Robust Principal Component Analysis

Shihua Zhang

Fall 2019

Overview

History

- Principal component analysis (PCA) was first proposed in 1901 and widely used as a tool to process data [Wold et al., 1987].

History

- Principal component analysis (PCA) was first proposed in 1901 and widely used as a tool to process data [Wold et al., 1987].
- Robust PCA (RPCA) was first raised to overcome the sensitivity of PCA to grossly corrupted points [Candès et al., 2011].

History

- Principal component analysis (PCA) was first proposed in 1901 and widely used as a tool to process data [Wold et al., 1987].
- Robust PCA (RPCA) was first raised to overcome the sensitivity of PCA to grossly corrupted points [Candès et al., 2011].
- The RPCA with an additional dense noise component was considered [Zhou et al., 2010].

History

- Principal component analysis (PCA) was first proposed in 1901 and widely used as a tool to process data [Wold et al., 1987].
- Robust PCA (RPCA) was first raised to overcome the sensitivity of PCA to grossly corrupted points [Candès et al., 2011].
- The RPCA with an additional dense noise component was considered [Zhou et al., 2010].
- A nonconvex model of RPCA was studied [Yi et al., 2016].

History

- Principal component analysis (PCA) was first proposed in 1901 and widely used as a tool to process data [Wold et al., 1987].
- Robust PCA (RPCA) was first raised to overcome the sensitivity of PCA to grossly corrupted points [Candès et al., 2011].
- The RPCA with an additional dense noise component was considered [Zhou et al., 2010].
- A nonconvex model of RPCA was studied [Yi et al., 2016].
- A nonconvex optimization formulation with manifold constraint for RPCA was considered [Zhang and Yang, 2018].

Motivations for PCA

- Processing high-dimensional data is computationally expensive and usually difficult.
- Most data are not randomly distributed over the high-dimensional space.
- They usually have patterns (structured, sparse or low-rank).
- In a machine learning view, low-rank models belong to unsupervised learning.
- PCA is a classical approach to achieve that.

PCA

PCA is a statistical procedure concerning with elucidating the covariance structure of a set of variables.

- finding the low-dimensional subspace that best approximates a given dataset.

PCA

PCA is a statistical procedure concerning with elucidating the covariance structure of a set of variables.

- finding the low-dimensional subspace that best approximates a given dataset.

What does PCA do?

PCA

PCA is a statistical procedure concerning with elucidating the covariance structure of a set of variables.

- finding the low-dimensional subspace that best approximates a given dataset.

What does PCA do?

Dimensionality Reduction!!!

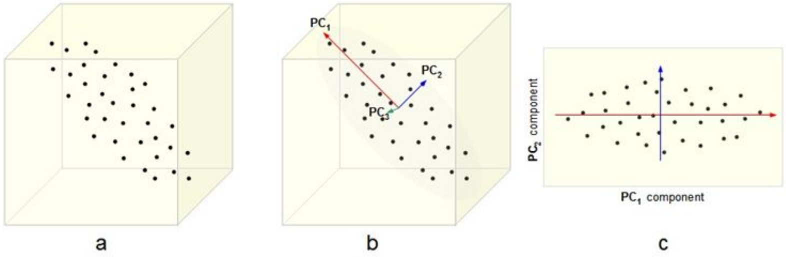
PCA

PCA is a statistical procedure concerning with elucidating the covariance structure of a set of variables.

- finding the low-dimensional subspace that best approximates a given dataset.

What does PCA do?

Dimensionality Reduction!!!

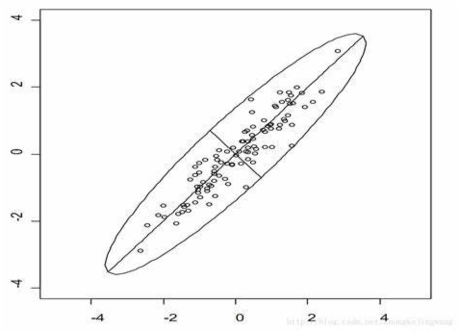


Goals of PCA

- Extract the most important information from the data table;
- Compress the size of the data set by keeping only this important information;
- Simplify the description of the data set;
- Analyze the structure of the observations and the variables.

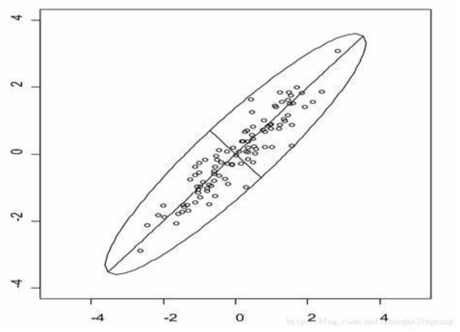
Two ways to derive PCA:

- Minimum projection distance of the samples
- Maximum projection variance of the samples



Two ways to derive PCA:

- Minimum projection distance of the samples
- Maximum projection variance of the samples



Proximity reconfiguration $\Leftrightarrow$ Maximum separability

Mathematical notation

Suppose:

- Given a data matrix $X = [x_1, x_2, ..., x_n] \in R^{m \times n}$ with n samples that are centered ($\sum_{i=1}^{n} x_i = 0$), where m is the dimension of data.
- Let $W = [w_1, w_2, ..., w_k]$, where $\{w_1, w_2, ..., w_k\}$ is a standard orthogonal basis for a low dimensional space ($||w_i||_2 = 1, w_i^\top w_j = 0$).
- The coordinate of the sample x_i projection in low dimensional space is $h_i \in R^k$ ($h_i = W^\top x_i$).
- $l_i = Wh_i$ is reconstructed data of x_i by h_i where W is a matrix with the standard orthogonal basis.

Minimum projection distance of the samples

Proximity reconfiguration:

The sample points are close enough to the hyperplane.

$$\min \sum_{i}^{n} \|l_i - x_i\|_2^2$$

Minimum projection distance of the samples

Proximity reconfiguration:

The sample points are close enough to the hyperplane.

$$\min \sum_{i}^{n} \|l_i - x_i\|_2^2$$

$$\begin{aligned}\sum_{i}^{n} \|l_i - x_i\|_2^2 &= \sum_{i}^{n} \|Wh_i - x_i\|_2^2 = \|X - WH\|_F^2 \\ &= \sum_{i=1}^{n} h_i^\top h_i - 2\sum_{i=1}^{n} h_i^\top W^\top x_i + const \\ &\propto -\mathrm{tr}(W^\top X X^\top W)\end{aligned}$$

Maximum projection variance of the samples

Maximum separability:

The projection of the sample points in the hyperplane is spread out as much as possible.

$$\max \sum_{i=1}^{n} h_i^\top h_i$$

Maximum projection variance of the samples

Maximum separability:

The projection of the sample points in the hyperplane is spread out as much as possible.

$$\max \sum_{i=1}^{n} h_i^\top h_i$$

$$\begin{aligned}\sum_{i=1}^{n} h_i^\top h_i &= \sum_{i=1}^{n} x_i^\top W W^\top x_i \\ &= \operatorname{tr}(W^\top X X^\top W)\end{aligned}$$

Optimization of PCA

$$\begin{aligned} \min_{W} & -\operatorname{tr}(W^\top X X^\top W) \\ \text{s.t.} & \quad W^\top W = I. \end{aligned}$$

Optimization of PCA

$$\min_{W} -\mathrm{tr}(W^\top XX^\top W)$$
$$\text{s.t.} \quad W^\top W = I.$$

Using the Lagrangian function,

$$J(W) = -\mathrm{tr}(W^\top XX^\top W) + \lambda(W^\top W - I)$$

Find the derivative of $J(W)$ with respect to W,

$$-XX^\top W + \lambda W = 0$$

Thus, $XX^\top W = \lambda W$

- W is made up of k eigenvectors of $XX^\top$.
- λ is a diagonal matrix made up of eigenvalues of $XX^\top$.

Principal components

Suppose that $\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3 \geqslant ... \geqslant 0$ are the eigenvalues of $XX^\top$ and the eigenvector of λ_i is w_i.

- According to the size of the eigenvalues, w_i is called the *i*th principal component.
- If a k-dimensional hyperplane projection is needed, select the first k principal components $W = (w_1, w_2, ..., w_k)$ as an orthonormal basis.
- Reduce the dimension of the original data set to the k dimension data set with the minimum projection distance through

$$h_i = W^\top x_i$$

.

Procedure of PCA

- Calculate the singular value decomposition (SVD) of X.

$$X = U\Sigma V^\top$$

- The columns of U are the principal components of X.
- Take the first k singular values of X, and suppress the rest.

$$\Sigma' = \begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_k & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

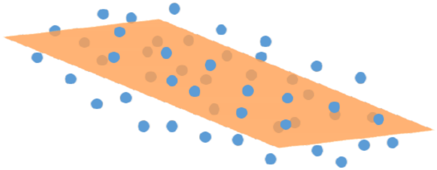
- Calculate $L = U\Sigma' V^\top$ as the reconstructed data.

PCA is a low-rank model

- Seeking k directions that explain the most variance of data

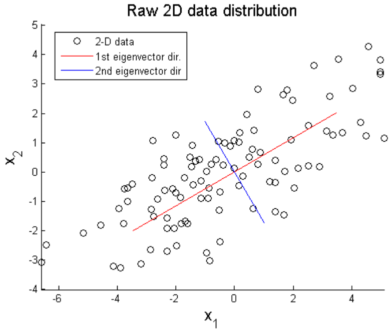
$$\min_{L:\text{rank}(L)\leqslant k} \|X - L\|_F$$

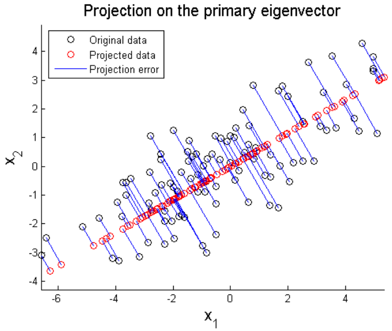
  where L is the reconstructed data set.
- The best rank-k approximation of X.



Application 1: Dimensionality reduction

In this example, PCA was implemented to project one hundred of 2-D data $X \in R^{2\times 100}$ onto 1-D space [Jang, 2014].





Application 2: Image compression

In this example, PCA was applied into the compression of a 512-by-512 grey-scale image $X \in R^{512\times512}$ [Richardson, 2009].

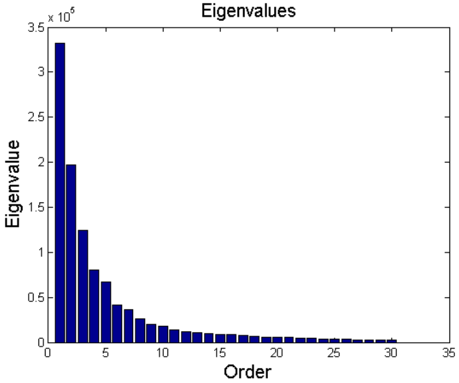
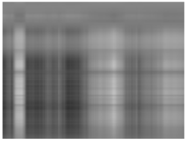


Figure: The original greyscale image and its ordered eigenvalues

Application 2: Image compression


(a) 1 principal component

(b) 5 principal component

(c) 9 principal component

(d) 13 principal component

(e) 17 principal component

(f) 21 principal component

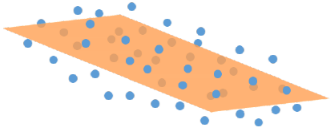
(g) 25 principal component

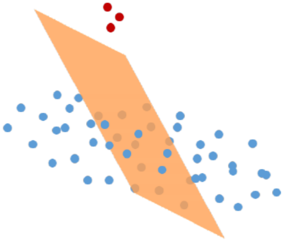
(h) 29 principal component

Figure: PCA results

Limitations of PCA

- PCA, though being so popular, has quite a few limitations.
- Outlier is one of the most important ones.
- What if some samples are corrupted (e.g. due to sensor errors/ attacks)?





Limitations of PCA

- In this example, only three outliers makes PCA fail.
- It is well known that PCA is very sensitive to outliers.
- Why is PCA sensitive to outliers? Think of PCA as an minimization problem!

$$\min_{L:\operatorname{rank}(L)\leqslant k} ||X - L||_F$$

- PCA minimizes the squared distance between the data (X) and the approximation (L)
- Squared distance amplifies outliers.

Limitations of PCA

- In this example, only three outliers makes PCA fail.
- It is well known that PCA is very sensitive to outliers.
- Why is PCA sensitive to outliers? Think of PCA as an minimization problem!

$$\min_{L:\operatorname{rank}(L)\leqslant k} ||X - L||_F$$

- PCA minimizes the squared distance between the data (X) and the approximation (L)
- Squared distance amplifies outliers.

Classical PCA fails even with a few outliers!!

Degenerate data

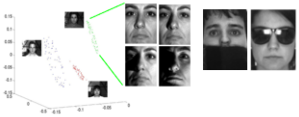
Lots of data have low-rank structure and abnormal points.

Examples of degenerate data:

Face images

Degeneracy: illumination models

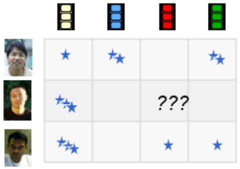
Errors: occlusion, corruption



Relevancy data

Degeneracy: user preferences co-predict

Errors: Missing rankings, manipulation



Video

Degeneracy: temporal, dynamic structures

Errors: anomalous events, mismatches…

Motivation of Robust PCA

How to handle this kind of data to improve the robustness of PCA?

Motivation of Robust PCA

How to handle this kind of data to improve the robustness of PCA?

If we can distinguish between normal and abnormal parts of the data, the model will be robust.

Idea of Robust PCA

- **Goal**: The goal is to find a low-rank representation for the data matrix just like PCA does and it is robust to outliers.

Idea of Robust PCA

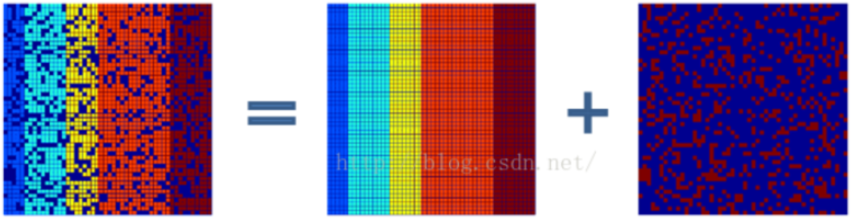
- **Goal**: The goal is to find a low-rank representation for the data matrix just like PCA does and it is robust to outliers.
- **Idea**: Decompose the data matrix $M \in R^{m \times n}$ into a low-rank matrix L and a sparse matrix S containing outliers.

$$M = L + S$$

Idea of Robust PCA

- **Goal**: The goal is to find a low-rank representation for the data matrix just like PCA does and it is robust to outliers.
- **Idea**: Decompose the data matrix $M \in R^{m \times n}$ into a low-rank matrix L and a sparse matrix S containing outliers.

$$M = L + S$$



Mathematical formulation

- Robust PCA (RPCA) can be stated as follows [Candès et al., 2011]:

$$\begin{aligned} \min_{L,S \in R^{m\times n}} \; & \operatorname{rank}(L) + \rho \|S\|_0 \\ \text{s.t.} \quad & L + S = M \end{aligned} \tag{1}$$

  where $\|S\|_0$ is called the l_0-norm of S and counts the number of nonzero entries of S, and $\rho > 0$ is a tradeoff parameter.

- For convenience, we call this problem (1) as the RPCA.

Connection between PCA and RPCA

- They both can be viewed as low-rank modeling techniques
  - PCA: (low-rank perspective)

$$\min_{L:\text{rank}(L)\leqslant r} \|M - L\|_F \tag{2}$$

Connection between PCA and RPCA

- They both can be viewed as low-rank modeling techniques
  - PCA: (low-rank perspective)

$$\min_{L:\mathrm{rank}(L)\leqslant r} ||M - L||_F \tag{2}$$

  - RPCA:

$$\min_{L,S\in R^{m\times n}} \mathrm{rank}(L) + \rho||S||_0 \tag{3}$$

$$\text{s.t.} \quad L + S = M \tag{4}$$

Connection between PCA and RPCA

- They both can be viewed as low-rank modeling techniques
  - PCA: (low-rank perspective)

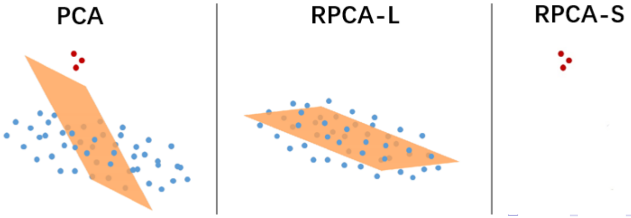
$$\min_{L:\text{rank}(L)\leqslant r} \|M - L\|_F \tag{2}$$

  - RPCA:

$$\min_{L,S\in R^{m\times n}} \text{rank}(L) + \rho\|S\|_0 \tag{3}$$

$$\text{s.t.} \quad L + S = M \tag{4}$$

- RPCA is more robust than PCA.



Ill-posed problems

Suppose the real data is made up of two parts superimposed:

$$M = L^o + S^o$$

such that

- $r = \text{rank}(L^o) \ll \min\{m, n\}$
- $s = ||S^o||_0 \ll mn$

Ill-posed problems

Suppose the real data is made up of two parts superimposed:

$$M = L^o + S^o$$

such that

- $r = \text{rank}(L^o) \ll \min\{m, n\}$
- $s = ||S^o||_0 \ll mn$

Ill-posed problems

$$\begin{cases} S^o : \text{low-rank} \\ L^o : \text{sparse} \end{cases} \Rightarrow \text{Recovery is hopeless}$$

Ill-posed problems

Assuming L^o is low-rank and S^o is sparse,

$$M = L^o + S^o$$

if $L^o = e_i e_i^\top$ is also sparse, M can be decomposed into:

$$M = 0 + (S^o + e_i e_i^\top)$$

if S^o is also low-rank, M can be decomposed into:

$$M = (S^o + L^o) + 0$$

Ill-posed problems

Assuming L^o is low-rank and S^o is sparse,

$$M = L^o + S^o$$

if $L^o = e_i e_i^\top$ is also sparse, M can be decomposed into:

$$M = 0 + (S^o + e_i e_i^\top)$$

if S^o is also low-rank, M can be decomposed into:

$$M = (S^o + L^o) + 0$$

How to avoid these ill-posed problems? Incoherence and randomness conditions [Chandrasekaran et al., 2011].

The incoherence conditions

Suppose the singular value decomposition (SVD) of L^o is

$$L^o = \Sigma_{i=1}^{r} \sigma_i u_i v_i^{\mathrm{T}}$$

where $U = [u_1, ..., u_r]$ and $V = [v_1, ..., v_r]$ are formed by its left- and right-singular vectors.

The incoherence conditions

Suppose the singular value decomposition (SVD) of L^o is

$$L^o = \Sigma_{i=1}^{r} \sigma_i u_i v_i^{\mathrm{T}}$$

where $U = [u_1, ..., u_r]$ and $V = [v_1, ..., v_r]$ are formed by its left- and right-singular vectors.

The incoherence conditions of L^o is as follows:

$$\max_i \|U^{\mathrm{T}} e_i\|^2 \leqslant \delta r/m \qquad \max_i \|V^{\mathrm{T}} e_i\|^2 \leqslant \delta r/n \tag{5a}$$

$$\|UV^{\mathrm{T}}\|_\infty \leqslant \sqrt{\frac{\delta r}{mn}} \tag{5b}$$

where $\|Z\|_\infty = max_{ij}|Z_{ij}|$ and e_i denotes the i-th unit vector.

Incoherence conditions $\Rightarrow$ L^o **is not sparse**

The randomness conditions

- For S^o, assuming that the set of indices for the nonzero entries is random and follows a uniform distribution among all the subsets of cardinality s.

Randomness conditions $\Rightarrow$ S^o **is not low-rank**

Difficulties of designing algorithms:

- rank$(*)$ and $||*||_0$ are both nonconvex. Attain an exact recovery of RPCA is numerically intractable and NP-hard.

Difficulties of designing algorithms:

- rank($*$) and $||*||_0$ are both nonconvex. Attain an exact recovery of RPCA is numerically intractable and NP-hard.
- How to find a practical equivalent of RPCA?

Why RPCA is NP-hard?

RPCA is NP-hard

Process of proof:

$$\begin{aligned} \textit{Rank norm} &\Leftrightarrow \textit{SAS problem} \\ &\Leftrightarrow \textit{Exact cover by } 3 \textit{ sets} \Leftarrow \text{NP-hard} \end{aligned}$$

RPCA is NP-hard

SAS problem (Sparse Approximate Solutions to Linear Systems) [Natarajan, 1995]:

Given a matrix $A \in R^{m \times n}$, a vector $b \in R^m$, and $\epsilon > 0$, compute a vector x satisfying $\|Ax - b\|_2 \leqslant \epsilon$ if such exists, such that x has the fewest number of nonzero entries overall such vectors.

When the matrix is restricted to a diagonal matrix, the minimum rank norm of the matrix is equivalent to making the diagonal elements the most sparse.

Rank norm $\Leftrightarrow$ *SAS problem*

RPCA is NP-hard

Exact cover by 3 sets:

Instance: A set S, and a collection C of 3-element subsets of S.
Question: Does C contains an exact cover for S, i.e., a sub-collection C' of C such that every element of occurs exactly once in C'?

SAS is NP-hard!

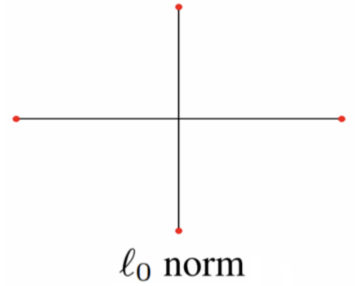
The proof is by reduction from the problem of **exact cover by 3 sets**, as in the proof of hardness for the problem of **"minimum weight solutions to linear systems"** [Garey and Johnson, 1990].

SAS problem $\Leftrightarrow$ Exact cover by 3 sets

Nonconvexity of RPCA:

Note: rank($*$) and $\|*\|_0$ are both nonconvex.
rank(L) is equivalent to l_0-norm of its singular values.

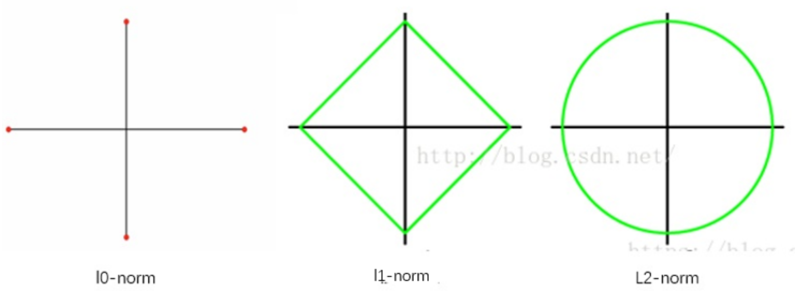
Let's consider $\|*\|_0$ (l_0-norm):



The goal of $\|S\|_0$ is to make sure S is sparse. To avoid its nonconvexity, we need to consider its convex relaxation.

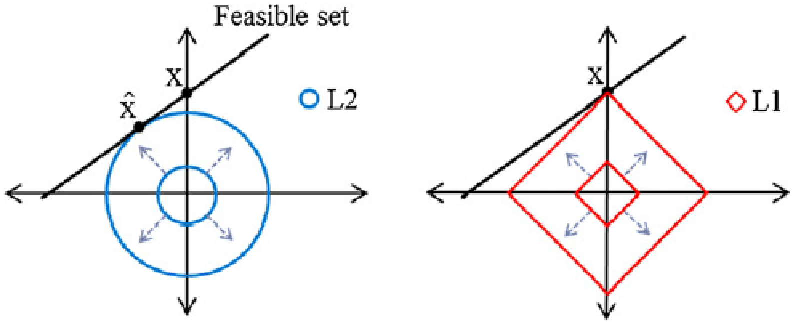
Convex relaxation of l_0-norm

l_0-norm and its usual convex relaxation:



Which is the better convex relaxation of l_0-norm?

Convex relaxation of l_0-norm



l_1-norm $\Rightarrow$ Sparsity.
l_2-norm $\nRightarrow$ Sparsity.
l_1-norm is the best convex relaxation of l_0-norm.

Convex relaxation of RPCA

$$\min \operatorname{rank}(L) \Leftrightarrow \text{Sparsity of the singular values of } L$$

Nuclear norm $||L||_* = \sum_i \sigma_i(L)$ is equivalent to l_1-norm of singular values of L. Thus,

$$\operatorname{rank}(L) \Rightarrow ||L||_*$$
$$||S||_0 \Rightarrow ||S||_1$$

Convex relaxation of RPCA

$$\min \operatorname{rank}(L) \Leftrightarrow \text{Sparsity of the singular values of } L$$

Nuclear norm $||L||_* = \sum_i \sigma_i(L)$ is equivalent to l_1-norm of singular values of L. Thus,

$$\operatorname{rank}(L) \Rightarrow ||L||_*$$
$$||S||_0 \Rightarrow ||S||_1$$

We can replace RPCA with the following form!!!

Robust Principal Component Pursuit (RPCP)

Under incoherence and randomness conditions, RPCA is equivalent to the following convex program with high probability:

$$\begin{aligned} \min_{L,S\in R^{m\times n}} & ||L||_* + \rho||S||_1 \\ \text{s.t.} \quad & L + S = M \end{aligned} \tag{6}$$

where $||L||_* = \sum_i \sigma_i(L)$, and $||S||_1 = \sum_{ij} |S_{ij}|$.

For convenience, this problem (6) is called **robust principal component pursuit (RPCP).**

Data missing

Assuming we only have observations on M_{ij} for some indices (i,j) form a subset Ω, then (6) can be reformulated as

$$\min_{L,S \in R^{m\times n}} \|L\|_* + \rho\|S\|_1 \text{ s.t. } P_\Omega(L+S-M) = 0 \tag{7}$$

The operator $P_\Omega : R^{m\times n} \to R^{m\times n}$ is defined as

$$P_\Omega(M)_{ij} = \begin{cases} M_{ij} & (i,j) \in \Omega \\ 0 & (i,j) \notin \Omega \end{cases}$$

Most algorithms in this course for solving RPCP can be used to solve (7) directly or with very little modification.

So, we only consider the situation with full data in this course.

Optimization of RPCP

Instead of optimize RPCA, we transform RPCA to RPCP. However, is optimization of RPCP feasible or can we get a perfect recovery?

Optimization of RPCP

Instead of optimize RPCA, we transform RPCA to RPCP. However, is optimization of RPCP feasible or can we get a perfect recovery?

YES!
RPCP can be recast as a Semi-definite Programming (SDP) [Chandrasekaran et al., 2011].

SDP for RPCP

- RPCP can be recast as a Semi-definite Programming (SDP).
- Assuming SVD of matrix L is:

$$L = U\Sigma V^\top$$

- Define $W_1 = U\Sigma U^\top$, $W_2 = V\Sigma V^\top$, $X' = \begin{pmatrix} W_1 & L \\ L^\top & W_2 \end{pmatrix}$
- Under above settings, the following properties is satisfied:
  - $\|X'\|_* = \|W_1\|_* + \|W_2\|_*$
  - $X' \succeq 0$ (positive semidefinite matrix):

$$X' = \begin{pmatrix} U \\ V \end{pmatrix} \Sigma \begin{pmatrix} U \\ V \end{pmatrix}^\top = \left(\begin{pmatrix} U \\ V \end{pmatrix} \sqrt{\Sigma}\right) \left(\begin{pmatrix} U \\ V \end{pmatrix} \sqrt{\Sigma}\right)^\top \succeq 0$$

SDP for RPCP

- For a symmetric matrix X', we have

$$\operatorname{trace}(X') = \|X'\|_*$$

- Therefore, the problem of minimizing the nuclear norm can be reduced [Recht et al., 2010]:

$$\begin{aligned} \min \operatorname{trace}(X') &= \min \operatorname{trace}(W_1) + \min \operatorname{trace}(W_2) \\ &= \min 2\|L\|_* \end{aligned}$$

SDP for RPCP

RPCP can be rewritten as:

$$\begin{aligned}
\min_{L,S,W_1,W_2,Z} \ & \rho 1_m^\top Z 1_n + \frac{1}{2}(\text{trace}(W_1) + \text{trace}(W_2)) \\
\text{s.t.} \quad & \begin{pmatrix} W_1 & L \\ L^\top & W_2 \end{pmatrix} \succeq 0 \\
& -Z_{ij} \leqslant S_{ij} \leqslant Z_{ij} \\
& L + S = M
\end{aligned} \tag{8}$$

- Now, the reduced form can be solved by SDP.
- Solving RPCP, we can get a global optimal solution.

Guarantee of recovery

Theorem 1

Suppose L^o is $n \times n$, obeys the incoherence condition, and that the support set of S^o is uniformly distributed among all sets of cardinality s. Then there is a numerical constant c such that with probability at least $1 - cn^{-10}$ (over the choice of support of S^o), Principal Component Pursuit with $\rho = 1/\sqrt{n}$ is exact, i.e. $L^* = L^o$ and $S^* = S^o$, provided that

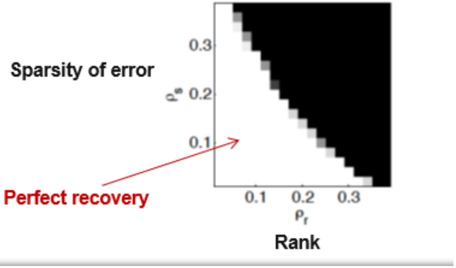
$$rank(L^o) \leqslant \rho_r n \delta^{-1} (\log n)^{-2} \quad and \quad s \leqslant \rho_s n^2$$

Above, ρ_r and ρ_s are positive numerical constants.
In the general rectangular case where L^o is $m \times n$, PCP with $\rho = \frac{1}{\sqrt{\max(m,n)}}$ succeeds with probability at least $1 - c * \max(m,n)^{-10}$, provided that $rank(L^o) \leqslant \rho_r * \min(m,n) * \delta^{-1} (\log(max(m,n)))^{-2}$ and $s \leqslant \rho_s mn$.

Analysis of RPCP

$$\min_{L,S\in R^{m\times n}} ||L||_* + \rho||S||_1 \quad \text{s.t.} \quad L + S = M$$

- **Empirical evidence**: probability of correct recovery vs rank and sparsity



Complexity of the interior point methods

- Since the problem above is a semidefinite programming, it can be solved by the interior point methods.
- The interior point methods are not directly amenable to large problems due to the $\mathcal{O}(n^6)$ complexity of computing a step direction.
- Thus, faster and less memory-required algorithms are needed.

Ideas for solving RPCP

Now, we recall the problem of RPCP again.

RPCP

$$\min_{L,S \in R^{m\times n}} ||L||_* + \rho ||S||_1$$
$$\text{s.t.} \quad L + S = M$$

Two ideas of designing algorithms

- Iterative shrinkage has been successfully applied to l_1 minimization [Cai et al., 2010] [Wright et al., 2009].
- The augmented Lagrangian alternating direction method minimizes the augmented Lagrangian function of model of RPCA.

The proximal gradient method (PGM)

Loss function of PGM:

$$\min_{L,S \in R^{m\times n}} ||L||_* + \rho||S||_1 + \frac{\beta}{2}||L+S-M||_F^2$$

Optimization procedure:

$$\begin{aligned} G^k &= \beta(L^k + S^k - M) \\ L^{k+1} &= \operatorname{argmin}_L ||L||_* + \frac{1}{2\tau}\left\|L - \left(L^k - \tau G^k\right)\right\|_F^2 \\ S^{k+1} &= \operatorname{argmin}_S \rho||S||_1 + \frac{1}{2\tau}\left\|S - \left(S^k - \tau G^k\right)\right\|_F^2 \end{aligned} \tag{9}$$

The proximal gradient method (PGM)

- G^k is the gradient of the quadratic penalty function in the loss function of PGM.
- $\tau > 0$ denotes a step size.
- The two subproblems in the optimization procedure of PGM both admit easy closed-form optimal solutions.

Subproblems (L-subproblem)

The solution of L-subproblem corresponds to the proximal mapping of the nuclear norm, which is given by

$$L^{k+1} = \mathcal{S}_\tau \left(L^k - \tau G^k \right)$$

where the matrix shrinkage operation is defined as

$$\mathcal{S}_\nu(Z) = U \operatorname{diag}\left((\sigma - \nu)_+ \right) V^\top \tag{10}$$

where $Z = U \operatorname{diag}(\sigma) V^\top$ is the SVD of Z, and $z_+ = \max(0, z)$.

Subproblems (S-subproblem)

The solution of S-subproblem correspond to the proximal mapping of the l_1-norm, which is given by

$$S^{k+1} = s_{\rho\tau}\left(S^k - \tau G^k\right)$$

where the vector shrinkage operation is defined as

$$[s_\nu(Z)]_{ij} = \text{sign}\left(Z_{ij}\right) \circ \max\{0, |Z_{ij}| - \nu\} \tag{11}$$

where sign(a) denotes the sign of a, and $\circ$ denotes the Hadamard product.

The accelerated PGM (APGM)

APGM incorporates Neterov's acceleration technique [Nesterov, 1983] and updates the variables with $t_{-1} = t_0 = 1$.

$$\begin{aligned}
\overline{L}^k &:= L^k + \frac{t_{k-1} - 1}{t_k}\left(L^k - L^{k-1}\right) \\
\overline{S}^k &:= S^k + \frac{t_{k-1} - 1}{t_k}\left(S^k - S^{k-1}\right) \\
\overline{G}^k &:= \beta\left(\overline{L}^k + \overline{S}^k - M\right) \\
L^{k+1} &:= \operatorname{argmin}_L \|L\|_* + \frac{1}{2\tau}\left\|L - \left(\overline{L}^k - \tau\overline{G}^k\right)\right\|_F^2 \\
S^{k+1} &:= \operatorname{argmin}_S \rho\|S\|_1 + \frac{1}{2\tau}\left\|S - \left(\overline{S}^k - \tau\overline{G}^k\right)\right\|_F^2 \\
t_{k+1} &:= \left(1 + \sqrt{1 + 4t_k^2}\right)/2
\end{aligned} \tag{12}$$

PGM and APGM

Nesterov's acceleration technique

For a certain algorithm, before each update, make the initial point shift so that the sublinear convergence algorithm is transformed into a better sublinear convergence algorithm.

- PGM: $\mathcal{O}(1/\epsilon)$
- APGM: $\mathcal{O}(1/\sqrt{\epsilon})$

Drawback of APGM

Comparison between the loss functions of PGM and RPCP:

$$\min_{L,S\in R^{m\times n}} ||L||_* + \rho||S||_1 + \frac{\beta}{2}||L+S-M||_F^2$$

$$\min_{L,S\in R^{m\times n}} ||L||_* + \rho||S||_1 \text{ s.t. } L+S=M$$

The loss function of PGM is equivalent to RPCP only when $\beta \leftarrow +\infty$. For any fixed $\beta > 0$ there is always a residual term which does not go to zero.

Drawback of APGM

Comparison between the loss functions of PGM and RPCP:

$$\min_{L,S \in R^{m\times n}} ||L||_* + \rho||S||_1 + \frac{\beta}{2}||L + S - M||_F^2$$

$$\min_{L,S \in R^{m\times n}} ||L||_* + \rho||S||_1 \text{ s.t. } L + S = M$$

The loss function of PGM is equivalent to RPCP only when $\beta \leftarrow +\infty$. For any fixed $\beta > 0$ there is always a residual term which does not go to zero.

How to remedy the residual term?

The augmented Lagrangian method (ALM)

ALM considers the augmented Lagrangian function of RPCP to transform RPCP to a unlimited problem.

Loss function of ALM:

$$\mathcal{L}_{\beta}(L, S; \Lambda) := \|L\|_* + \rho\|S\|_1 - \langle \Lambda, L + S - M \rangle + \frac{\beta}{2}\|L + S - M\|_F^2$$

where $\beta > 0$ is a vector of the penalty parameter and $\beta_0 > 0$, $\beta_{k+1} = t\beta_k$ ($t \geqslant 1$ is a constant). Λ is a Largrange multiplier to the linear equality constraint.

Subproblems of ALM

Optimization procedure:

$$\left(L^{k+1}, S^{k+1}\right) := \underset{L,S}{\operatorname{argmin}} \mathcal{L}_\beta \left(L, S; \Lambda^k, \beta_k\right) \tag{13a}$$

$$\Lambda^{k+1} := \Lambda^k - \beta_k \left(L^{k+1} + S^{k+1} - M\right) \tag{13b}$$

(L, S)-subproblem

$$L_{j+1}^{k+1} := \underset{L}{\operatorname{argmin}} \mathcal{L}_\beta \left(L, S_j^{k+1}; \Lambda^k, \beta_k\right)$$
$$S_{j+1}^{k+1} := \underset{S}{\operatorname{argmin}} \mathcal{L}_\beta \left(L, L_{j+1}^{k+1}; \Lambda^k, \beta_k\right)$$

Analysis of EALM

This ALM algorithm is an exact one (EALM).

- β: if β^k grows geometrically, EALM will converge Q-linearly; and if β^k grows faster, EALM will also converge faster.
- Problem of β: numerical tests show that for larger β^k, the iterative thresholding approach to solve the (L, S)-subproblem will converge slower.

Convergence analysis

Theorem 2

For EALM algorithm, any accumulation point (L^*, S^*) *of* (L^k, S^k) *is an optimal solution to RPCA problem and the convergence rate is at least* $\mathcal{O}(\beta_k^{-1})$ *in the sense that*

$$||\|L^k\|_* + \rho\|S^k\|_1 - f^*| = \mathcal{O}(\beta_{k-1}^{-1})$$

where f^* *is the optimal value of RPCA problem.*

Inexact ALM (IALM)

Optimization procedure of IALM:

$$
\begin{aligned}
L^{k+1} &:= \underset{L}{\operatorname{argmin}} \mathcal{L}_\beta \left(L, S^k; \Lambda^k, \beta_k\right) \\
S^{k+1} &:= \underset{S}{\operatorname{argmin}} \mathcal{L}_\beta \left(L, L^{k+1}; \Lambda^k, \beta_k\right) \\
\Lambda^{k+1} &:= \Lambda^k - \beta_k \left(L^{k+1} + S^{k+1} - M\right)
\end{aligned}
$$

EALM vs IALM

Difference:

(L, S)-subproblem:

- EALM: the iterative thresholding approach $\rightarrow$ Repeatedly,
- IALM: the iterative thresholding approach $\rightarrow$ One time.

- EALM guarantees convergence$(\mathcal{O}(1/\epsilon))$ but does not specify the rate of convergence for the inexact ALM method.
- The exact convergence rate of the inexact ALM method is difficult to obtain in theory, extensive numerical experiments have shown that for geometrically growing β^k, it still converges Q-linearly.

Convergence analysis

Theorem 3

For Algorithm IALM, if β_k *does not increase too rapidly, so that* $\sum_{k=1}^{\infty} \beta_k^{-2}\beta_{k+1} < \infty$ *and* $\lim_{k\to\infty} \beta_k(S^{k+1} - S^k) = 0$*, then* (L^k, S^k) *converges to an optimal solution* (L^*, S^*) *to the RPCA problem.*

Comparison between APG, EALM and IALM

m	algorithm	$\frac{\|\hat{A}-A^*\|_F}{\|A^*\|_F}$	rank$(\hat{A})$	$\|\hat{E}\|_0$	#SVD	time (s)
rank$(A^*) = 0.05\,m$, $\|E^*\|_0 = 0.05\,m^2$						
500	APG	1.12e-5	25	12542	127	11.01
	EALM	3.99e-7	25	12499	28	4.08
	IALM	5.21e-7	25	12499	20	1.72
800	APG	9.84e-6	40	32092	126	37.21
	EALM	1.47e-7	40	32002	29	18.59
	IALM	3.29e-7	40	31999	21	5.87
1000	APG	8.79e-6	50	50082	126	57.62
	EALM	7.85e-8	50	50000	29	33.28
	IALM	2.67e-7	50	49999	22	10.13
1500	APG	7.16e-6	75	112659	126	163.80
	EALM	7.55e-8	75	112500	29	104.97
	IALM	1.86e-7	75	112500	22	30.80
2000	APG	6.27e-6	100	200243	126	353.63
	EALM	4.61e-8	100	200000	30	243.64
	IALM	9.54e-8	100	200000	22	68.69
3000	APG	5.20e-6	150	450411	126	1106.22
	EALM	4.39e-8	150	449998	30	764.66
	IALM	1.49e-7	150	449993	22	212.34

where A (L) represents the lower rank part of RPCP and E (S) represents the sparsity part.

Noise assumption

Noise assumption

For observed $m \times n$ matrix M, assume $M = L^o + S^o + N^o$, where L^o is a low-rank matrix, S^o is a sparse matrix and N^o is the noise.

- Why this assumption is needed?
- In reality, the observations are often corrupted by noise.
- Face recognition
  - The human face is not a strictly convex and Lambertian surface hence small perturbation.
- Ranking and collaborative filtering
  - User's ratings could be noisy because of the lack of control in the data collection process.

SPCP

One variant of RPCP deals with an additional dense noise component N^o such that $||N^o|| \leqslant \sigma$ for some noise level $\sigma > 0$

$$M = L^o + S^o + N^o$$

Under this formulation, the **stable principal component pursuit (SPCP)** is defined as:

$$\begin{aligned} \min_{L,S \in R^{m\times n}} \ & ||L||_* + \rho ||S||_1 \\ \text{s.t.} \quad & ||L + S - M||_F \leqslant \sigma \end{aligned} \tag{14}$$

Data missing

Assuming we only have observations on M_{ij} for some indices (i, j) form a subset Ω, then (14) can be reformulated as

$$\min_{L,S \in R^{m \times n}} ||L||_* + \rho ||S||_1 \text{ s.t. } ||P_\Omega(L + S - M)||_F \leqslant \sigma \tag{15}$$

The operator $P_\Omega : R^{m \times n} \to R^{m \times n}$ is defined as

$$P_\Omega(M)_{ij} = \begin{cases} M_{ij} & (i,j) \in \Omega \\ 0 & (i,j) \notin \Omega \end{cases}$$

Most algorithms in this course for solving RPCP can be used to solve the condition of data missing directly or with very little modification.

Guarantee of recovery

Theorem 4

Suppose that L^o and S^o obey incoherence and randonness conditions, Then if L^o and S^o satisfy

$$\mathrm{rank}(L^o) \leqslant \rho_r n \delta^{-1} (\log n)^{-2} \quad \text{and} \quad s \leqslant \rho_s n^2$$

with $\rho_r, \rho_s > 0$ being sufficiently small numerical constants, with high probability in the support of S^o, for any N^o with $\|N^o\|_F \leqslant \sigma$ the solution (L^*, S^*) to the convex program (14) satisfies

$$\|L^o - L^*\|_F^2 + \|S^o - S^*\|_F^2 \leqslant C n^2 \sigma^2$$

where C is a numerical constant.

Algorithms for SPCP

- The alternating splitting augmented Lagrangian method (ASALM) [Tao and Yuan, 2011]
- The alternating direction method with increasing penalty (ADMIP) [Aybat and Iyengar, 2015]
- Three-block ADMM [Lin et al., 2018]

ASALM

Optimized problem (ASALM) :

$$\begin{array}{ll} \min_{L,S,N\in\mathbb{R}^{m\times n}} & \|L\|_* + \rho\|S\|_1 \\ \text{s.t.} & L+S+N=M,\ \|N\|_F \leqslant \sigma \end{array} \tag{16}$$

The augmented Lagrangian function ASALM is

$$\begin{array}{l} \mathcal{L}_\beta(L,S,N;\Lambda) := \|L\|_* + \rho\|S\|_1 + \mathbf{1}\left(N \| \|N\|_F \leqslant \sigma\right) \\ -\langle \Lambda, L+S+N-M\rangle + \frac{\beta}{2}\|L+S+N-M\|_F^2 \end{array}$$

where $\mathbf{1}(N|\mathcal{N})$ denotes the indicator function of the set $N \in \mathcal{N}$ e.g.

$$\mathbf{1}(N|\mathcal{N}) = \begin{cases} 0 & N \in \mathcal{N} \\ \infty & N \notin \mathcal{N} \end{cases}$$

ASALM

Optimization procedure:

$$
\begin{aligned}
L^{k+1} &:= \operatorname{argmin}_L \mathcal{L}_\beta\left(L, S^k, N^k; \Lambda^k\right) \\
S^{k+1} &:= \operatorname{argmin}_S \mathcal{L}_\beta\left(L^{k+1}, S, N^k; \Lambda^k\right) \\
N^{k+1} &:= \operatorname{argmin}_{N\in\mathcal{N}} \mathcal{L}_\beta\left(L^{k+1}, S^{k+1}, N; \Lambda^k\right) \\
\Lambda^{k+1} &:= \Lambda^k - \beta\left(L^{k+1} + S^{k+1} + N^{k+1} - M\right)
\end{aligned} \tag{17}
$$

ASALM (Subproblems:)

Three subproblems all have closed-form solutions:

- L-subproblem corresponds to the proximal mapping of $||L||_*$
- S-subproblem corresponds to proximal mapping of $||S||_1$
- N-subproblem corresponds to projection onto the set $(N|||N||_F \leqslant \sigma)$

Convergence analysis

Theorem 5

Let the sequence $\{L^k\}, \{S^k\}, \{N^k\}, \{\Lambda_k\}$ be generated by the proposed ASALM. Then, in the implementation of the proposed ASALM, if the penalty parameter β is allowed to vary dynamically and the sequence $\{\beta_k\}$ is chosen appropriately such that $\sum_{k=1}^{\infty} \beta_k^{-2}\beta_{k+1} < \infty$, then the sequences $\{L^k\}, \{S^k\}, \{N^k\}, \{\Lambda_k\}$ generated by the proposed ASALM are all bounded.

Problem of ASALM:

- In practice, this three-block ADMM usually works very well.
- However, it was later discovered that the ADMM with more than two-block variables is not necessarily convergent in general.

How to turn three to two?

ADMIP

Solving the following equivalent formulation for SPCP using the variable splitting trick:

$$\min_{\hat{L}, L, S \in \mathbb{R}^{m \times n}} \|\hat{L}\|_* + \rho \|S\|_1 \text{ s.t. } (L, S) \in \chi, \; L = \hat{L} \tag{18}$$

The augmented Lagrangian function of ADMIP can be written as

$$\mathcal{L}_\beta(\hat{L}, L, S; \Lambda) := \|\hat{L}\|_* + \rho \|S\|_1 - \langle \Lambda, \hat{L} - L \rangle + \frac{\beta}{2} \|\hat{L} - L\|_F^2$$

where β is a nondecreasing penalty parameter sequence $\left\{\beta^k\right\}_{k \in \mathbb{Z}_+}$ and $\sum_k \left(\beta^k\right)^{-1} = +\infty$.

ADMIP

ADMIP updates the variables as follows:

$$\begin{aligned} \hat{L}^{k+1} &:= \underset{\hat{L}}{\operatorname{argmin}} \mathcal{L}_{\beta^k}\left(\hat{L}, L^k, S^k; \Lambda^k\right) \\ \left(L^{k+1}, S^{k+1}\right) &:= \underset{(L,S)\in\chi}{\operatorname{argmin}} \mathcal{L}_{\beta^k}\left(\hat{L}^{k+1}, L, S; \Lambda^k\right) \\ \Lambda^{k+1} &:= \Lambda^k - \beta^k\left(\hat{L}^{k+1} - L^{k+1}\right) \end{aligned} \tag{19}$$

Convergence analysis

Theorem 6

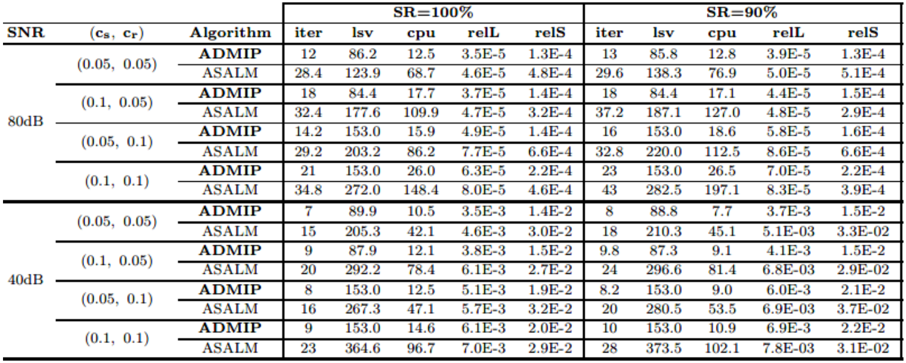
Let the sequence $\{L^k\}$, $\{\hat{L}^k\}$, $\{S^k\}$, $\{\Lambda_k\}$ *be generated by the proposed ADMIP. Then, in the implementation of the proposed ADMIP, then the limit of* $\lim_{k\in Z_+} L^k$ *and* $\lim_{k\in Z_+} \hat{L}^k$ *exists and* $L^* = \lim_{k\in Z_+} L^k = \lim_{k\in Z_+} \hat{L}^k$. $S^* = \lim_{k\in Z_+}$ *exists.* $(L^*, \hat{L}^*, S^*)$ *is a saddle point of the Lagrangian function* $\mathcal{L}(L^*, \hat{L}^*, S^*)$

ADMIP

The main advantages of adopting an increasing sequence of penalties are as follows:

- For different data, we are no need to search a fixed parameter β^*.
- The algorithm is likely to achieve primal feasibility faster.
- The complexity of initial (transient) iterations can be controlled through controlling $\{\beta^k\}$. To compute the solution to the (L, S)-subproblem in (19), one does not need to compute singular values smaller than β^{-1}.

Comparison between ADMIP and ASALM



			SR=100%					SR=90%				
SNR	**(c_s, c_r)**	**Algorithm**	**iter**	**lsv**	**cpu**	**relL**	**relS**	**iter**	**lsv**	**cpu**	**relL**	**relS**
80dB	(0.05, 0.05)	**ADMIP**	12	86.2	12.5	3.5E-5	1.3E-4	13	85.8	12.8	3.9E-5	1.3E-4
		ASALM	28.4	123.9	68.7	4.6E-5	4.8E-4	29.6	138.3	76.9	5.0E-5	5.1E-4
	(0.1, 0.05)	**ADMIP**	18	84.4	17.7	3.7E-5	1.4E-4	18	84.4	17.1	4.4E-5	1.5E-4
		ASALM	32.4	177.6	109.9	4.7E-5	3.2E-4	37.2	187.1	127.0	4.8E-5	2.9E-4
	(0.05, 0.1)	**ADMIP**	14.2	153.0	15.9	4.9E-5	1.4E-4	16	153.0	18.6	5.8E-5	1.6E-4
		ASALM	29.2	203.2	86.2	7.7E-5	6.6E-4	32.8	220.0	112.5	8.6E-5	6.6E-4
	(0.1, 0.1)	**ADMIP**	21	153.0	26.0	6.3E-5	2.2E-4	23	153.0	26.5	7.0E-5	2.2E-4
		ASALM	34.8	272.0	148.4	8.0E-5	4.6E-4	43	282.5	197.1	8.3E-5	3.9E-4
40dB	(0.05, 0.05)	**ADMIP**	7	89.9	10.5	3.5E-3	1.4E-2	8	88.8	7.7	3.7E-3	1.5E-2
		ASALM	15	205.3	42.1	4.6E-3	3.0E-2	18	210.3	45.1	5.1E-03	3.3E-02
	(0.1, 0.05)	**ADMIP**	9	87.9	12.1	3.8E-3	1.5E-2	9.8	87.3	9.1	4.1E-3	1.5E-2
		ASALM	20	292.2	78.4	6.1E-3	2.7E-2	24	296.6	81.4	6.8E-03	2.9E-02
	(0.05, 0.1)	**ADMIP**	8	153.0	12.5	5.1E-3	1.9E-2	8.2	153.0	9.0	6.0E-3	2.1E-2
		ASALM	16	267.3	47.1	5.7E-3	3.2E-2	20	280.5	53.5	6.9E-03	3.7E-02
	(0.1, 0.1)	**ADMIP**	9	153.0	14.6	6.1E-3	2.0E-2	10	153.0	10.9	6.9E-3	2.2E-2
		ASALM	23	364.6	96.7	7.0E-3	2.9E-2	28	373.5	102.1	7.8E-03	3.1E-02

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\mathrm{E}\left[\|\pi_\Omega(L^0+S^0)\|_F^2\right]}{\mathrm{E}\left[\|\pi_\Omega(N^0)\|_F^2\right]}\right) \qquad \mathrm{SR} = \frac{|\Omega|}{n^2}$$

Three-block ADMM

The penalty formulation of the SPCP problem is considered [Lin et al., 2018]. It is equivalent to solving SPCP for certain noise level $\sigma > 0$:

$$\begin{array}{ll} \min & \|L\|_* + \rho\|S\|_1 + \mu\|N\|_F^2 \\ \text{s.t.} & L + S + N = M \end{array} \tag{20}$$

where $\rho > 0$ is the sparsity tradeoff parameter and $\mu > 0$ is a suitable penalty parameter depending on the noise level $\sigma > 0$. where the augmented Lagrangian function is :

$$\begin{array}{l} \mathcal{L}_\beta(L, S, N; \Lambda) := \|L\|_* + \rho\|S\|_1 + \mu\|N\|_F^2 \\ -\langle \Lambda, L + S + N - M \rangle + \frac{\beta}{2}\|L + S + N - M\|_F^2 \end{array}$$

Three-block ADMM

Optimization procedure :

$$\begin{aligned} L^{k+1} &:= \operatorname{argmin}_L \mathcal{L}_\beta \left(L, S^k, N^k; \Lambda^k\right) \\ S^{k+1} &:= \operatorname{argmin}_S \mathcal{L}_\beta \left(L^{k+1}, S, N^k; \Lambda^k\right) \\ N^{k+1} &:= \operatorname{argmin}_N \mathcal{L}_\beta \left(L^{k+1}, S^{k+1}, N; \Lambda^k\right) \\ \Lambda^{k+1} &:= \Lambda^k - \beta \left(L^{k+1} + S^{k+1} + N^{k+1} - M\right) \end{aligned} \tag{21}$$

- L-subproblem corresponds to the proximal mapping of $||L||_*$
- S-subproblem corresponds to proximal mapping of $||S||_1$
- N-subproblem admits a very easy analytical solution

Analysis of three-block ADMM

Three-block ADMM is equivalent to solving SPCP for certain noise level $\sigma > 0$. Although the rate of convergence and computational complexity are not reduced, the authors showed that solving three-bloock ADMM globally converges for any penalty parameter $\beta > 0$.

Insufficiency of convex algorithms

Although the convex optimization algorithm of RPCA is easy to have good convergence, it is inevitable to carry out SVD in each iteration.

SVD greatly limits the speed of convex optimization algorithm.

How to avoid SVD decomposition?

Nonconvex relaxation of RPCA

The reason why it is difficult to avoid SVD is that the nuclear norm of the low-rank part of the data requires little.

We can avoid SVD by the following transform:

$$\begin{aligned} & \min_{U,V,S} \left\| UV^\top + S - M \right\|_F^2 \\ \text{s.t.} \quad & \operatorname{rank}(U) = \operatorname{rank}(V) \leqslant \tau_r, \ \|S\|_0 \leqslant \tau_s \end{aligned}$$

- Low-rank Matrix Fitting (LMafit) [Shen et al., 2014].

The nonconvex relaxations/variants

The nonconvex relaxations :

In order to avoid SVD calculations, we factorize the low rank matrix $L \in R^{m\times n}$ as a product of two low-rank matrices, i.e., factorize $L = UV^\top$, where $U \in R^{m\times r}$, $N \in R^{n\times r}$ and $r \ll \min\{m, n\}$ such that r is an upper bound on rank(L^o).

$$\begin{array}{rc} & \min_{U,V,S} \left\| UV^\top + S - M \right\|_F^2 \\ \text{s.t.} & \text{rank}(U) = \text{rank}(V) \leqslant \tau_r, \quad \|S\|_0 \leqslant \tau_s \end{array} \tag{22}$$

where τ_r are given parameters to controls the rank of U and V and τ_s controls the sparsity of S.

Algorithms of nonconvex variant

Shen et al. (2014) proposed a simple reformulation (LMafit):

Problem :

$$\min_{U,V} \left\| UV^\top - M \right\|_1 \tag{23}$$

Rewrite the above problem :

$$\min \|S\|_1 \text{ s.t. } S + UV^\top = M \tag{24}$$

By associating a Lagrange multiplier Λ to the constraint, the augmented Lagrangian function for (24) can be written as

$$\begin{aligned}\mathcal{L}_\beta(U, V, S; \Lambda) :=& \|S\|_1 - \left\langle \Lambda, UV^\top + S - M \right\rangle \\ &+ \frac{\beta}{2} \left\| UV^\top + S - M \right\|_F^2\end{aligned}$$

LMafit

Optimization procedure:

$$\begin{aligned} U^{k+1} &:= \operatorname{argmin}_U \mathcal{L}_\beta\left(U, V^k, S^k; \Lambda^k\right) \\ V^{k+1} &:= \operatorname{argmin}_V \mathcal{L}_\beta\left(U^{k+1}, V, S^k; \Lambda^k\right) \\ S^{k+1} &:= \operatorname{argmin}_S \mathcal{L}_\beta\left(U^{k+1}, V^{k+1}, S; \Lambda^k\right) \\ \Lambda^{k+1} &:= \Lambda^k - \beta\left(U^{k+1}V^{k+1} + S^{k+1} - M\right) \end{aligned} \tag{25}$$

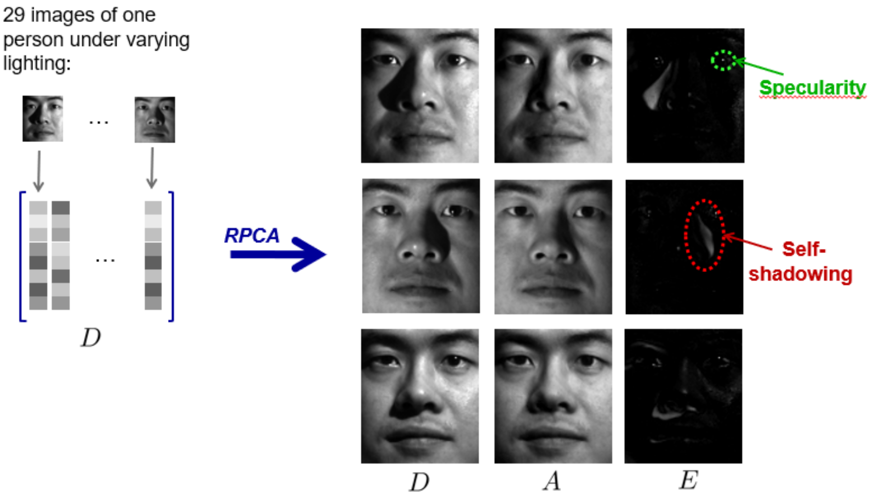
Convergence rates of different algorithms

Algorithm	Convex vs Nonconvex	Convergence rate
PGM	Convex	$1/\epsilon$
APGM	Convex	$1/\epsilon^2$
IALM	Convex	Convergence
ASALM	Convex	Convergence unclear
ADMIP	Convex	$1/\epsilon$
3-block ADMM	Convex	Convergence
LMafit	Nonconvex	Convergence

Applications of RPCA

- Video surveillance
- Face recognition
- Latent semantic indexing
- Ranking and collaborative filtering

Face recognition

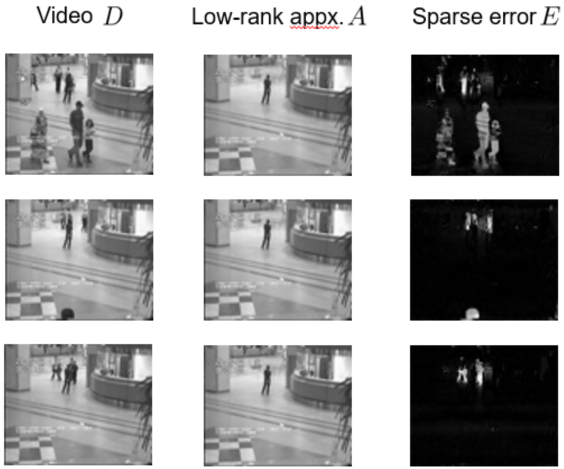


Video surveillance

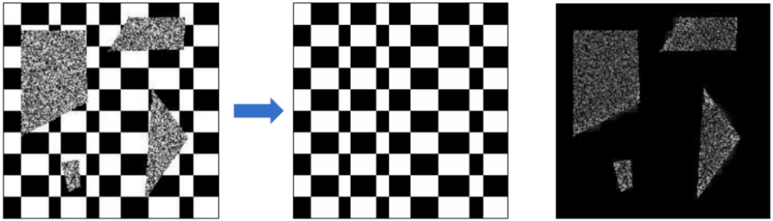
Static camera surveillance video

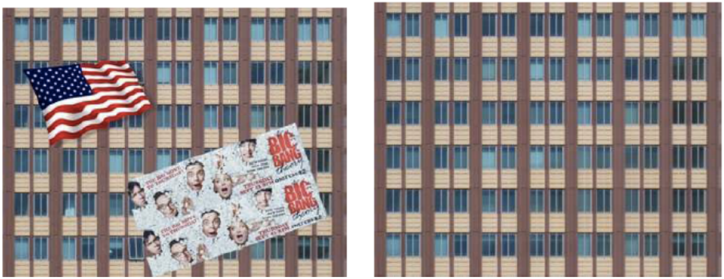
200 frames,
72 x 88 pixels,

Significant foreground motion



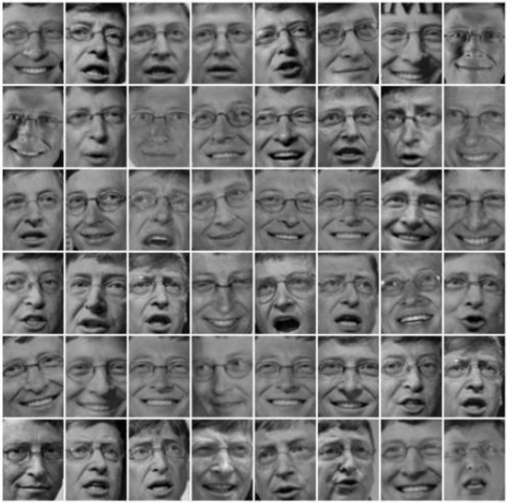
Repairing low-rank textures





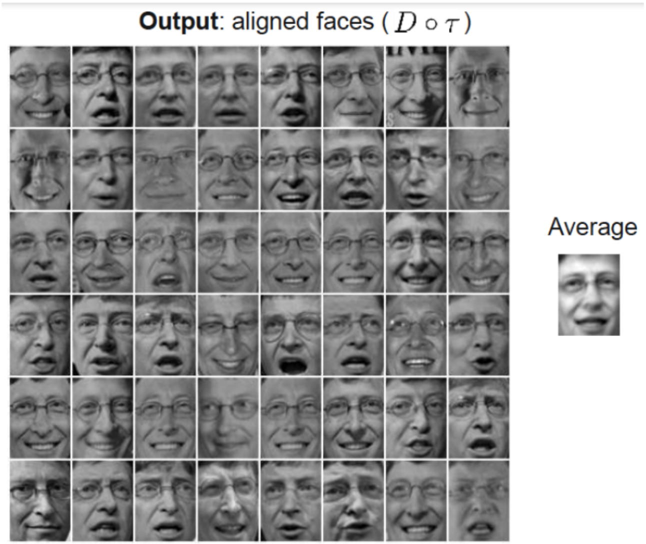
Face alignment

Input: faces detected by a face detector (D)



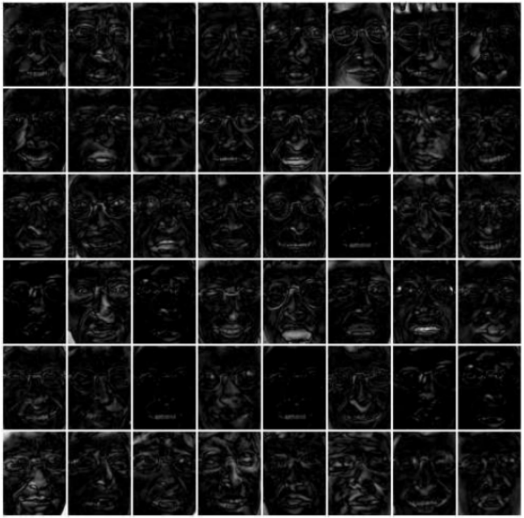
Average

Face alignment



Face alignment

Output: sparse error images (E)



References I

Aybat, N. S. and Iyengar, G. (2015).
An alternating direction method with increasing penalty for stable principal component pursuit.
Computational Optimization and Applications, 61(3):635–668.

Cai, J.-F., Candès, E. J., and Shen, Z. (2010).
A singular value thresholding algorithm for matrix completion.
SIAM Journal on optimization, 20(4):1956–1982.

Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011).
Robust principal component analysis?
Journal of the ACM (JACM), 58(3):11.

Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. (2011).
Rank-sparsity incoherence for matrix decomposition.
SIAM Journal on Optimization, 21(2):572–596.

Garey, M. R. and Johnson, D. S. (1990).
Computers and Intractability; A Guide to the Theory of NP-Completeness.
W. H. Freeman & Co., New York, NY, USA.

Jang, S. (2014).
Basics and examples of principal component analysis (pca).
https://www.projectrhea.org/rhea/index.php/PCA_Theory_Examples/.

References II

Lin, T., Ma, S., and Zhang, S. (2018).
Global convergence of unmodified 3-block admm for a class of convex minimization problems.
Journal of Scientific Computing, 76(1):69–88.

Natarajan, B. K. (1995).
Sparse approximate solutions to linear systems.
SIAM journal on computing, 24(2):227–234.

Nesterov, Y. (1983).
A method of solving a convex programming problem with convergence rate $\mathcal{O}(\frac{1}{k^2})$.
In *Soviet Math. Dokl*, volume 27.

Recht, B., Fazel, M., and Parrilo, P. A. (2010).
Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization.
SIAM review, 52(3):471–501.

Richardson, M. (2009).
Principal component analysis.
http://people.maths.ox.ac.uk/richardsonm/SignalProcPCA.pdf/.

Shen, Y., Wen, Z., and Zhang, Y. (2014).
Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization.
Optimization Methods and Software, 29(2):239–263.

References III

Tao, M. and Yuan, X. (2011).
Recovering low-rank and sparse components of matrices from incomplete and noisy observations.
SIAM Journal on Optimization, 21(1):57–81.

Wold, S., Esbensen, K., and Geladi, P. (1987).
Principal component analysis.
Chemometrics and intelligent laboratory systems, 2(1-3):37–52.

Wright, J., Ganesh, A., Rao, S., Peng, Y., and Ma, Y. (2009).
Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization.
In *Advances in neural information processing systems*, pages 2080–2088.

Yi, X., Park, D., Chen, Y., and Caramanis, C. (2016).
Fast algorithms for robust pca via gradient descent.
In *Advances in neural information processing systems*, pages 4152–4160.

Zhang, T. and Yang, Y. (2018).
Robust pca by manifold optimization.
The Journal of Machine Learning Research, 19(1):3101–3139.

References IV

Zhou, Z., Li, X., Wright, J., Candes, E., and Ma, Y. (2010).
Stable principal component pursuit.
In *2010 IEEE international symposium on information theory*, pages 1518–1522. IEEE.