

A convex optimization approach to PCA

Vincenzo Bazzucchi

June 9, 2018

Abstract

Principal Component Analysis is a fundamental technique in statistics and unsupervised dimensionality reduction. In this paper **Principal Component Pursuit (PCP)** is explored, a convex problem whose solution separates a lower rank representation and the sparse noise from a given data matrix M

1 Introduction

Classical Principal Component analysis of the matrix $M \in \mathbb{R}^{n \times m}$ can be formulated as the following problem:

$$\begin{aligned} & \underset{L}{\text{minimize}} && \|M - L\| \\ & \text{subject to} && \text{rank}(L) \leq \lambda \end{aligned} \tag{1}$$

However this approach is sensitive to corrupted measurements which are very common in image processing, web data analysis and bioinformatics just to quote a few. [1] introduced **Principal Component Pursuit**:

$$\begin{aligned} & \underset{L, S}{\text{minimize}} && \|L\|_* + \lambda \|S\|_1 \\ & \text{subject to} && L + S = M \end{aligned} \tag{2}$$

Where L is a lower rank matrix, S a sparse matrix, $\|M\|_* = \sum_i \sigma_i(M)$ is the nuclear norm (the sum of the eigenvalues) of the matrix M and $\|M\|_1 = \sum_{i,j} |M_{ij}|$

The first obvious remark is that the the problem 2 is convex: all the p -norms are convex functions and the constraint is linear.

2 A Semi-Definite Programming Problem

2 can be rewritten by leveraging that $M = L + S \Leftrightarrow L = M - S$.

$$\underset{L}{\text{minimize}} \quad \|M - S\|_* + \lambda \|S\|_1 \tag{3}$$

The optimal value of 2 then becomes

$$\begin{aligned} p^* &= \min_S \|M - S\|_* + \lambda \|S\|_1 \\ &= \min_S \max_{Y, Z} \text{trace}(Y^T(M - S)) + \text{trace}(Z^T S) : \|Y\|_2 \leq 1, \|Z\|_\infty \leq \lambda \\ &= \max_{Y, Z} \min_S \text{trace}(Y^T(M - S)) + \text{trace}(Z^T S) : \|Y\|_2 \leq 1, \|Z\|_\infty \leq \lambda \\ &= \max_Y \text{trace}(Y^T M) : \|Y\|_2 \leq 1, \|Y\|_\infty \leq \lambda \end{aligned} \tag{4}$$

Where $Y \in \mathbb{R}^{n \times n}$ is the *Lagrange multiplier*

The condition $\|Y\| \leq 1$ is equivalent to $I - YY^T \succeq 0$ which can be written as a linear matrix inequality using Shur complements:

$$X = \begin{bmatrix} I & Y \\ Y^T & I \end{bmatrix} \succeq 0$$

This problem can therefore be expressed as an SDP:

$$\begin{aligned} & \underset{Y}{\text{maximize}} && \text{trace}(Y^T M) \\ & \text{subject to} && \begin{bmatrix} I & Y \\ Y^T & I \end{bmatrix} \succeq 0 \\ & && \lambda \leq Y_{ij} \leq \lambda, 1 \leq i \leq j \leq n \end{aligned} \tag{5}$$

3 The Alternating Direction Method

First let's define the augmented Lagrangian:

$$l(L, S, Y) = \|L\|_* + \lambda \|S\|_1 + \text{trace}(Y^T(M - L - S)) + \frac{\mu}{2} \|M - L - S\|_F^2$$

Then iteratively solve the problem is solved by optimizing the primal variables and then by using the new values of the primal variables to update the dual variable:

1. $L_k, S_k = \arg \min_{L, S} l(L, S, Y_k)$
2. $Y_k = Y_k + \mu(M - L_k - S_k)$

To compute L_k, S_k [1] introduced

- the **shrinkage** operator: $\mathcal{S}_\tau(x) = \text{sgn}(x) \max(|x| - \tau, 0)$
- the **thresholding** operator: $\mathcal{D}_\tau(A) = U\mathcal{S}_\tau(\Sigma)V^T$ where $A = U\Sigma V^T$ is a SVD decomposition of A

Then

$$\arg \min_L l(S_k, L, Y_k) = \mathcal{S}_{\lambda\mu^{-1}}(M - L + \mu^{-1}Y_k)$$

$$\arg \min_S l(S, L_k, Y_k) = \mathcal{D}_{\mu^{-1}}(M - S + \mu^{-1}Y_k)$$

Using $\mu = \frac{nm}{4\|M\|_1}$, $\lambda = \frac{1}{\sqrt{\max(n,m)}}$ and stopping when $\|M - L - S\|_F \leq 10^{-7} \|M\|_F$

An implementation of this algorithm in Python can be found in the `robust_pca.py` file.

4 Recovering corrupted images

RPCA is often used to recover data from corrupted matrices. [1] shows this generating random matrices and then perturbing them before applying the algorithm. In some papers using Robust PCA that were analyzed, the algorithm is used to extract the background from video surveillance footage ([1], [2]) or for data analysis [3].

For this project, I attempted to apply this powerful algorithm to a type of data mentioned in the original paper: images. Some random uniform noise was generated and applied to random pixels of the red, green and blue channels of an image. Then robust PCA was executed on the perturbed images and the lower rank matrix was compared to the original image. The experiment was replicated varying the ratio of perturbed pixels. The results are displayed in figure 1 while the code necessary to corrupt and recover the images is in `image_recovery.py`.

The accuracy of the recovery was measured by computing the Pearson correlation coefficient¹ between the original image and the corrupted image channel by channel and by averaging over the three channels. The same computation is performed on the recovered image. The similarities values are also shown in Figure 1. As a reminder, the Pearson correlation coefficient value are in the range $[-1, 1]$ where a value close to 1 indicates high similarity. The experiment can be run executing `run.py`

References

- [1] Cands, E. J., Li, X., Ma, Y. and Wright, J. (2009). *Robust principal component analysis?*
- [2] Pupusha, I. (2011). Fast automatic background extraction via Robust PCA
- [3] Fan, J., Sun, Q., Zhou, W.X., Zhu, Z. *Principal component analysis for big data*

¹https://en.wikipedia.org/wiki/Pearson_correlation_coefficient#For_a_sample

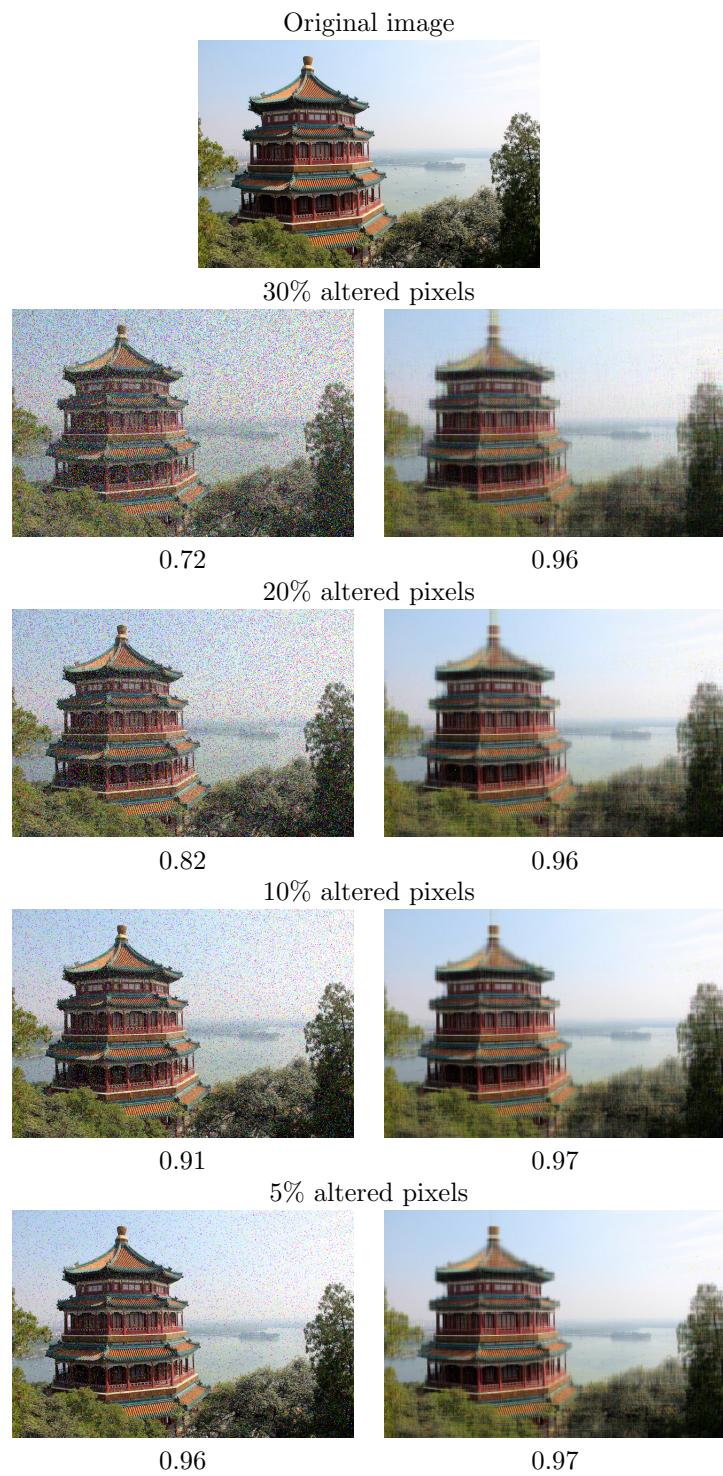


Figure 1: Recovering a corrupted image. On the left the corrupted images with different ratios of altered pixels, on the recovered image