Dictionary Learning & Robust PCA

Lyu Xinrui, Hadi Daneshmand

June 8/9, 2017

Overview

Sparse Coding, Learning the Dictionary

- Advantage: we adapt a dictionary to signal characteristics
  $\Rightarrow$ same approximation error achievable with smaller L
- Disadvantage: we have to solve a matrix factorization problem

$$\underset{D \times N}{\boxed{\mathbf{X}}} \approx \underset{D \times L}{\boxed{\mathbf{U}}} \cdot \underset{L \times N}{\boxed{\mathbf{Z}}}$$

subject to sparsity constraint on $\mathbf{Z}$ and atom norm constraint on $\mathbf{U}$.

Matrix Factorization

$$(\mathbf{U}^\star, \mathbf{Z}^\star) \in \underset{\mathbf{U},\mathbf{Z}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{U} \cdot \mathbf{Z}\|_F^2$$

- Frobenius norm: $\|\mathbf{R}\|_F^2 = \sum_{i,j} r_{i,j}^2$ (sum of squared errors)
- Objective function not convex in both $\mathbf{U}$ and $\mathbf{Z}$ (local minima)
- But convex in either $\mathbf{U}$ or $\mathbf{Z}$ (unique minimum)

[1] This minimization is not convex due to sparsity.

Matrix Factorization

$$(\mathbf{U}^\star, \mathbf{Z}^\star) \in \underset{\mathbf{U},\mathbf{Z}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{U} \cdot \mathbf{Z}\|_F^2$$

- Frobenius norm: $\|\mathbf{R}\|_F^2 = \sum_{i,j} r_{i,j}^2$ (sum of squared errors)
- Objective function not convex in both $\mathbf{U}$ and $\mathbf{Z}$ (local minima)
- But convex in either $\mathbf{U}$ or $\mathbf{Z}$ (unique minimum)

Iterative greedy minimization

1. **Coding step**: $\mathbf{Z}^{t+1} \in \operatorname{argmin}_{\mathbf{Z}} \left\|\mathbf{X} - \mathbf{U}^t \cdot \mathbf{Z}\right\|_F^2$, subject to $\mathbf{Z}$ being sparse and fixed $\mathbf{U}$.[1]

[1] This minimization is not convex due to sparsity.

Matrix Factorization

$$(\mathbf{U}^\star, \mathbf{Z}^\star) \in \underset{\mathbf{U},\mathbf{Z}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{U} \cdot \mathbf{Z}\|_F^2$$

- Frobenius norm: $\|\mathbf{R}\|_F^2 = \sum_{i,j} r_{i,j}^2$ (sum of squared errors)
- Objective function not convex in both $\mathbf{U}$ and $\mathbf{Z}$ (local minima)
- But convex in either $\mathbf{U}$ or $\mathbf{Z}$ (unique minimum)

Iterative greedy minimization

1. **Coding step**: $\mathbf{Z}^{t+1} \in \operatorname{argmin}_{\mathbf{Z}} \left\|\mathbf{X} - \mathbf{U}^t \cdot \mathbf{Z}\right\|_F^2$, subject to $\mathbf{Z}$ being sparse and fixed $\mathbf{U}$.[1]
2. **Dictionary update step**: $\mathbf{U}^{t+1} \in \operatorname{argmin}_{\mathbf{U}} \left\|\mathbf{X} - \mathbf{U} \cdot \mathbf{Z}^{t+1}\right\|_F^2$, subject to $\|\mathbf{u}_l\|_2 = 1$ for all $l = 1, \ldots, L$ and fixed $\mathbf{Z}$.

[1] This minimization is not convex due to sparsity.

Coding Step

$$\mathbf{Z}^{t+1} \in \underset{\mathbf{Z}}{\operatorname{argmin}} \left\| \mathbf{X} - \mathbf{U}^t \mathbf{Z} \right\|_F^2$$

- Column separable residual: $\|\mathbf{R}\|_F^2 = \sum_{i,j} r_{i,j}^2 = \sum_j \|\mathbf{r}_j\|_2^2$
- N independent sparse coding steps: for all $n = 1, \dots N$

$$\begin{aligned} \mathbf{z}_n^{t+1} \quad &\in \quad \underset{\mathbf{z}}{\operatorname{argmin}} \|\mathbf{z}\|_0 \\ \text{s.t.} \quad & \quad \left\| \mathbf{x}_n - \mathbf{U}^t \mathbf{z} \right\|_2 \leq \sigma \cdot \|\mathbf{x}_n\|_2 \end{aligned}$$

Dictionary Update I

$$\mathbf{U}^{t+1} \in \underset{\mathbf{U}}{\operatorname{argmin}} \left\| \mathbf{X} - \mathbf{U}\mathbf{Z}^{t+1} \right\|_F^2$$

- Residual not separable in atoms (columns of $\mathbf{U}$)
- Approximation: update one atom at a time ($\forall l$)
  1. Set $\mathbf{U} = [\mathbf{u}_1^t \cdots \mathbf{u}_l \cdots \mathbf{u}_L^t]$, i.e. fix all atoms except $\mathbf{u}_l$.
  2. Isolate $\mathbf{R}_l^t$, the residual that is due to atom $\mathbf{u}_l$.
  3. Find $\mathbf{u}_l^*$ that minimizes $\mathbf{R}_l^t$, subject to $\|\mathbf{u}_l^*\|_2 = 1$.

Dictionary Update II

- Isolate $\mathbf{R}_l^t$: residual due to atom $\mathbf{u}_l$

$$\begin{aligned}
& \left\| \mathbf{X} - \left[\mathbf{u}_1^t \ \cdots \ \mathbf{u}_l \ \cdots \mathbf{u}_L^t \right] \cdot \mathbf{Z}^{t+1} \right\|_F^2 \\
= & \left\| \mathbf{X} - \left(\sum_{e \neq l} \mathbf{u}_e^t \left(\mathbf{z}_e^{t+1} \right)^\top + \mathbf{u}_l \left(\mathbf{z}_l^{t+1} \right)^\top \right) \right\|_F^2 \\
= & \left\| \mathbf{R}_l^t - \mathbf{u}_l \left(\mathbf{z}_l^{t+1} \right)^\top \right\|_F^2
\end{aligned}$$

  - $\mathbf{z}_l^\top$ is the l-th row of matrix $\mathbf{Z}$.

Dictionary Update III

Finding $\mathbf{u}_l^*$:

- $\mathbf{u}_l \left(\mathbf{z}_l^{t+1}\right)^\top$ is an outer product, i.e. a matrix
- Minimizing residual

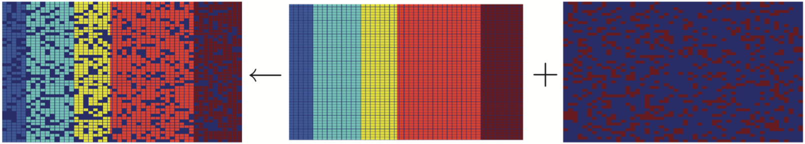
$$\left\|\mathbf{R}_l^t - \mathbf{u}_l \left(\mathbf{z}_l^{t+1}\right)^\top\right\|_F^2$$

  by approximating $\mathbf{R}_l^t$ with rank-1 matrix $\mathbf{u}_l \left(\mathbf{z}_l^{t+1}\right)^\top$
- "Approximately" achieved by SVD of $\mathbf{R}_l^t$:

$$\mathbf{R}_l^t = \tilde{\mathbf{U}}\boldsymbol{\Sigma}\tilde{\mathbf{M}}^\top = \sum_i \sigma_i \tilde{\mathbf{u}}_i \tilde{\mathbf{v}}_i^\top$$

  - $\mathbf{u}_l^* = \tilde{\mathbf{u}}_1$ is first left-singular vector.
  - $\|\mathbf{u}_l^*\|_2 = 1$ naturally satisfied.

A Separation Problem



$$\mathbf{X} = \mathbf{L} + \mathbf{S}$$

- Find $\mathbf{L}$ and $\mathbf{S}$ such that
- $\mathbf{L}$ should be a low-rank matrix
- Number of nonzero entries in $\mathbf{S}$ should be small

A Separation Problem

$$\begin{pmatrix}1 & 1 & 4 & 5 & 7\\ 1 & 2 & 4 & 4 & 7\\ 1 & 1 & 4 & 4 & 9\end{pmatrix} = \begin{pmatrix}1 & 1 & 4 & 4 & 7\\ 1 & 1 & 4 & 4 & 7\\ 1 & 1 & 4 & 4 & 7\end{pmatrix} + \begin{pmatrix}0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 2\end{pmatrix}$$

- $\mathbf{X} = \mathbf{L} + \mathbf{S}$
- rank$(\mathbf{L}) =$

A Separation Problem

$$\begin{pmatrix} 1 & 1 & 4 & 5 & 7 \\ 1 & 2 & 4 & 4 & 7 \\ 1 & 1 & 4 & 4 & 9 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 4 & 4 & 7 \\ 1 & 1 & 4 & 4 & 7 \\ 1 & 1 & 4 & 4 & 7 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

- $\mathbf{X} = \mathbf{L} + \mathbf{S}$
- $\mathrm{rank}(\mathbf{L}) = 1$
- $\|\mathbf{S}\|_0 =$

A Separation Problem

$$\begin{pmatrix} 1 & 1 & 4 & 5 & 7 \\ 1 & 2 & 4 & 4 & 7 \\ 1 & 1 & 4 & 4 & 9 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 4 & 4 & 7 \\ 1 & 1 & 4 & 4 & 7 \\ 1 & 1 & 4 & 4 & 7 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

- $\mathbf{X} = \mathbf{L} + \mathbf{S}$
- $\mathrm{rank}(\mathbf{L}) = 1$
- $\|\mathbf{S}\|_0 = 3$

PCA vs Robust PCA

$$\mathbf{X} \simeq \mathbf{L} + \mathbf{S}$$

- PCA:

$$\min_{\mathbf{L}} \|\mathbf{X} - \mathbf{L}\|_F$$
$$\mathsf{rank}(\mathbf{L}) \leq K$$

- RPCA:

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

Details of PCA

Seek the best rank-k estimate of $\mathbf{X}$

$$\begin{array}{ll} \text{minimize}_{\mathbf{L}} & \|\mathbf{X} - \mathbf{L}\|_{\text{Frob}}^2 \\ \text{subject to} & \text{rank}(\mathbf{L}) \leq k \end{array}$$

- Efficiently solved with SVD
- $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \sum_i \sigma_i u_i v_i^T$
- $\mathbf{L} = \sum_{i=1}^{k} \sigma_i u_i v_i^T$
- May fail with single corrupted observation

Pen & Paper: PCA vs RPCA

$$\mathbf{A} = \begin{pmatrix} 2 & 10 & 2 \\ 3 & 3 & 3 \\ 1 & 1 & 0 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} 2.01 & 1.99 & 2.05 \\ 3.03 & 3.01 & 2.99 \\ 1.1 & 0.99 & 0.98 \end{pmatrix}$$

Q1: Which matrix can be approximated better by a 1-rank matrix using PCA/RPCA?

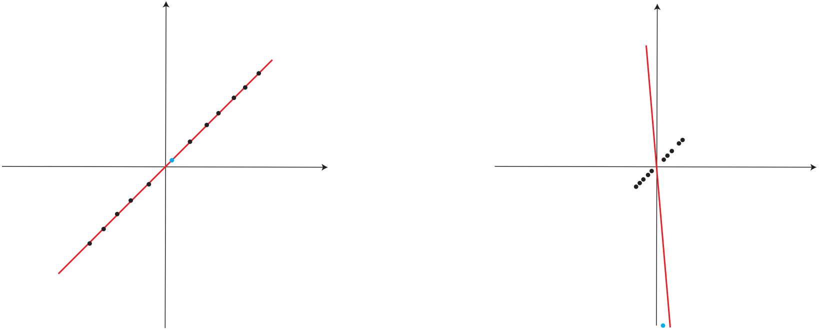
Pen & Paper: PCA vs RPCA

$$\mathbf{A} = \begin{pmatrix} 2 & 10 & 2 \\ 3 & 3 & 3 \\ 1 & 1 & 0 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} 2.01 & 1.99 & 2.05 \\ 3.03 & 3.01 & 2.99 \\ 1.1 & 0.99 & 0.98 \end{pmatrix}$$

Q1: Which matrix can be approximated better by a 1-rank matrix using PCA/RPCA?
solution: $\mathbf{A}$ with RPCA and $\mathbf{B}$ using PCA.

Problem with PCA

Very sensitive to outliers

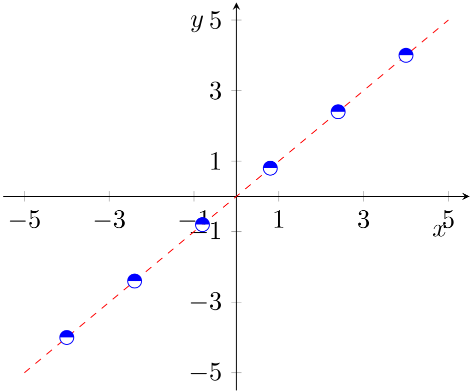


Single corrupted point completely changes principal component
$\Longrightarrow$ breakpoint for PCA is zero

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.

Pen& Paper

- Suggest a corrupted version of these points such that PCA can still estimate the principal component u_1.
- Suggest a corrupted version of these points such that Robust PCA estimates the principal component more reliably than PCA.



The Principal Component Pursuit (PCP) Problem

- RPCA:

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

- PCP:

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

where

$$\|\mathbf{S}\|_1 = \sum_{i,j} |s_{ij}|$$
$$\|\mathbf{L}\|_* = \sum_{i} \sigma_i(\mathbf{L})$$

Pen & Paper : Relaxing the L_0 Norm

- Why does minimizing $||.||_1$norm enforce minimizing the number of non-zero elements (cardinality)?
- A matrix with high $||.||_1$ norm and low cardinality?
- A matrix with high cardinality and low $||.||_1$ norm?

Why Nuclear Relaxation?

Consider vector of singular values $\sigma(\mathbf{L}) = \langle \sigma_1(\mathbf{L}), \ldots, \sigma_k(\mathbf{L}) \rangle$, then rank and nuclear norm can be interpreted as:

$$\begin{aligned} \mathsf{rank}(\mathbf{L}) &= \|\sigma(\mathbf{L})\|_0 \\ \|\mathbf{L}\|_* &= \sum_i \sigma_i(\mathbf{L}) = \sum_i |\sigma_i(\mathbf{L})| = \|\sigma(\mathbf{L})\|_1 \end{aligned}$$

Pen & Paper : Relaxing the Rank

- Why does minimizing the nuclear norm lead to minimizing the rank of the matrix?
- A matrix with high nuclear norm and low rank?
- A matrix with high rank and low nuclear norm?

Theory of PCP

- RPCA:

$$(\mathbf{L}_0, \mathbf{S}_0) = \arg\min_{\mathbf{L},\mathbf{S}} \mathsf{rank}(\mathbf{L}) + \mu \|\mathbf{S}\|_0$$
$$\text{subject to } \mathbf{X} = \mathbf{L} + \mathbf{S}$$

- PCP:

$$(\mathbf{L}_*, \mathbf{S}_*) = \arg\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \mu \|\mathbf{S}\|_1$$
$$\text{subject to } \mathbf{X} = \mathbf{L} + \mathbf{S}$$

Under some conditions

$$\mathbf{L}_0 = \mathbf{L}_*$$

Pen & Paper: Recovery Conditions for PCP

$$\mathbf{A} = \mathbf{L} + \mathbf{S}$$

$$\mathbf{L} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Q: Explain why the above choices for $\mathbf{L}$ is not good for recovery of $\mathbf{A}$.

Pen & Paper: Recovery Conditions for PCP

$$\mathbf{A} = \mathbf{L} + \mathbf{S}$$

$$\mathbf{L} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Q: Explain why the above choices for $\mathbf{L}$ is not good for recovery of $\mathbf{A}$.

Solution: The the low-rank component should not be sparse.

Pen & Paper: Recovery Conditions for PCP

$$\mathbf{A} = \mathbf{L} + \mathbf{S}$$

$$\mathbf{S} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Q: Explain why the above choices for $\mathbf{L}$ and $\mathbf{S}$ are not good for recovery of $\mathbf{A}$.

Pen & Paper: Recovery Conditions for PCP

$$\mathbf{A} = \mathbf{L} + \mathbf{S}$$

$$\mathbf{S} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Q: Explain why the above choices for $\mathbf{L}$ and $\mathbf{S}$ are not good for recovery of $\mathbf{A}$.

Solution: The sparse component should not be low-rank.

Alternating Direction Method of Multipliers

$$\begin{array}{lll} \text{minimize}_{\mathbf{x}_1,\mathbf{x}_2} & f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) & f_1, f_2 \text{ convex} \\ \text{subject to} & A_1\mathbf{x}_1 + A_2\mathbf{x}_2 = \mathbf{b} & \end{array}$$

Augmented Lagrangian

$$\begin{aligned} L_\rho(\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\nu}) \;=\; & f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) \;+ \boldsymbol{\nu}^T(A_1\mathbf{x}_1 + A_2\mathbf{x}_2 - \mathbf{b}) \\ & + \frac{\rho}{2}\|A_1\mathbf{x}_1 + A_2\mathbf{x}_2 - \mathbf{b}\|_2^2 \end{aligned}$$

ADMM

$$\begin{aligned} \mathbf{x}_1^{(t+1)} &:= \operatorname*{argmin}_{\mathbf{x}_1} L_\rho(\mathbf{x}_1, \mathbf{x}_2^{(t)}, \boldsymbol{\nu}^{(t)}) \\ \mathbf{x}_2^{(t+1)} &:= \operatorname*{argmin}_{\mathbf{x}_2} L_\rho(\mathbf{x}_1^{(t+1)}, \mathbf{x}_2, \boldsymbol{\nu}^{(t)}) \\ \boldsymbol{\nu}^{(t+1)} &:= \boldsymbol{\nu}^{(t)} + \rho(A_1\mathbf{x}_1^{(t+1)} + A_2\mathbf{x}_2^{(t+1)} - \mathbf{b}) \end{aligned}$$

ADMM special case

A special case is the constraint that the two blocks of variables are forced to be equal

$$\begin{array}{lll} \text{minimize}_{\mathbf{x}_1,\mathbf{x}_2} & f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) & f_1, f_2 \text{ convex} \\ \text{subject to} & \mathbf{x}_1 - \mathbf{x}_2 = 0 & \end{array}$$

Augmented Lagrangian

$$L_\rho(\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\nu}) \;=\; f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + \boldsymbol{\nu}^T(\mathbf{x}_1 - \mathbf{x}_2) + \frac{\rho}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2$$

ADMM

$$\begin{aligned} \mathbf{x}_1^{(t+1)} &:= \operatorname*{argmin}_{\mathbf{x}_1} L_\rho(\mathbf{x}_1, \mathbf{x}_2^{(t)}, \boldsymbol{\nu}^{(t)}) \\ \mathbf{x}_2^{(t+1)} &:= \operatorname*{argmin}_{\mathbf{x}_2} L_\rho(\mathbf{x}_1^{(t+1)}, \mathbf{x}_2, \boldsymbol{\nu}^{(t)}) \\ \boldsymbol{\nu}^{(t+1)} &:= \boldsymbol{\nu}^{(t)} + \rho\big(\mathbf{x}_1^{(t+1)} - \mathbf{x}_2^{(t+1)}\big) \end{aligned}$$

ADMM for LASSO

$$\begin{array}{lll} \text{minimize}_{\mathbf{x}_1,\mathbf{x}_2} & f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) & f_1, f_2 \text{ convex} \\ \text{subject to} & \mathbf{x}_1 - \mathbf{x}_2 = 0 & \end{array}$$

$$L_\rho(\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\nu}) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + \boldsymbol{\nu}^T(\mathbf{x}_1 - \mathbf{x}_2) + \frac{\rho}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2$$

ADMM

$$\begin{aligned} \mathbf{x}_1^{(t+1)} &:= \operatorname*{argmin}_{\mathbf{x}_1} L_\rho(\mathbf{x}_1, \mathbf{x}_2^{(t)}, \boldsymbol{\nu}^{(t)}) \\ \mathbf{x}_2^{(t+1)} &:= \operatorname*{argmin}_{\mathbf{x}_2} L_\rho(\mathbf{x}_1^{(t+1)}, \mathbf{x}_2, \boldsymbol{\nu}^{(t)}) \\ \boldsymbol{\nu}^{(t+1)} &:= \boldsymbol{\nu}^{(t)} + \rho\big(\mathbf{x}_1^{(t+1)} - \mathbf{x}_2^{(t+1)}\big) \end{aligned}$$

That can be easily adopted to LASSO:

$$f_1(\mathbf{x}_1) := \frac{1}{2}\|A\mathbf{x}_1 - \mathbf{b}\|_2^2, \qquad f_2(\mathbf{x}_2) := \lambda\|\mathbf{x}_2\|_1$$

ADMM for LASSO

$$\begin{array}{ll} \text{minimize}_{\mathbf{x}_1,\mathbf{x}_2} & \frac{1}{2}\|A\mathbf{x}_1 - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}_2\|_1 \\ \text{subject to} & \mathbf{x}_1 - \mathbf{x}_2 = 0 \end{array}$$

$$L_\rho(\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\nu}) = \frac{1}{2}\|A\mathbf{x}_1 - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}_2\|_1 + \boldsymbol{\nu}^T(\mathbf{x}_1 - \mathbf{x}_2) + \frac{\rho}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2$$

$$\begin{aligned} \mathbf{x}_1^{(t+1)} &:= \operatorname*{argmin}_{\mathbf{x}_1} L_\rho(\mathbf{x}_1, \mathbf{x}_2^{(t)}, \boldsymbol{\nu}^{(t)}) \\ \mathbf{x}_2^{(t+1)} &:= \operatorname*{argmin}_{\mathbf{x}_2} L_\rho(\mathbf{x}_1^{(t+1)}, \mathbf{x}_2, \boldsymbol{\nu}^{(t)}) \\ \boldsymbol{\nu}^{(t+1)} &:= \boldsymbol{\nu}^{(t)} + \rho\left(\mathbf{x}_1^{(t+1)} - \mathbf{x}_2^{(t+1)}\right) \end{aligned}$$

ADMM for LASSO

$$L_\rho(\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\nu}) = \frac{1}{2}\|A\mathbf{x}_1 - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}_2\|_1 + \boldsymbol{\nu}^T(\mathbf{x}_1 - \mathbf{x}_2) + \frac{\rho}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2$$

$$\begin{aligned}
\operatorname*{argmin}_{\mathbf{x}_1} L_\rho(\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\nu}) &= (A^T A + \rho I)^{-1}(A^T \mathbf{b} + \rho \mathbf{x}_2 - \mathbf{v}) \\
\operatorname*{argmin}_{\mathbf{x}_2} L_\rho(\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\nu}) &= S_{\lambda/\rho}(\mathbf{x}_1 + \boldsymbol{\nu}/\rho)
\end{aligned}$$

Shrinkage operator

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

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

ADMM for RPCA

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

- 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

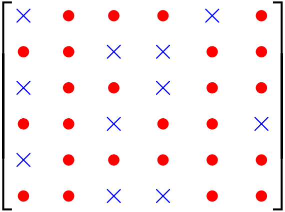
$$\begin{aligned}\mathbf{L}^{t+1} &:= \operatorname*{argmin}_{\mathbf{L}} \mathcal{L}_\rho(\mathbf{L},\mathbf{S}^t,\boldsymbol{\lambda}^t) \\ \mathbf{S}^{t+1} &:= \operatorname*{argmin}_{\mathbf{S}} \mathcal{L}_\rho(\mathbf{L}^{t+1},\mathbf{S},\boldsymbol{\lambda}^t) \\ \boldsymbol{\lambda}^{t+1} &:=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

Foreground Detection

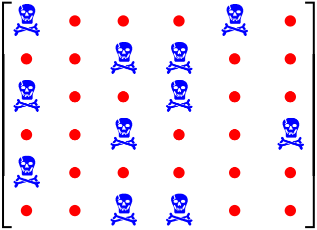
We see that each person in the video is considered as foreground, except the one in the rectangle. Can you explain why this is the case?

Missing vs Corrupted

MC: Missing entries



RPCA: Corrupted entries

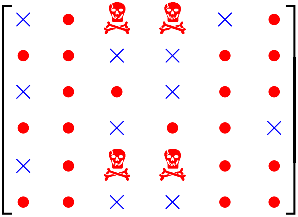


- ● Rating
- × Missing Value
- ☠ Corrupted Entry

Detecting corrupted entry is harder.

Shilling Attacks in Collaborative Filtering

Manipulations by malicious users giving lots of positive ratings for their own product and negative for competitors.



RPCA for Collaborative Filtering

But how to deal with missing values?

Instead of solving the problem

$$\begin{array}{ll} \text{minimize}_{\mathbf{L},\mathbf{S}} & \|\mathbf{L}\|_* + \mu\|\mathbf{S}\|_1 \\ \text{subject to} & \mathbf{L} + \mathbf{S} = \mathbf{X} \end{array}$$

we will solve the minimization with a different (weaker) constraint

$$\begin{array}{ll} \text{minimize}_{\mathbf{L},\mathbf{S}} & \|\mathbf{L}\|_* + \mu\|\mathbf{S}\|_1 \\ \text{subject to} & \mathbf{L}_{ij} + \mathbf{S}_{ij} = \mathbf{X}_{ij}, \quad \forall (i,j) \in \Omega_{obs}\,, \end{array}$$

where Ω_{obs} is the set of observed matrix entries.