# Computational Intelligence Laboratory

## Lecture 11

## Robust Principal Component Analysis

**Thomas Hofmann**

ETH Zurich – `cil.inf.ethz.ch`

June 2, 2017

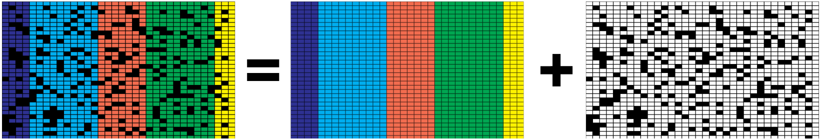
Section 1

# Robust PCA

# Robust Principal Component Analysis

**Goal:** Find a low rank representation of a matrix $\mathbf{X}$, which is corrupted by a sparse perturbation or sparse structured noise.

$$\underset{\text{original}}{\boxed{\mathbf{X}}} \approx \underset{\text{low-rank}}{\boxed{\mathbf{L_0}}} + \underset{\text{sparse}}{\boxed{\mathbf{S_0}}}$$



The low-rank matrix specifies the signal in the data and the sparse matrix captures the perturbation or structured noise.

# Additive matrix decomposition

## Matrix Factorization

... is what you have seen so far in PCA, SVD, Clustering, pLSA, Compressive Sensing, and sparse dictionary learning:

$$\underset{D \times N}{\boxed{\mathbf{X}}} \approx \underset{D \times K}{\boxed{\mathbf{A}}} \cdot \underset{K \times N}{\boxed{\mathbf{B}}}$$

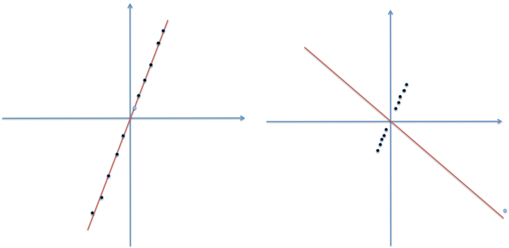
## Decomposition into Sum

In this chapter: *Additive decomposition* into a sum of matrices with specific properties (like sparsity, rank, ...)

$$\underset{\text{original}}{\boxed{\mathbf{X}}} \approx \underset{\text{low-rank}}{\boxed{\mathbf{L_0}}} + \underset{\text{sparse}}{\boxed{\mathbf{S_0}}}$$

# Robustness of Classical PCA

## Very sensitive to outliers



*Single* corrupted point completely changes principal component
$\Longrightarrow$ breakpoint for PCA is zero

- The *breakpoint* of an estimator is defined as the smallest proportion of data elements which can be changed without resulting in an arbitrarily-large change in the estimator.

# Motivation for Robust PCA

How can we make PCA more robust?
*A:* Explicitly model the errors as sparse noise!

Find decomposition into low-rank and sparse matrix: $\mathbf{X} = \mathbf{L_0} + \mathbf{S_0}$.

## Separation into low-rank and sparse

$$\min_{\mathbf{L},\mathbf{S}} \; \mathsf{rank}(\mathbf{L}) + \mu \|\mathbf{S}\|_0, \quad \text{s.t. } \mathbf{L} + \mathbf{S} = \mathbf{X}$$

# Advantages and Applications of Robust PCA

- Robust to strongly corrupted observations of $\mathbf{X}$ (outliers)
- Does not need to know sparsity pattern of $\mathbf{S_0}$ beforehand
- Can easily be extended to matrix completion.

**Applications** where large errors frequently occur are

| | |
|---|---|
| *Image Processing* | Salt & Pepper noise |
| *Web Data Analysis* | Adversarial information |
| *Bioinformatics* | Spurious errors in measurements, sensor failures |
| *Computer Vision* | Occlusions |

Section 2

# Convex Robust PCA

# Separating low rank and sparse matrix components

## Additive decomposition problem

$$\begin{aligned} \min_{\mathbf{L},\mathbf{S}} \quad & \mathsf{rank}(\mathbf{L}) + \mu \|\mathbf{S}\|_0 \\ \text{subject to} \quad & \mathbf{L} + \mathbf{S} = \mathbf{X} \end{aligned}$$

- separating a matrix into a low-rank part and a sparse part
- decomposition is difficult to compute in general (non-convex problem)
- $\Longrightarrow$ solve alternative, surrogate problem ...

# Principal Component Pursuit (PCP)

$$\begin{aligned} \min_{\mathbf{L},\mathbf{S}} \quad & \|\mathbf{L}\|_* + \mu\|\mathbf{S}\|_1 \\ \text{subject to} \quad & \mathbf{L} + \mathbf{S} = \mathbf{X} \end{aligned}$$

- $\|\cdot\|_*$ nuclear norm: $\|\mathbf{M}\|_* = \sum_{i=1}^{\min\{m,n\}} \sigma_i$
- $\|\cdot\|_1$ sum of absolute values of entries, $\|\mathbf{M}\|_1 = \sum_{i,j} |m_{ij}|$
- Notes:
  - $\|\cdot\|_1$: convex relaxation of cardinality
  - $\|\cdot\|_*$: convex relaxation of rank

# Principal Component Pursuit (PCP)

Remember: want to solve following problem:

$$\begin{aligned} \min_{\mathbf{L},\mathbf{S}} \quad & \|\mathbf{L}\|_* + \mu \|\mathbf{S}\|_1 \\ \text{subject to} \quad & \mathbf{L} + \mathbf{S} = \mathbf{X} \end{aligned}$$

- Optimization: Alternating Direction Method of Multipliers
  - builds upon two other methods: dual decomposition & method of multipliers

# Recap: Convex Optimization

## Convex Optimization with Equality Constraints

$$\begin{array}{rl} \min\limits_{\mathbf{x}} & f(\mathbf{x}) \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \end{array}$$

- Lagrangian: $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top(\mathbf{A}\mathbf{x} - \mathbf{b})$
- Dual function: $\mathcal{D}(\boldsymbol{\lambda}) = \inf_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$
- Dual problem: $\max_{\boldsymbol{\lambda}} \mathcal{D}(\boldsymbol{\lambda}), \quad \boldsymbol{\lambda}^\star \in \arg\max_{\boldsymbol{\lambda}} \mathcal{D}(\boldsymbol{\lambda})$
- Recover optimal solution: $\mathbf{x}^\star \in \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^\star)$

# Dual Ascent

## Gradient Method for Dual Problem

$$\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \eta^t \nabla \mathcal{D}(\boldsymbol{\lambda}^t)$$

$$\nabla \mathcal{D}(\boldsymbol{\lambda}) = \mathbf{A}\mathbf{x}^* - \mathbf{b}, \quad \text{for} \quad \mathbf{x}^* \in \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$$

- $\nabla \mathcal{D}$: gradient of the dual function
- $\eta^t$: step size sequence

# Dual Decomposition

- Suppose $f(\mathbf{x})$ is separable

$$\min_{\mathbf{x}} \left[ f(\mathbf{x}) := f_1(\mathbf{x}_1) + \ldots + f_N(\mathbf{x}_N) \right], \quad \mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$$
$$s.t. \quad \left[ A\mathbf{x} := \textstyle\sum_{i=1}^N A_i \mathbf{x}_i \right] = \mathbf{b}$$

(Meaning: each $\mathbf{x}_i \in \mathbb{R}^{d_i}$ is a shorter vector, and their concatenation is $\mathbf{x} \in \mathbb{R}^d$.
$\mathbf{A}_i$ always exist: partitioning columns of $\mathbf{A}$)

- $\Rightarrow$ Lagrangian $\mathcal{L}$ also becomes separable

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathcal{L}_1(\mathbf{x}_1, \boldsymbol{\lambda}) + \ldots + \mathcal{L}_N(\mathbf{x}_N, \boldsymbol{\lambda}) - \boldsymbol{\lambda}^\top \mathbf{b}$$
$$\mathcal{L}_i(\mathbf{x}_i, \boldsymbol{\lambda}) := f_i(\mathbf{x}_i) + \boldsymbol{\lambda}^\top \mathbf{A}_i \mathbf{x}_i$$

- Split $\mathbf{x}$-minimization step

$$\mathbf{x}_i^{t+1} := \arg\min_{\mathbf{x}_i} \mathcal{L}_i(\mathbf{x}_i, \boldsymbol{\lambda}^t)$$

# Dual Decomposition

## Dual Decomposition (for Dual Ascent)

$$\mathbf{x}_i^{t+1} := \arg\min_{\mathbf{x}_i} \mathcal{L}_i(\mathbf{x}_i, \boldsymbol{\lambda}^t), \quad i = 1, \ldots, N$$

$$\boldsymbol{\lambda}^{t+1} := \boldsymbol{\lambda}^t + \eta^t \left( \sum_{i=1}^{N} \mathbf{A}_i \mathbf{x}_i^{t+1} - \mathbf{b} \right)$$

- sub-problems can be optimized independently (e.g. in parallel)

# Dual Decomposition

**Algorithm 1** Parallelizable Dual Ascent Algorithm

initialize $\boldsymbol{\lambda}^0$, $\mathbf{x}^0$
**for** $t = 1, \dots, T$ **do**
  **for** $i = 1, \dots, N$ **do**
    $\mathbf{x}_i^t \leftarrow \arg\min_{\mathbf{z}} \mathcal{L}_i(\mathbf{z}, \boldsymbol{\lambda}^{t-1})$
    $\Theta_i^t \leftarrow \mathbf{A}_i \mathbf{x}_i^t$
  **end for**
  $\boldsymbol{\lambda}^t \leftarrow \boldsymbol{\lambda}^{t-1} + \eta^t \left( \sum_{i=i}^N \Theta_i^t - \mathbf{b} \right)$
**end for**
return $\mathbf{x}^T$

# Method of Multipliers

To overcome some problems of dual ascent, namely convergence only under strict convexity and finiteness of $f$, we introduce:

## Augmented Lagrangian

$$\mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top(\mathbf{A}\mathbf{x} - \mathbf{b}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

For any feasible $\mathbf{x}$: $\mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}) = \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x})$

## Method of Multipliers

$$\begin{aligned} \mathbf{x}^{t+1} &:= \arg\min_{\mathbf{x}} \mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}^t) \\ \boldsymbol{\lambda}^{t+1} &:= \boldsymbol{\lambda}^t + \rho(\mathbf{A}\mathbf{x}^{t+1} - \mathbf{b}) \end{aligned}$$

- additional penalty for violating the constraints
- converges under more general conditions than dual ascent

# Dual Update Step - Why Choose $\rho$ as Step Size?

## Method of Multipliers

$$\mathbf{x}^{t+1} := \arg\min_{\mathbf{x}} \mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}^t), \quad \boldsymbol{\lambda}^{t+1} := \boldsymbol{\lambda}^t + \rho(\mathbf{A}\mathbf{x}^{t+1} - \mathbf{b})$$

**Optimality Conditions.** 'Dual feasibility'

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \nabla_{\mathbf{x}} f(\mathbf{x}) + \mathbf{A}^\top \boldsymbol{\lambda} \stackrel{!}{=} 0$$

First order optimality of $\mathbf{x}^{t+1}$

$$\begin{aligned} 0 &= \nabla_{\mathbf{x}} \mathcal{L}_\rho(\mathbf{x}^{t+1}, \boldsymbol{\lambda}^t) \\ &= \nabla_{\mathbf{x}} f(\mathbf{x}^{t+1}) + \mathbf{A}^\top \left(\boldsymbol{\lambda}^t + \rho(\mathbf{A}\mathbf{x}^{t+1} - \mathbf{b})\right) \\ &= \nabla_{\mathbf{x}} f(\mathbf{x}^{t+1}) + \mathbf{A}^\top \boldsymbol{\lambda}^{t+1} \end{aligned}$$

- using $\rho$ as step size: iterate $(\mathbf{x}^{t+1}, \boldsymbol{\lambda}^{t+1})$ always dual feasible
- convergence of $f(\mathbf{x}^t)$ and residual $\mathbf{A}\mathbf{x}^t - \mathbf{b}$ can be shown (under reasonable assumptions)

# Alternating Direction Method of Multipliers

- Disadvantage of the method of multipliers: augmented Lagrangian $\mathcal{L}_\rho$ is not separable
  - $\Longrightarrow$ can not parallelize $\mathbf{x}$-minimization
- Alternating Direction Method of Multipliers:
  - superior convergence properties $+$ decomposability
  - split primal variable into two parts: $\mathbf{x}$ and $\mathbf{z}$
  - assumption: objective function is separable over splitting
  - *separately* minimize $\mathbf{x}$ and $\mathbf{z}$

# Alternating Direction Method of Multipliers

$$\begin{aligned} \min_{\mathbf{x}_1, \mathbf{x}_2} \quad & f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) \qquad f_1, f_2 \text{ convex} \\ \text{subject to} \quad & \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 = \mathbf{b} \end{aligned}$$

## Augmented Lagrangian

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\lambda}) \;=\; f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) \; & + \boldsymbol{\lambda}^\top(\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}) \\ & + \frac{\rho}{2}\|\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}\|_2^2 \end{aligned}$$

## ADMM

$$\mathbf{x}_1^{t+1} := \arg\min_{\mathbf{x}_1} \mathcal{L}_\rho(\mathbf{x}_1, \mathbf{x}_2^t, \boldsymbol{\lambda}^t), \quad \mathbf{x}_2^{t+1} := \arg\min_{\mathbf{x}_2} \mathcal{L}_\rho(\mathbf{x}_1^{t+1}, \mathbf{x}_2, \boldsymbol{\lambda}^t)$$

$$\boldsymbol{\lambda}^{t+1} := \boldsymbol{\lambda}^t + \rho(\mathbf{A}_1\mathbf{x}_1^{t+1} + \mathbf{A}_2\mathbf{x}_2^{t+1} - \mathbf{b})$$

# Solving Robust PCA

Remember that we want to solve

$$\begin{aligned} \min_{\mathbf{L},\mathbf{S}} \quad & \|\mathbf{L}\|_* + \mu\|\mathbf{S}\|_1 \\ \text{subject to} \quad & \mathbf{L} + \mathbf{S} = \mathbf{X} \end{aligned}$$

- Use ADMM (Alternating Direction Method of Multipliers)
- Natural splitting of objective into $\|\mathbf{L}\|_*$ and $\|\mathbf{S}\|_1$

# ADMM for RPCA

$$\min_{\mathbf{L},\mathbf{S}} \ \|\mathbf{L}\|_* + \mu\|\mathbf{S}\|_1, \quad \text{s.t.} \ \ \mathbf{L} + \mathbf{S} = \mathbf{X}$$

- Hence: $f_1(\mathbf{x}_1) = \|\mathbf{L}\|_*$ and $f_2(\mathbf{x}_2) = \mu\|\mathbf{S}\|_1$.
- Augmented Lagrangian (vec$(\cdot)$: vectorize matrix)

$$\begin{aligned}\mathcal{L}_\rho(\mathbf{L}, \mathbf{S}, \boldsymbol{\lambda}) = &\|\mathbf{L}\|_* + \mu\|\mathbf{S}\|_1 \\ &+ \langle \boldsymbol{\lambda}, \text{vec}(\mathbf{L} + \mathbf{S} - \mathbf{X})\rangle + \frac{\rho}{2}\|\mathbf{L} + \mathbf{S} - \mathbf{X}\|_F^2\end{aligned}$$

- ADMM updates for RPCA

$$\begin{aligned}\mathbf{L}^{t+1} &:= \arg\min_{\mathbf{L}} \mathcal{L}_\rho(\mathbf{L}, \mathbf{S}^t, \boldsymbol{\lambda}^t) \\ \mathbf{S}^{t+1} &:= \arg\min_{\mathbf{S}} \mathcal{L}_\rho(\mathbf{L}^{t+1}, \mathbf{S}, \boldsymbol{\lambda}^t) \\ \boldsymbol{\lambda}^{t+1} &:= \boldsymbol{\lambda}^t + \rho\, \text{vec}(\mathbf{L}^{t+1} + \mathbf{S}^{t+1} - \mathbf{X})\end{aligned}$$

# Individual Primal Minimization

Explicit minimization over $\mathbf{L}$ and $\mathbf{S}$ (mat: convert vector to matrix)

$$\arg\min_{\mathbf{L}} \mathcal{L}_\rho(\mathbf{L}, \mathbf{S}, \boldsymbol{\lambda}) = \mathcal{D}_{\rho^{-1}}(\mathbf{X} - \mathbf{S} - \rho^{-1}\mathsf{mat}(\boldsymbol{\lambda}))$$

$$\arg\min_{\mathbf{S}} \mathcal{L}_\rho(\mathbf{L}, \mathbf{S}, \boldsymbol{\lambda}) = \mathcal{S}_{\mu\rho^{-1}}(\mathbf{X} - \mathbf{L} - \rho^{-1}\mathsf{mat}(\boldsymbol{\lambda}))$$

## Shrinkage and Singular Value Thresholding

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

$$\mathcal{S}_\tau(\mathbf{X})\text{: apply } \mathcal{S}_\tau \text{ to each element}$$

$$\mathcal{D}_\tau(\mathbf{X}) = \mathbf{U}\mathcal{S}_\tau(\mathbf{\Sigma})\mathbf{V}^T$$

$$\text{where SVD } \mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

# ADMM for RPCA

Final algorithm to solve Robust PCA

---

**Algorithm 2** ADMM for RPCA

---

**Require:** data $\mathbf{X}$, regularizer $\mu$, ADMM parameter $\rho > 0$

  $\mathbf{S}^0 := 0$, $\boldsymbol{\lambda}^0 := 0$, $t := 0$

  **while** not converged **do**

    $\mathbf{L}^{t+1} := \mathcal{D}_{\rho^{-1}}(\mathbf{X} - \mathbf{S}^t - \rho^{-1}\mathsf{mat}(\boldsymbol{\lambda}^t))$

    $\mathbf{S}^{t+1} := \mathcal{S}_{\mu\rho^{-1}}(\mathbf{X} - \mathbf{L}^{t+1} - \rho^{-1}\mathsf{mat}(\boldsymbol{\lambda}^t))$

    $\mathsf{mat}(\boldsymbol{\lambda}^{t+1}) := \mathsf{mat}(\boldsymbol{\lambda}^t) + \rho(\mathbf{L}^{t+1} + \mathbf{S}^{t+1} - \mathbf{X})$

    $t := t + 1$

  **end while**

  **return** $\mathbf{L}^t$, $\mathbf{S}^t$

---

Section 3

# Applications of Robust PCA

# Audio Denoising

We separate a corrupted audio stream $\mathbf{X}$ into the low-rank music part $\mathbf{L}$ and sparse noise part $\mathbf{S}$.

## Audio sample

- Noisy music ($\mathbf{X}$): ▷ Play
- Low-Rank recovery ($\mathbf{L}$): ▷ Play
- Sparse noise ($\mathbf{S}$): ▷ Play
- Original music ($\mathbf{L}_0$): ▷ Play

# Application: Video Surveillance

**Data** Sequence of frames of a surveillance camera.

**Task** Estimate background model of the scene.

**Challenge**
- Presence of foreground objects
- Want to be flexible to changes in the scene (illumination, . . .).

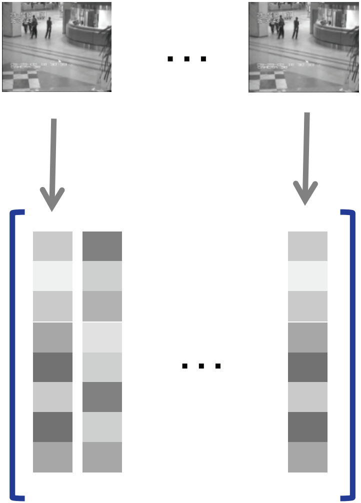
**Assumptions** Background variations are approximately low-rank, foreground sparse.

# Foreground Detection in Videos

- Interested in sparse part $\mathbf{S}$ of decomposition
- Stack all video frames into columns of $\mathbf{X}$
  - background: approximately low-rank (inter-frame correlations)
  - foreground: sparse part
  - use plain vanilla" Robust PCA!

# Construction of matrix $\mathbf{X}$

We stack the frames as columns of matrix $\mathbf{X}$



and solve **Separation**

$$\mathbf{X} = \mathbf{L_0} + \mathbf{S_0}$$

Dataset: 200 grayscale frames of resolution $176 \times 144$ giving a matrix $\mathbf{X} \in \mathbb{R}^{25,344 \times 200}$

# Sparse Foreground Detection

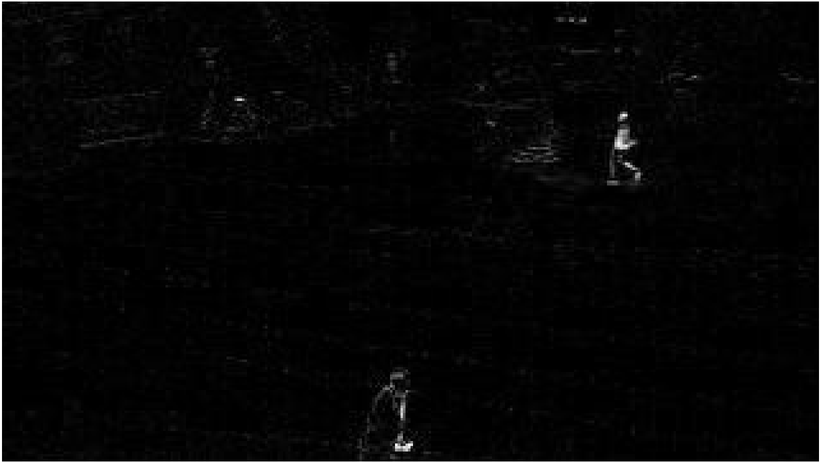


Figure: Sparse component $\mathbf{S}$: foreground

# Low-Rank Background

Figure: Low-rank component $\mathbf{L}$: background

# Application: Corrupted Images

- Corrupted data in computer vision often occurs when dealing with real images.
- Removing occlusion helps face detection.


Figure: Sunglasses or shadows occluding the face; old films

These are highly corrupted measurements: Gaussian assumption for noise is inadequate.

# Application: Repairing Vintage Movies

Original $D$


Repaired $A$


Corruptions

Frame 1

$480 \times 620$ pixels

# Application: Repairing Vintage Movies

Original $D$


Corruptions

Repaired $A$


Frame 2

# Application: Repairing Vintage Movies

Original $D$


Repaired $A$


Corruptions

Frame 3

# Application: Repairing Vintage Movies

Original $D$ | Repaired $A$


Corruptions

Frame 4

# Application: Repairing Vintage Movies

**Original** $D$


**Repaired** $A$


Corruptions

Frame 5

# Application: Repairing Vintage Movies

Original $D$ | Repaired $A$


Corruptions

Frame 6

# Application: Repairing Vintage Movies

Original $D$ | Repaired $A$


Corruptions

Frame 7

# Section 4

## Theory of Robust PCA

## Recovery of $\mathrm{L}_0$ and $\mathrm{S}_0$

*Goal:* Given $\mathbf{X}$ alone, find the true decomposition into low-rank and sparse matrix: $\mathbf{X} = \mathbf{L_0} + \mathbf{S_0}$.

*Theorem:*

Under broad conditions, the recovery via robust PCA is perfect, i.e. the solution $(\mathbf{L}^\star, \mathbf{S}^\star)$ obeys

$$\mathbf{L}^\star = \mathbf{L_0}, \quad \mathbf{S}^\star = \mathbf{S_0}$$

## Identifiability of $\mathrm{S}_0, \mathrm{L}_0$

- What conditions need to be met, s.t. we can possibly decompose $\mathbf{X}$ into a low-rank and sparse matrix?
- Even if $\mathbf{X}$ is the sum of a low-rank and a sparse matrix, this may be impossible.
- When $\mathbf{S_0}$ is low-rank or $\mathbf{L_0}$ is sparse, components may be confused for each other.

We will formalize this in the next slides.

# When can we hope to separate $\mathbf{X}$?

$\mathbf{X}$ cannot be both low-rank *and* sparse, e.g. not,

$$\mathbf{X} = \mathbf{e}_1 \mathbf{e}_n^T = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix},$$

where $\mathbf{e}_i$ denotes the $i$-th standard basis vector,

$$\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$$

# Condition on the Low-Rank Part $\mathbf{L}_0$

### $\mathbf{L}_0$ cannot be sparse

$$\mathbf{L_0} \in \mathbb{R}^{n \times n} = \mathbf{U}\Sigma\mathbf{V}^T = \sum_{1 \le i \le r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T, \quad r = \mathsf{rank}(\mathbf{L_0})$$

### Coherence condition:

$$\|\mathbf{U}^T \mathbf{e}_i\|^2 \le \frac{\nu r}{n}, \quad \|\mathbf{V}^T \mathbf{e}_i\|^2 \le \frac{\nu r}{n}, \quad |\mathbf{U}\mathbf{V}^T|_{ij}^2 \le \frac{\nu r}{n^2}$$

$\Longrightarrow$ Principal components must not be sparse (spiky).

# Identifiability: Main Result

## Theorem

If $\mathbf{L_0}$ satisfies the coherence condition for some $\nu > 0$, and if we additionally assume that

- $\mathbf{L_0} : n \times n$, of rank$(\mathbf{L_0}) \leq \rho_r n \nu^{-1} (\log n)^{-2}$
- $\mathbf{S_0} : n \times n$, random sparsity pattern of cardinality $m \leq \rho_s n^2$

Then with probability $1 - O(n^{-10})$, the minimizer of the PCP formulation for $\mathbf{X}$, with $\mu = \frac{1}{\sqrt{n}}$, recovers the true solution exactly:

$$\mathbf{L}^\star = \mathbf{L_0}, \quad \mathbf{S}^\star = \mathbf{S_0}$$

$\rho_s, \rho_r$ are positive constants.

# Exact Recovery

### Theorem

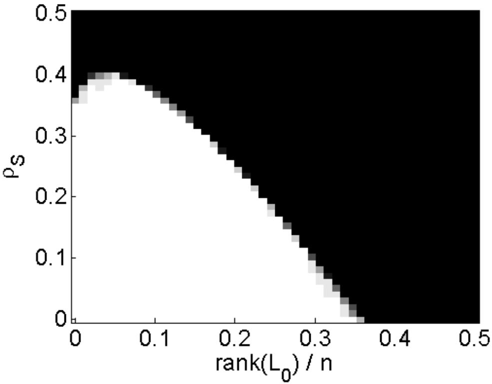
The same holds for rectangular matrices with $\mu = \frac{1}{\sqrt{\max(n,m)}}$

*Advantages of the Convex Relaxation*

- Convex (can use efficient algorithms)
- No tuning parameter!
- Exact recovery
- Recovery is independent of magnitudes of $\mathbf{L_0}, \mathbf{S_0}$

# Phase Transition in Recovery

We plot the fraction of correct recoveries of 10 trials, as a function of the rank of $\mathbf{L_0}$ (x-axis) and the sparsity of $\mathbf{S_0}$ (y-axis).




*This and all the following figures are from E. Candes et al. 2011.

- $\rho_s$ is the fraction of non-zero entries in $\mathbf{S_0}$

# Reading Material

Most material in the slides is from the following papers and books:

- **Robust PCA**: *E. Candes, X. Li, Y. Ma, J. Wright*: **Robust principal component analysis?**. Journal of ACM 58(1), 1-37, (2011).
- **Optimization**: *S. Boyd, L. Vandenberghe*: **Convex Optimization**. Cambridge Univ. Press, (2004). Mostly chapters 4 and 5. PDF file is freely available:

  http://www.stanford.edu/~boyd/cvxbook/
- **ADMM**: *S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein*: **Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers**. Foundations and Trends in Machine Learning: Vol. 3: No 1, pp 1-122, (2010).

  http://www.stanford.edu/~boyd/papers/admm_distr_stats.html