Signal Processing and Optimization for Big Data

7 November 2025

1)Let us consider the following observation model

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

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

$$\mathbf{y} = \mathrm{vec}(\mathbf{X}) = \mathrm{vec}(\mathbf{C_n}\mathbf{S}\mathbf{C_n^t}) = (\mathbf{C_n} \otimes \mathbf{C_n^t})\,\mathrm{vec}(\mathbf{S}). \tag{2}$$

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

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

- Use this function to reconstruct an image $\mathbf{X}$ of your choice from $M < N$ samples, and evaluate the reconstruction quality $\hat{\mathbf{X}}$ through the peak signal to noise ratio (PSNR), given by

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

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

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

  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

$$\begin{aligned} &\min_{\mathbf{s}} \|\mathbf{y} - \boldsymbol{\theta}\mathbf{s}\|_2^2 \\ &\text{s.t.} \;\; \|\mathbf{s}\|_1^1 \leq t \end{aligned} \tag{5}$$

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

- Consider an image $\mathbf{X}$ of your choice and its noisy version $\mathbf{Y}$, corrupted with additive noise. De-noise the image exploiting the solution of (5). Plot the PSNR of the reconstructed solution for different values on the sparsity constraint t (for computational convenience, consider an image with size 32×32 px).
- 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

$$\begin{aligned} &\min_{\mathbf{s}} \|\mathbf{s}\|_1^1 \\ &\text{s.t.} \;\; \|\mathbf{y} - \boldsymbol{\theta}\mathbf{s}\|_2^2 \leq \eta \end{aligned} \tag{6}$$

- Write a Matlab function to solve the problem, and use it to de-noise an image of your choice, by also evaluating the reconstruction PSNR as a function of the constraint η. Split the image in patches to deal with large images (e.g., with size 256×256 px).
- 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

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

These models are known as *low-rank plus sparse models*, and they are particularly useful in the networking domain, as they model multiple traffic flows routed on the same link, (i.e., $F >> L$). In this scenario, a typical task consists to predict the regular and the anomalous traffic flows, represented by $\mathbf{X}$ and $\mathbf{A}$ respectively, from the observations of the aggregated traffic $\mathbf{Y}$ and partial (i.e., sub-sampled) observations of the traffic flows $\mathbf{Z}_\Pi = \mathcal{P}_\Pi(\mathbf{X} + \mathbf{A}) + \mathbf{W}_\Pi$. To this aim, we consider the following optimization problem

$$\begin{aligned} \min_{\mathbf{X},\mathbf{A}} \quad & \|\mathbf{X}\|_* + \lambda\|\mathbf{A}\|_1 \\ \text{s.t.} \quad & Y = \mathbf{R}(\mathbf{X} + \mathbf{A}) \\ & \mathbf{Z}_\Pi = \mathbf{P}_\Pi(\mathbf{X} + \mathbf{A}) \end{aligned} \tag{8}$$

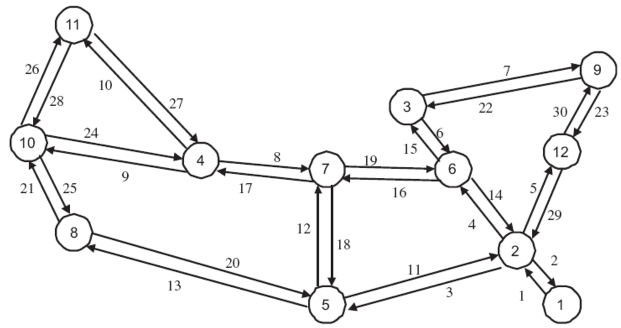

Figure 1: Example of network topology (Abilene network).

- Write a Matlab function to solve Problem (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

$$\begin{aligned} \min_{\mathbf{X},\mathbf{A}} \ & \|\mathbf{X}\|_* + \lambda\|\mathbf{A}\|_1 \\ \text{s.t. } & \mathbf{Y} = \mathbf{X} + \mathbf{A}, \end{aligned} \tag{9}$$

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

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

  .

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

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

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