Signal Processing and Optimization for Big Data

31 October 2025

1)Let us consider the following observation model

$$\boldsymbol{y} = \Phi\Psi\boldsymbol{s}, \tag{1}$$

Where $\boldsymbol{y} \in \mathbb{R}^{M\times 1}$ represents a vectorized gray-scale image, $\Phi \in \mathbb{R}^{N\times N}$ is a sampling matrix, $\Psi \in \mathbb{R}^{M\times N}$ is an orthonormal basis matrix, and $\boldsymbol{s} \in \mathbb{R}^{N\times 1}$ is a k-sparse vector. Let's Suppose that the matrix Ψ is obtained from the inverse 2D-DCT matrix, denoted as $\boldsymbol{C}_n$. The observation model becomes as follows

$$\boldsymbol{x} = \operatorname{vec}(\mathbf{X}) = \operatorname{vec}(\boldsymbol{C_n}\mathbf{S}\boldsymbol{C_n^t}) = (\boldsymbol{C_n} \otimes \boldsymbol{C_n^t})\operatorname{vec}(\mathbf{S}). \tag{2}$$

- How can you choose M (i.e., the number of samples) and Φ to guarantee a proper reconstruction of the original signal $\boldsymbol{x} = \Psi\boldsymbol{s}$?
- Write a function to solve the *Basis-Pursuit* problem through ADMM

$$\begin{aligned} \min_{\boldsymbol{s}} \quad & \|\boldsymbol{s}\|_1^1 \\ \text{s.t.} \quad & \boldsymbol{y} = \Phi\Psi\boldsymbol{s} \end{aligned}$$

- Use the previous function to reconstruct an image X of your choice from $M < N$ samples and evaluate the quality of your reconstruction $\hat{\mathrm{X}}$ using the Peak Signal To Noise Ratio (PSNR), given by

$$\gamma = 20\log_{10}\left(\frac{m_I}{\sqrt{mse}}\right), \tag{4}$$

  where m_I is the maximum pixel value of the original image, and the (normalized) mean squared error is defined as

$$mse = \frac{\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2}{\|\mathbf{X}\|_F^2}$$

  where $\|\cdot\|_F$ denotes the Frobenius norm.[1]
- How can you deal with large images (e.g., with size 256×256 px)? [2] [3]

[1] *Pay attention to the image size!*
[2] *There are several test images (e.g.,* `"peppers.png"`*, obtainable with the command* `imread("peppers.png")`*.)*
[3] *To convert an image in grey-scale use* `rgb2grey()`*. To normalize the image pixels in* $[0, 1]$ *use* `im2double()`*.*

2) Let us consider the following optimization problem

$$\begin{aligned} \min_{\boldsymbol{s}} \quad & \|\boldsymbol{y} - \boldsymbol{\theta}\boldsymbol{s}\|_2^2 \\ \text{s.t.} \quad & \|\boldsymbol{s}\|_1^1 \leq t \end{aligned}$$

where $\boldsymbol{y} \in \mathbb{R}^{N\times 1}$ represents a (vectorized) image, $\boldsymbol{\theta} \in \mathbb{R}^{N\times N}$ is an orthonormal matrix and $\boldsymbol{s} \in \mathbb{R}^{N\times 1}$ represents a k-sparse representation of the image. Assuming that $\boldsymbol{y}$ represents a noisy version of the image, the previous optimization problem can be exploited to perform signal **de-noising**.

- Consider an image X of your choice, and corrupt it with additive noise, thus obtaining its noisy version Y. For computational convenience, consider a small image (e.g., with size 32×32 px).
- Solve the previous LASSO optimization problem to de-noise the image. Plot the PSNR of the reconstructed solution for different values on the sparsity constraint t.
- Plot the reconstructed image for the best number of t.

3) Another proper way to solve the image reconstruction problem in presence of noise is given by the *Quadratically Constrained Basis-Pursuit*, defined as

$$\begin{aligned} \min_{\boldsymbol{s}} \quad & \|\boldsymbol{s}\|_1^1 \\ \text{s.t.} \quad & \|\boldsymbol{y} - \boldsymbol{\theta}\boldsymbol{s}\|_2^2 \leq \eta \end{aligned}$$

- Write a Matlab function to solve the problem. Assume to split the image in patches to deal also with large images (e.g., with size 256×256 px).
- Use the function to de-noise the image you considered at the previous point, show the reconstructed image, and evaluate the reconstruction PSNR.
- Do you find any issue with the reconstructed image? If yes, how can you avoid it?

4) Let us consider the following observation model

$$\begin{aligned} \mathbf{Y} &= \mathbf{R}(\mathbf{X} + \mathbf{A}) + \mathbf{V}, \\ \mathbf{Z}_\Pi &= \mathcal{P}_\Pi(\mathbf{X} + \mathbf{A}) + \mathbf{W}_\Pi \end{aligned} \tag{7}$$

where $\mathbf{Y} \in \mathbb{R}^{L\times T}$, $\mathbf{R} \in \mathbb{R}^{L\times F}$, $\mathbf{X} \in \mathbb{R}^{F\times T}$, $\mathbf{V} \in \mathbb{R}^{L\times T}$, and P_Π is a sampling operator. We make the following assumptions

- $\mathbf{X}$ is a low-rank matrix.
- $\mathbf{A}$ is a sparse vector.

These models are known as *low-rank plus sparse models*, and they have different use cases. A typical example comes from the domain of networking, where many traffic flows are routed on the same link, i.e., $F >> L$.
The goal is to predict the regular traffic flows, represented by $\mathbf{X}$, and the anomalies, represented by $\mathbf{A}$, from observations of the aggregated traffic $\mathbf{Y}$ and the partial (subsampled) observations of the traffic flows $\mathbf{Z}_\Pi = \mathcal{P}_\Pi(\mathbf{X} + \mathbf{A}) + \mathbf{W}_\Pi$.

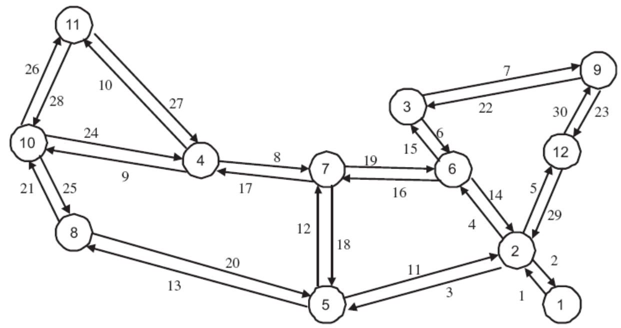

Figure 1: Example of network topology (Abilene network).

To this aim, a typical problem which can be considered is the following

$$\min_{\mathbf{X},\mathbf{A}} \quad \|\mathbf{X}\|_* + \lambda\|\mathbf{A}\|_1 \tag{8}$$

$$\text{s.t.} \quad Y = \mathbf{R}(\mathbf{X}+\mathbf{A}) \tag{9}$$

$$\mathbf{Z}_\Pi = \mathbf{P}_\Pi(\mathbf{X}+\mathbf{A}) \tag{10}$$

- Write a Matlab function to solve the optimization problem in (8).
- Test the function on the provided traffic flows dataset.
- Plot the traffic matrices and temporal behavior of the estimated traffic and estimated anomalies for a generic flow (e.g., f=70).
- Compute the NMSE of the obtained solution as

  $$E = \frac{\|\mathbf{Q} - (\mathbf{Y}+\mathbf{A})\|_F^2}{\|\mathbf{Q}\|_F^2},$$

  where $\mathbf{Q}$ is the ground-truth matrix.

5)Let us consider the following problem

$$\begin{aligned} \min_{\mathbf{X},\mathbf{A}} \|\mathbf{X}\|_* + \lambda\|\mathbf{A}\|_1 \\ \text{s.t. } \mathbf{Y} = \mathbf{X}+\mathbf{A}, \end{aligned} \tag{11}$$

where $\mathbf{Y} \in \mathbb{R}^{n\times m}$ is the observed (possibly noisy) signal, $\mathbf{X} \in \mathbb{R}^{n\times m}$ is the low-rank component and $\mathbf{A} \in \mathbb{R}^{n\times m}$ is the sparse component. In image processing $\mathbf{X}$ can model the signal of interest, while $\mathbf{A}$ can model a sparse noise term.

- Write a function to solve the previous optimization problem.
- Using `concorde.jpg` (provided in the exercise materials), corrupt the image with a *salt and pepper*[4] noise. Then evaluate the MSE as

  $$E = \frac{\|\mathbf{Y_n} - \mathbf{Y}\|_F^2}{\|\mathbf{Y}\|_F^2}$$

  .

[4]https://en.wikipedia.org/wiki/Salt-and-pepper_noise

- Apply the previously defined function to perform image de-noising, and evaluate the MSE of the de-noised image[5].

[5] Hint: you can split the image in different sub-patches to reduce the computational complexity.