

Conjecture: \forall constant $\epsilon > 0$, if $n_0 \leq n^{0.5-\epsilon}$, then no tractable algorithm can find the clique from \mathcal{G} with prob. $1 - o(1)$

— often used as a hardness assumption

Lemma

If there is an algorithm that allows recovery of any \mathbf{L} from \mathbf{Y} with $\text{rank}(\mathbf{L}) \leq \frac{n}{\mu_1 \text{polylog}(n)}$, then the above conjecture is violated.

Proof of Lemma 6

Suppose \mathbf{L} is the true adjacency matrix,

$$L_{i,j} = \begin{cases} 1, & \text{if } i, j \text{ are both in the clique} \\ 0, & \text{else} \end{cases}$$

Let \mathbf{A} be the adjacency matrix of \mathcal{G} , and generate \mathbf{Y} s.t.

$$Y_{i,j} = \begin{cases} A_{i,j}, & \text{with prob. } 2/3 \\ 0, & \text{else} \end{cases}$$

Therefore, one can write

$$\mathbf{Y} = \mathbf{L} + \underbrace{\mathbf{Y} - \mathbf{L}}_{\text{each entry is nonzero w.p. } 1/3}$$

Proof of Lemma 6

Note that

$$\mu_1 = \frac{n}{n_0} \quad \text{and} \quad \mu_2 = \frac{n^2}{n_0^2}$$

If there is an algorithm that can recover any \mathbf{L} of rank $\frac{n}{\mu_1 \text{polylog}(n)}$ from \mathbf{M} , then

$$\text{rank}(\mathbf{L}) = 1 \leq \frac{n}{\mu_1 \text{polylog}(n)} \iff n_0 \geq \text{polylog}(n)$$

But this contradicts the conjecture (which claims computational infeasibility to recover \mathbf{L} unless $n_0 \geq n^{0.5-o(1)}$)

Assignments

- Reading: Section 5.1 - 5.3 of Chapter 5.
- Written Homework #3.