Report | HW 5

CS754 - Advanced Image Processing

Omkar Shirpure (22B0910)
Krish Rakholiya (22B0927)
Suryansh Patidar (22B1036)

Declaration: The work submitted is our own, and we have adhered to the principles of academic honesty while completing and submitting this work. We have not referred to any unauthorized sources, and we have not used generative AI tools for the work submitted here.

Q1

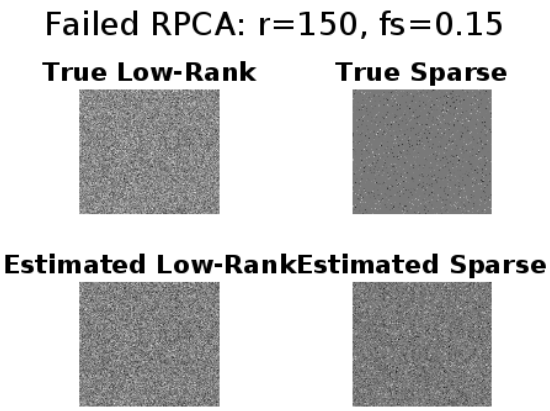

Figure 1

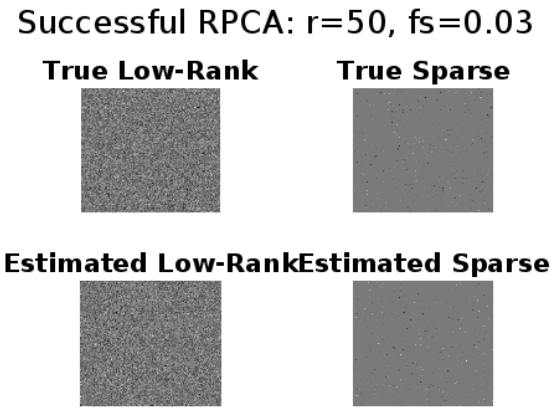

Figure 2

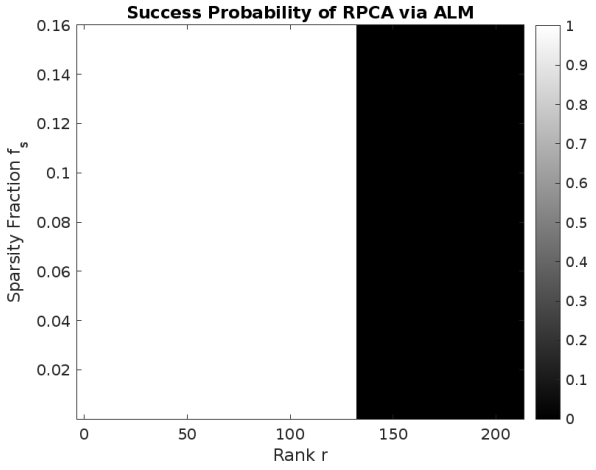

Figure 3

Q2

Three Fundamental Differences Between Robust PCA and L1-norm PCA

1. **Optimization Objective**

   - **Robust PCA**: Decomposes the data matrix D into a low-rank component L and a sparse component S by solving:

     $$\min_{L,S} \|L\|_* + \lambda \|S\|_1 \quad \text{subject to } D = L + S$$

     where $\|L\|_*$ is the nuclear norm (sum of singular values) promoting low-rank structure, and $\|S\|_1$ is the element-wise ℓ_1-norm promoting sparsity.

   - **L1-norm PCA**: Seeks projection directions w that maximize the sum of absolute values of projected data:

     $$\max_{w} \sum_{i} |w^\top x_i| \quad \text{subject to } \|w\| = 1$$

     replacing the traditional ℓ_2-norm with ℓ_1-norm to reduce sensitivity to outliers.

2. **Robustness Mechanism**

   - **Robust PCA**: Explicitly separates noise/outliers (in S) from clean data (in L). Assumes noise is sparse but possibly large in magnitude.

   - **L1-norm PCA**: Uses the ℓ_1-norm in the projection objective to downweight the influence of outliers, but does not explicitly model or separate them from the data.

3. **Output and Interpretation**

   - **Robust PCA**: Produces two outputs—a low-rank matrix L (clean data) and a sparse matrix S (outliers). This provides a denoised version of the data.

   - **L1-norm PCA**: Yields robust principal components, but does not reconstruct a cleaned version of the data or explicitly identify outliers.

Q3

Title: Robust Principal Component Analysis using Density Power Divergence by *Subhrajyoty Roy, Ayanendranath Basu, Abhik Ghosh*
Link: `https://jmlr.org/papers/v25/23-1096.html`

Key Theorem

Theorem (High Breakdown Point of MDPD-based RPCA):
The proposed MDPD-based RPCA estimator achieves a high breakdown point, maintaining robustness against outliers irrespective of the data's dimensionality.

Advancement Over Classical RPCA

Traditional RPCA (such as Principal Component Pursuit) decomposes a data matrix D into a low-rank component L and a sparse component S by solving:

$$\min_{L,S} \|L\|_* + \lambda \|S\|_1 \quad \text{subject to } D = L + S$$

While effective, this method may suffer from:

- High computational cost in high-dimensional settings.
- Sensitivity to the proportion and magnitude of outliers.

The proposed MDPD-based RPCA method advances the theory by:

- Using a divergence measure that downweights the influence of outliers.
- Achieving a high breakdown point, offering robustness even when a significant portion of the data is corrupted.
- Providing improved computational efficiency for large-scale data.

Application

The paper demonstrates the application of MDPD-based RPCA in **fraud detection in credit card transactions**, a domain characterized by:

- High-dimensional feature spaces.
- Frequent occurrence of outliers in the form of fraudulent transactions.

The proposed method effectively identifies the underlying structure of legitimate transactions and isolates anomalies, thus enhancing fraud detection performance.

Q4

(a)

Suppose we have a dictionary $D = [d_1, d_2, \ldots, d_K] \in \mathbb{R}^{n \times K}$ learned to sparsely represent a class S of images (e.g., handwritten digits or letters). Now we are given a new class S_1 consisting of images obtained by applying known affine transforms A_1 and A_2 to subsets of images in S.

Assume:

- Images in S consist of a foreground on a constant zero-valued background.
- The affine transformations $A_1, A_2 \in \mathbb{R}^{2\times 3}$ do not move the foreground outside the image canvas.
- Dictionary re-learning is not allowed due to computational cost.

Goal is to Convert the original dictionary D into a new dictionary D' that can sparsely represent images in class S_1, without re-learning.

Affine transformations applied to images can be compensated by applying the same transformations to the dictionary atoms. Specifically, if $x' = Ax$ and $x \approx D\alpha$, then:

$$x' \approx AD\alpha$$

Thus, we can define a new dictionary:

$$D' = AD$$

to directly represent the transformed image x'.

Solution

For images in S_1 created by transforming with A_1 and A_2, define:

$$D_1 = A_1 D, \quad D_2 = A_2 D$$

Then:

- Use D_1 to sparsely code images transformed by A_1.
- Use D_2 to sparsely code images transformed by A_2.

(b)

$f = D\theta$, $\forall f \in S$, where θ_f is k-sparse. Now for the vectorised Image, f',

$$f' = \alpha(f \odot f) + \beta f + \gamma b$$

here $\odot$ denotes point wise multiplication and b=[111 ...1]^T

$$\begin{aligned}
\implies f' &= \alpha(D\theta_f \odot D\theta_f) + \beta D\theta_f + \gamma b \\
&= \alpha\left(\sum_i D_f\theta_{f_i}\right) \odot \left(\sum_j D_j\theta_{f_j}\right) + \beta D\theta_f + \gamma b \\
&= \alpha\left(\sum_{i,j}(D_i \odot D_j)\theta_{f_i}\theta_{f_j}\right) + \beta D\theta_F + \gamma b
\end{aligned}$$

Let $\tilde{D} = D_i \odot D_j, \forall i, j \leq K$, where K is no. of columns in D. So,

$$f' = \alpha\tilde{D}\tilde{\theta} + \beta D\theta + \gamma b$$

$$\implies f' = [b|D|\tilde{D}] \begin{bmatrix} \gamma \\ \beta\theta \\ \alpha\tilde{\theta} \end{bmatrix}$$

Here $\tilde{\theta} = vec\{\theta\theta^T\}$
Therefore , for class S3, Dictionary $D' = [b|D|\tilde{D}|$
The new dictionary contains the columns of D, element-wise products of all pairs of columns of D, and a column vector containing all 1s,(repetitive columns can be dropped)

(c)

Suppose we have a dictionary $D = [d_1, d_2, \ldots, d_K] \in \mathbb{R}^{n \times K}$ learned to sparsely represent a class S of images. Now consider a new image class S_4 obtained by downsampling the images in S by a factor of k in both X and Y directions.

We assume:

- Downsampling is known and consistent (e.g., bilinear or nearest-neighbor).
- The dictionary D is fixed and re-learning is not allowed due to computational cost.
- Each dictionary atom d_i corresponds to a vectorized image patch.

Goal is to transform the existing dictionary D into a new dictionary D' such that it can sparsely represent the downsampled images in class S_4.

Approach

Let $\mathcal{D}_k$ denote the downsampling operator by a factor of k in both spatial dimensions. Then, the transformed dictionary D' is defined by applying $\mathcal{D}_k$ to each atom in D:

$$\boxed{D' = [\mathcal{D}_k(d_1),\ \mathcal{D}_k(d_2),\ \ldots,\ \mathcal{D}_k(d_K)]}$$

Each atom d_i is first reshaped to its original 2D form, downsampled using $\mathcal{D}_k$, and then vectorized again to form the new dictionary D'.

Justification

If an original image $x \in S$ has a sparse representation $x \approx D\alpha$, then the downsampled version $x' = \mathcal{D}_k(x)$ satisfies:

$$x' = \mathcal{D}_k(x) \approx \mathcal{D}_k(D\alpha) = \mathcal{D}_k(D)\alpha = D'\alpha$$

Thus, the downsampled image x' can be approximately represented using the same sparse code α and the downsampled dictionary D'.

(d)

We have the blur kernel, b as $\sum_{i=1}^{n} \beta_i b_i$, where $\{b_1 b_2 ... b_n\}$ = B
So the new image f',

$$f' = b * reshaped(f) = \left(\sum_{i=1}^{n} \beta_j b_j \right) * Reshaped(D\theta_f)$$

$$= \left(\sum_{i=1}^{n} \beta_j b_j \right) * Reshaped \left(\sum_{k}^{K} D_k \theta_{f_k} \right)$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \beta_i b_i * Reshaped(D_k)\theta_k$$

Thus, for Class S5, dictionary D' is obtained by convolving each column of D(reshaped) with every blur kernel $\in B$

(e)

Let the radon transform matrix be R_θ, for angle θ We have,

$$f' = R_\theta reshaped(f)$$

$$f' = R_\theta reshaped(D\alpha_f) = \sum_i R_\theta reshaped(D_i \alpha_{f_i})$$

where α_i is k-sparse.
Therefore, for Class S6, dictionary is obtained as $D'_i = R_\theta reshaped(D_i)$, here every column vector of D is reshaped into a 2D matrix.

Q5

(a)

First of all, without loss of generality, let's assume that $m \leq n$.

Now let's consider the SVD of A as

$$A = UDV^\top$$

Now let's use the equation:

$$X_F^2 = X^\top X$$

Thus, we can rewrite the objective function as:

$$\begin{aligned} J(A_r) &= (A - A_r)^\top (A - A_r) \\ &= VV^\top (A - A_r)^\top UU^\top (A - A_r) \\ &= V^\top (A - A_r)^\top UU^\top (A - A_r)V \\ &= (D - Z)^\top (D - Z) \\ &= Z - D_F^2 \end{aligned}$$

, where $Z = U^\top A_r V$ is a rank-r matrix.

The minimum of this objective function is achieved when Z is diagonal, with the first r diagonal elements being the first r diagonal elements of D and the rest being zero. This is because the singular values are in decreasing order in D, so picking the largest r values would minimize the Frobenius norm.

Let M be an $m \times m$ matrix, with the first r elements being 1, and the rest being 0. Then, $Z = MD$ is the required rank-r matrix. Hence,

$$A_r = UMDV^\top$$

(b)

We will again use the following equation

$$X_F^2 = X^\top X$$

We can rewrite the objective function as:

$$\begin{aligned} J(R) &= (A - RB)^\top (A - RB) \\ &= A^\top A + B^\top R^\top RB - B^\top R^\top A - A^\top RB \\ &= A^\top A + B^\top R^\top RB - 2A^\top RB \\ &= A^\top A + B^\top B - 2A^\top RB \qquad [R \text{ is orthonormal}] \end{aligned}$$

The first 2 terms are constants, and minimizing a negative quantity is equivalent to maximizing the positive quantity.

Thus, we need to maximise

$$\begin{aligned} A^\top RB &= BA^\top R \\ &= ZR \end{aligned}$$

, where $Z = BA^\top$. Now consider the SVD of Z as $Z = UDV^\top$. The equation then simplifies to

$$\begin{aligned} ZR &= UDV^\top R \\ &= DV^\top RU \\ &= DQ \end{aligned}$$

, where Q is orthonormal. The maximum of this is achieved when $Q_{ii} = 1$ all along its diagonal. Since Q is orthonormal, $Q = I$.

Thus, $R = VU^\top$ is the required orthonormal matrix. However, there is a subtle problem. R won't be a rotation matrix if $\det(R) = -1$ instead of 1. And simply flipping the sign of R won't help, as the sign of the objective function will change.

In such a case, we can consider R as

$$R = VTU^\top$$

, where $T = \text{diag}(1, 1, \ldots, 1, -1)$ has shape $(n, 1)$. This way, we are only subtracting the smallest singular vector of R from the objective function and not changing the sign of the determinant.

So, we can define $W = \text{diag}(1, 1, \ldots, 1, \det(VU^\top))$.
Finally, we have

$$R = VWU^\top$$

(c)

$$J_3(A) = \|C - A\|_F^2 + \lambda \|A\|_1$$

where C is a known matrix.

Efficient Minimization

This is a sparse matrix approximation problem. The objective is to find a matrix A that is close to C in Frobenius norm while also being sparse, enforced via the ℓ_1-norm.

A standard method to minimize this cost is the **Iterative Soft Thresholding Algorithm (ISTA)**, or its accelerated version **FISTA (Fast ISTA)**.

The soft-thresholding operator is applied element-wise:

$$\text{SoftThreshold}(x, \tau) = \text{sign}(x) \cdot \max(|x| - \tau, 0)$$

Thus, at each iteration:

$$A^{(t+1)} = \text{SoftThreshold}(C, \tau), \quad \text{with } \tau = \frac{\lambda}{2}$$

Application

Image Denoising using Sparse Coding: In this context, C represents a noisy image, and we aim to find a sparse approximation A of it by minimizing J_3. The sparsity prior helps eliminate noise while preserving important image features.

(d)

$$J_4(A) = \|C - A\|_F^2 + \lambda \|A\|_*$$

where $\|A\|_*$ denotes the nuclear norm (sum of singular values of A).

Efficient Minimization

This problem promotes low-rank approximation of C. It is typically solved using the **Singular Value Thresholding (SVT)** algorithm.

Steps:

1. Compute the singular value decomposition (SVD) of C:

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

2. Apply soft-thresholding to the singular values:

$$\Sigma' = \text{diag}(\max(\sigma_i - \tau, 0)), \quad \text{where } \tau = \frac{\lambda}{2}$$

3. Reconstruct the matrix:

$$A = U\Sigma' V^\top$$

Application

Image Inpainting: This problem arises when some pixels of an image are missing or corrupted. By assuming the image has a low-rank structure, J_4 can be minimized to recover the missing data effectively, leveraging redundancy in the image.