

EE-556: Mathematics of data: from theory to computation.

Homework Exercise 3

Natalie Bolón Brun

12/2018

1 Learning-based Compressive Subsampling

1.1 Exercise 1.1

Prove that the estimator $\hat{x} = A^\dagger b = A^*(AA^*)^{-1}b$ is equivalent to expanding b to a p-dimensional vector by placing zeros in the entries corresponding to $\Omega^c = \{1, \dots, p\} \setminus \Omega$, and then applying the adjoint Ψ^* . Prove:

$$\hat{x} = \Psi^* P_\Omega^T b \quad (1)$$

Answer:

$$\hat{x} = A^*(AA^*)^{-1}b = \Psi^* P_\Omega^T (P_\Omega \Psi \Psi^* P_\Omega^T)^{-1} b \quad (2)$$

$$\Psi \Psi^* = I \quad (3)$$

$$P_\Omega^T P_\Omega = I_k \quad (4)$$

Therefore,

$$\hat{x} = \Psi^* P_\Omega^T (P_\Omega \Psi \Psi^* P_\Omega^T)^{-1} b = \Psi^* P_\Omega^T I_k b = \Psi^* P_\Omega^T b \quad (5)$$

1.2 Exercise 1.2

Show that the optimization problem 6 can be solved by sorting

$$\widehat{\Omega} = \arg \max_{\Omega: |\Omega|=n} \frac{1}{m} \sum_{j=1}^m \|P_\Omega \Psi x_j\|_2^2 \quad (6)$$

Answer:

We can do the express our objective function to be maximized as

$$\frac{1}{m} \sum_{j=1}^m \langle \Psi_j, x_j \rangle = \frac{1}{m} [\Psi_1 x_1, \Psi_2 x_2, \dots, \Psi_m x_m] \quad (7)$$

By sorting the samples in decreasing order (i.e. $\Psi_1 x_1 \geq \Psi_2 x_2 \geq \dots \geq \Psi_m x_m$) we can ensure that picking the first n indices the objective function is maximized. Then, $\widehat{\Omega}$ is equal to the first n indices.

Proof:

Consider $\exists x_k$ such that $k \notin \hat{\Omega}$ and which increases the value of $P_\Omega \Psi x_\Omega$ if we exchange x_k with any x_j such that $j \in \hat{\Omega}$.

Then, it will mean that $\Psi_k x_k > \Psi_j x_j \quad \forall j \in \hat{\Omega}$ but this is not possible since they are ordered in increasing value. Therefore, by just picking the n first indices we can ensure the objective value is maximized.

1.3 Exercise 1.3

Study the computational complexity of the Greedy mask optimization algorithm. The variables to be considered are the reconstruction complexity C_Δ^1 , the number of elements that we will sample n , the number of training elements m and the space dimension p .

1. Computational complexity of the greedy algorithm. How does adding k elements instead of 1 at each iteration could improve the complexity?

The computational complexity of the algorithm is

$$m * C_\Delta * \sum_{j=0}^{n-1} (p - j) \quad (8)$$

if we add one element per iteration of the **for** loop. At each iteration of the **for** loop, the non-linear operation is computed m times (one per number of dimensions of the input). Moreover, the **for** loop is computed p times for the first iteration of the while loop and then $p - j$ being j the current size of the set Ω until $j = n$.

If instead of adding 1 element to Ω at a time, k elements are added, the cost is reduced to

$$m * C_\Delta * \sum_{j=0}^{\frac{n-1}{k}} (p - j * k) \quad (9)$$

2. How does the parallelization of the for loop improve the complexity of the algorithm?

If the **for** loop is parallelized, the cost of the algorithm is reduced to:

$$m * C_\Delta * n \quad (10)$$

if 1 sample is added at each step or

$$m * C_\Delta * \frac{n}{k} \quad (11)$$

3. What else can be parallelized and what is the resulting complexity of the algorithm?

The computation of the non-linear operation Δ for each coordinate can be parallelized reducing the cost to

$$C_\Delta * \frac{n}{k} \quad (12)$$

where k represents the number of elements added to Ω per iteration.

4. What is the effect of replacing the whole set S with a batch S_t of cardinality t on the overall complexity? What are the benefits of such a procedure and why could you expect it to produce good results?

If the cardinality of S is limited to t , the cost of the algorithm without parallelized operations is

$$m * C_\Delta * \sum_{j=0}^{n-1} (t - j) \quad (13)$$

It must be taken into account that each iteration, the computation of the **for** loop is reduced by $k * m * C_\Delta$ being k the number of elements added to the set. Therefore, the initial operations have

always higher cost than the last operations. If the cardinality of S is reduced to t , it is like directly starting at iteration $(p - t)$ avoiding $(p - t)$ operations of higher cost.

If $(p - t) << n$ the algorithm may still produce good results as, in the worst case, $\frac{p-t}{n}$ percentage of proper indices will not be taken into account for the final set Ω but $\frac{n-(p-t)}{n}$ percentage remains equal than in the Ω set generated initially.

2 Proximal operators and image denoising

2.1 Exercise 2.1

Show that the problem 14 can be transformed into the computation of the proximal operator of the function $g(\mathbf{x}) := \lambda_1 \|\mathbf{x}\|_1$

$$\min_{\alpha \in \mathbb{R}^p} \left\{ \frac{1}{2} \|y - W^T \alpha\|_F^2 + \lambda_1 \|\alpha\|_1 \right\} \quad (14)$$

The definition of the proximal operator of $g(\mathbf{x})$ is:

$$prox_g(x) = \arg \min_z \left\{ \frac{1}{2} \|z - x\|_2^2 + \lambda_1 \|z\|_1 \right\} = \quad (15)$$

$$= \arg \min_z \left\{ \frac{1}{2} \|x - z\|_2^2 + \lambda_1 \|z\|_1 \right\} = \quad (16)$$

From problem 14 and taking $z = \alpha$ and $x = Wy$:

$$\arg \min_{\alpha} \left\{ \frac{1}{2} \|y - W^T \alpha\|_F^2 + \lambda_1 \|\alpha\|_1 \right\} = \quad (17)$$

$$= \arg \min_{\alpha} \left\{ \frac{1}{2} \|Wy - \alpha\|_F^2 + \lambda_1 \|\alpha\|_1 \right\} \quad (18)$$

This is equivalent to solving $prox_g(Wy)$ as both the Frobenius norm and the l_2 norm are computed identically along all the elements of the matrix or vector considered.

2.2 Exercise 2.2

Given $g(\mathbf{x}) := \lambda_1 \|\mathbf{x}\|_1$, show that the proximal function of $g(\mathbf{x})$ can be written as 19 where the operator *max* and *sign* are applied component-wise to the vector \mathbf{z} and \otimes stands for the component-wise multiplication.

$$prox_{\lambda g}(\mathbf{z}) = max(|\mathbf{z}| - \lambda, 0) \otimes sign(\mathbf{z}) =: S_\lambda(\mathbf{z}) \quad (19)$$

$$prox_g(y) = \arg \min_z \left\{ \frac{1}{2} \|z - y\|_2^2 + \lambda_1 \|z\|_1 \right\} \quad (20)$$

Computing the gradient of the function to be minimized and setting it to zero:

$$y = \begin{cases} z - \lambda & \text{if } z > 0 \\ z + \lambda & \text{if } z < 0 \\ 0 & \text{if } z = 0 \end{cases}$$

(21)

The first two expressions can be combined in a single one taking the absolute value of z:

$$y = \begin{cases} (|z| - \lambda) \otimes sign(z) & \text{if } z \neq 0 \\ 0 & \text{if } z = 0 \end{cases}$$

(22)

This can finally be reformulated as:

$$y = max(|z| - \lambda, 0) \otimes sign(z) \quad (23)$$

2.3 Exercise 2.3

Denoising.

The following denoised images are the result of solving 24 and 25

Lasso norm approach:

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} ||y - W^T \alpha||_F^2 + \lambda_1 ||\alpha||_1 \quad (24)$$

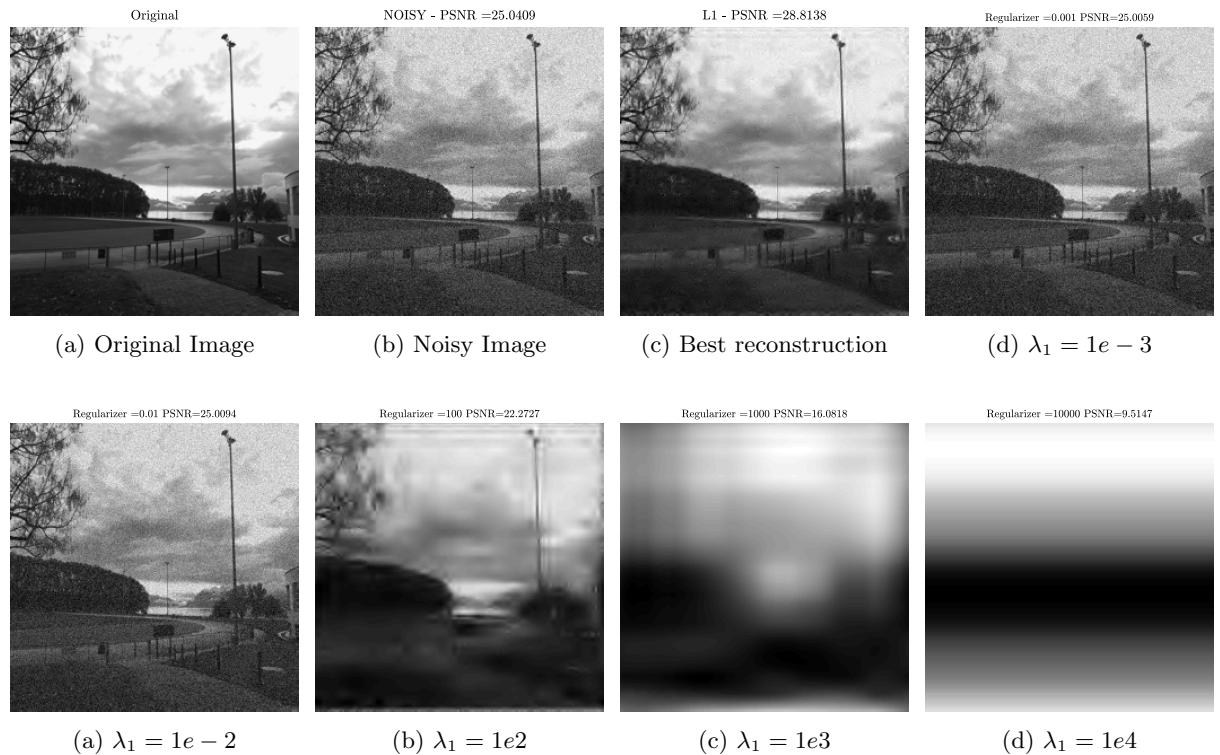


Figure 2: Reconstruction of image generated using very small and very large values of regularization parameter.

TV norm approach:

$$\min_{x \in \mathbb{R}^p} \frac{1}{2} \|y - x\|_F^2 + \lambda_{TV} \|x\|_{TV} \quad (25)$$

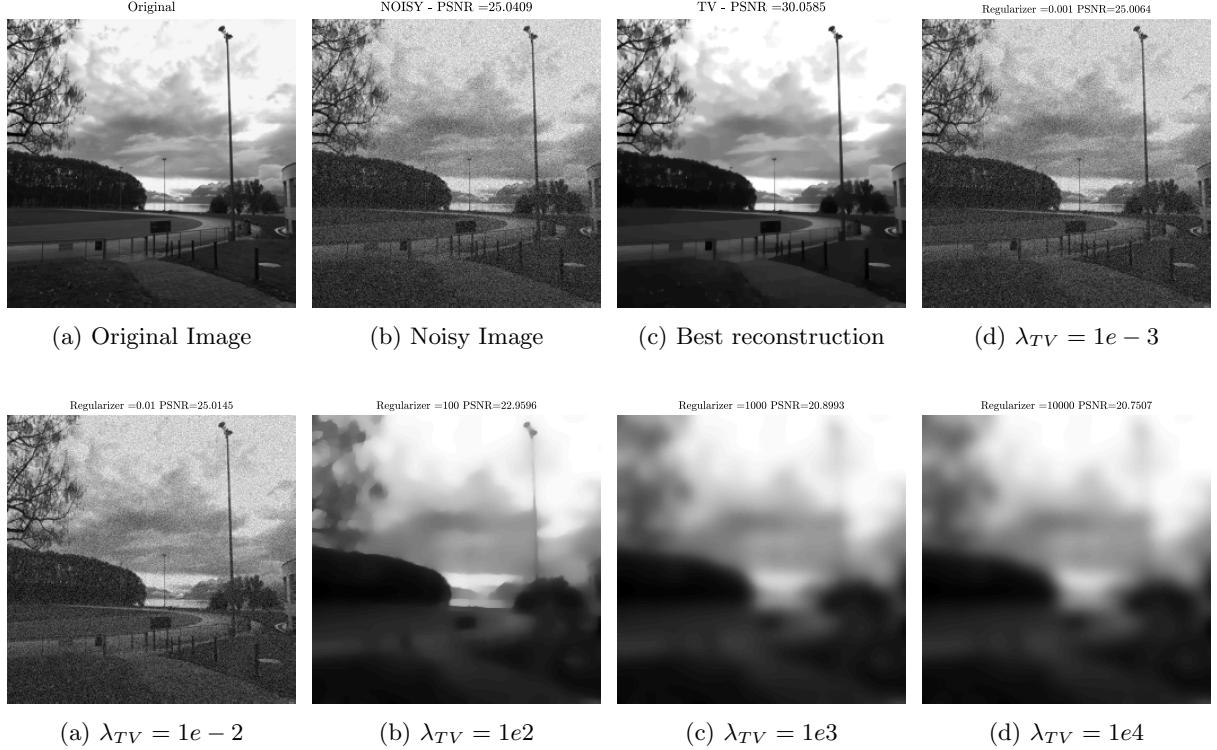


Figure 4: Reconstruction of image generated using very small and very large values of regularization parameter.

From the results obtained it can be seen that:

1. Small values of regularization parameter ($\lambda \leq 1$) do not allow to reduce the noise.
2. Large values of regularization parameter ($\lambda \geq 100$) result in a loss of the original image. In that case, the priority is given to obtaining a sparse structure or with low variance obtaining an image that losses the initial information.
In the case of l_1 regularization, the result is even more extrem, being impossible to intuit the initial image for regularization values higher than 100.

The best results are obtained with regularization parameters between 5 and 25 in both cases.

2.4 Exercise 2.4

Values of PSNR as a function of the regularization parameter.

Performing a parameter sweep the values found for λ_1 and λ_{TV} that result in higher PSNR are:

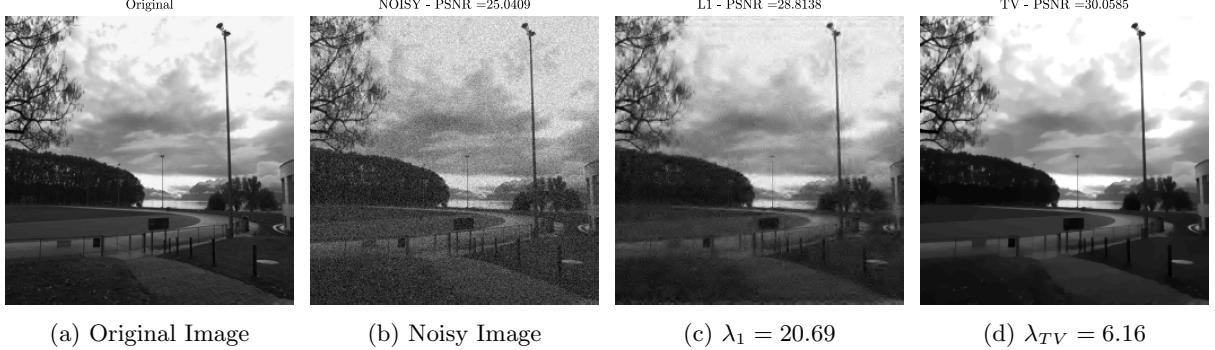


Figure 5: Comparison between methods. The images shown correspond to the original image, the noisy image generated adding Gaussian noise with standard deviation $\sigma = 15$, denoised image with l_1 norm on wavelet coefficients and denoised image with l_{TV} norm on the image domain.

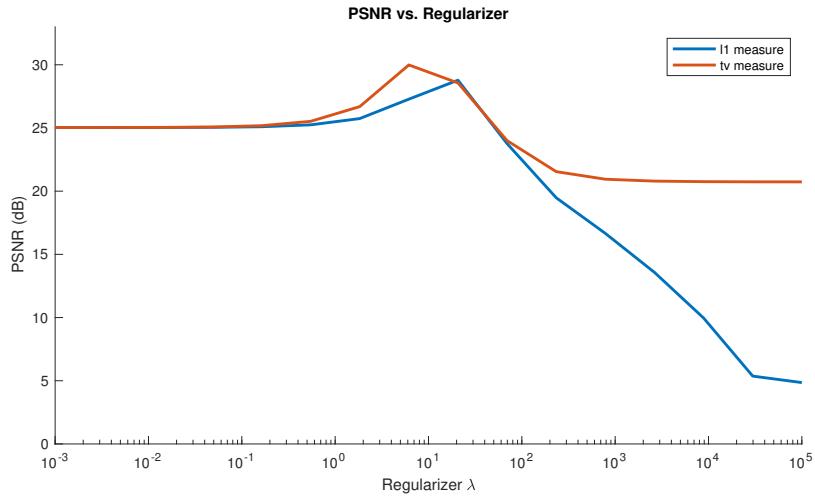


Figure 6: Regularization parameters sweep. The best results are obtained with $\lambda_1 = \mathbf{20.69}$ giving a **PSNR=28.81** and $\lambda_{TV} = \mathbf{6.16}$ giving a **PSNR=30.05**

The best result obtained with l_1 norm regularization is still noisy while it preserves sharp edges on the image. On the other side, the image obtained with TV-norm regularization is less noisy but the image is much smoother than the original one.

3 Non-smooth composite minimization and compressive MRI

In this section, the reconstruction image is given by solving the following minimization problem:

$$\min_{\alpha \in \mathbb{C}^p} \frac{1}{2} \|b - P_\Omega F W^T \alpha\|_2^2 + \lambda_1 \|\alpha\|_1 \quad (26)$$

The resulting images with the best found regularizers are:

3.1 Exercise 3.1

1. Write $f(\alpha)$ in Equation 26 as a function \tilde{f} of α_R and α_I where $\alpha = \alpha_R + \alpha_I i$.

Find the gradient $\nabla \tilde{f}(\alpha_R, \alpha_I)$

$$f(\alpha) = \frac{1}{2} \|b - P_\Omega F W^T \alpha\|_2^2 \quad (27)$$

$$\tilde{f}\left(\begin{bmatrix} \alpha_R \\ \alpha_I \end{bmatrix}\right) = \frac{1}{2} \|(b_R + ib_I) - (P_\Omega F_R W^T + i P_\Omega F_I W^T)(\alpha_R + i\alpha_I)\|_2^2 = \quad (28)$$

$$= \frac{1}{2} \|b_R - P_\Omega F_R W^T \alpha_R - (-1)P_\Omega F_I W^T \alpha_I\|_2^2 + \frac{1}{2} \|b_I - P_\Omega F_I W^T \alpha_R - P_\Omega F_R W^T \alpha_I\|_2^2 \quad (29)$$

$$= \frac{1}{2} \left\| b_R - P_\Omega [F_R \quad -F_I] W^T \begin{bmatrix} \alpha_R \\ \alpha_I \end{bmatrix} \right\|_2^2 + \frac{1}{2} \left\| b_I - P_\Omega [F_I \quad F_R] W^T \begin{bmatrix} \alpha_R \\ \alpha_I \end{bmatrix} \right\|_2^2 \quad (30)$$

The first term corresponds to the real terms while the second one corresponds to the imaginary ones.

To compute the gradient, each term will be treated separately: $\tilde{f}\left(\begin{bmatrix} \alpha_R \\ \alpha_I \end{bmatrix}\right) = \tilde{f}_1\left(\begin{bmatrix} \alpha_R \\ \alpha_I \end{bmatrix}\right) + \tilde{f}_2\left(\begin{bmatrix} \alpha_R \\ \alpha_I \end{bmatrix}\right)$

$$\nabla \tilde{f}_1 = -W \begin{bmatrix} F_R \\ -F_I \end{bmatrix} P_\Omega^T \left(b_R - P_\Omega [F_R \quad -F_I] W^T \begin{bmatrix} \alpha_R \\ \alpha_I \end{bmatrix} \right) \quad (31)$$

$$\nabla \tilde{f}_2 = -W \begin{bmatrix} F_I \\ F_R \end{bmatrix} P_\Omega^T \left(b_I - P_\Omega [F_I \quad F_R] W^T \begin{bmatrix} \alpha_R \\ \alpha_I \end{bmatrix} \right) \quad (32)$$

Finally, the gradient of $\tilde{f}\left(\begin{bmatrix} \alpha_R \\ \alpha_I \end{bmatrix}\right)$ is given by

$$\nabla \tilde{f}\left(\begin{bmatrix} \alpha_R \\ \alpha_I \end{bmatrix}\right) = \nabla \tilde{f}_1 + \nabla \tilde{f}_2 \quad (33)$$

2. Find the Lipschitz constant of $\nabla \tilde{f}(\alpha_R, \alpha_I)$

Let denote by A the operator $P_\Omega F W^T$, $A^* = P_\Omega F^* W^T$. To relate the procedure with the previous exercise, consider $A_R = P_\Omega F_R W^T$ and $A_I = P_\Omega F_I W^T$. Then, $\nabla^2 \tilde{f}$ can be expressed as:

$$\nabla^2 \tilde{f} = \nabla \left\{ \begin{bmatrix} -A_R^T \\ A_I^T \end{bmatrix} \left(b_I - [A_R \quad A_I] \begin{bmatrix} \alpha_R \\ \alpha_I \end{bmatrix} \right) - \begin{bmatrix} A_I^T \\ A_R^T \end{bmatrix} \left(b_I - [A_I \quad A_R] \begin{bmatrix} \alpha_R \\ \alpha_I \end{bmatrix} \right) \right\} = \quad (34)$$

$$\begin{pmatrix} A_R^T A_R & -A_R^T A_I \\ -A_I^T A_R & A_I^T A_I \end{pmatrix} + \begin{pmatrix} A_I^T A_I & A_I^T A_R \\ A_R^T A_I & A_R^T A_R \end{pmatrix} = \begin{pmatrix} A_R^T A_R + A_I^T A_I & -A_R^T A_I + A_I^T A_R \\ -A_I^T A_R + A_R^T A_I & A_I^T A_I + A_R^T A_R \end{pmatrix}$$

(35)

It can be seen that $A_R^T A_R + A_I^T A_I = (A^* A)_R$ and $-A_I^T A_R + A_R^T A_I = (A^* A)_I$. Therefore, we can express $\nabla^2 \tilde{f}$ as

$$\nabla^2 \tilde{f} = \begin{pmatrix} (A^* A)_R & -(A^* A)_I \\ (A^* A)_I & (A^* A)_R \end{pmatrix} \quad (36)$$

Using the previous procedure and taking into account that $\|A^* A\|_2 = \|WF^* P_\Omega^T P_\Omega FW^T\|_2 \leq 1$ as F and W are unitary matrices and $P_\Omega^T P_\Omega = I$ with $|\Omega|$ ones in the diagonal and the rest zero.

$$\left\| \nabla^2 \tilde{f} \begin{pmatrix} \alpha_R \\ \alpha_I \end{pmatrix} \right\|_2^2 = \left\| \begin{pmatrix} (A^* A)_R & -(A^* A)_I \\ (A^* A)_I & (A^* A)_R \end{pmatrix} \begin{pmatrix} \alpha_R \\ \alpha_I \end{pmatrix} \right\|_2^2 \quad (37)$$

$$= \left\| \begin{pmatrix} (A^* A)_R \alpha_R - (A^* A)_I \alpha_I \\ (A^* A)_I \alpha_R + (A^* A)_R \alpha_I \end{pmatrix} \right\|_2^2 = \|(A^* A)_R \alpha\|_2^2 + \|(A^* A)_I \alpha\|_2^2 = \|(A^* A) \alpha\|_2^2 \leq \|\alpha\|_2^2 \quad (38)$$

The Lipschitz constant is given by $\left\| \nabla^2 \tilde{f} \right\|_2$ and is shown to be 1.

3.2 Exercise 3.2

Implement the FISTA algorithm with restart for solving problem 39 where f and g are convex.

$$\min_{\alpha_R, \alpha_I} \{f(\alpha_R, \alpha_I) + g(\alpha_R, \alpha_I)\} \quad (39)$$

3.3 Exercise 3.3

Plot the obtained PSNR values as a function of the regularization parameter.

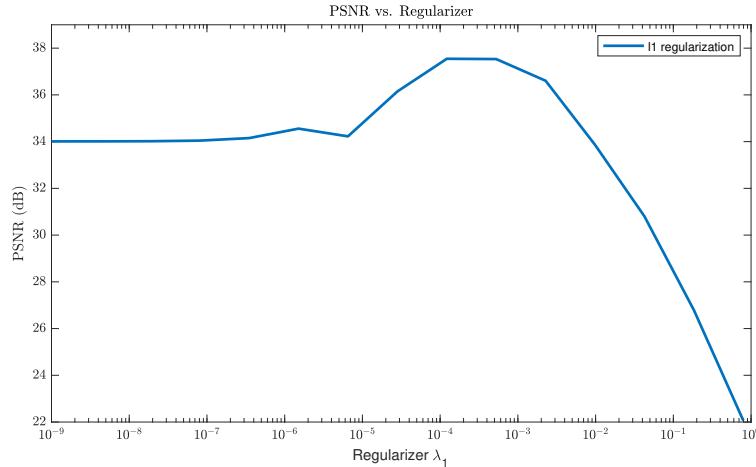


Figure 7: PSNR vs Regularizers. The highest PSNR value is achieved for $\lambda_1 = 1.22e-4$

With very small values of regularizer, the obtained image is quite similar to the original one but there are sharper values and the image is not completely properly defined. For very large values of regularizer, the image reconstructed is far from the original as it is too smoothed making it impossible to recover the desired image. Finally, for the best value obtained the reconstruction allows to perfectly read the original image.

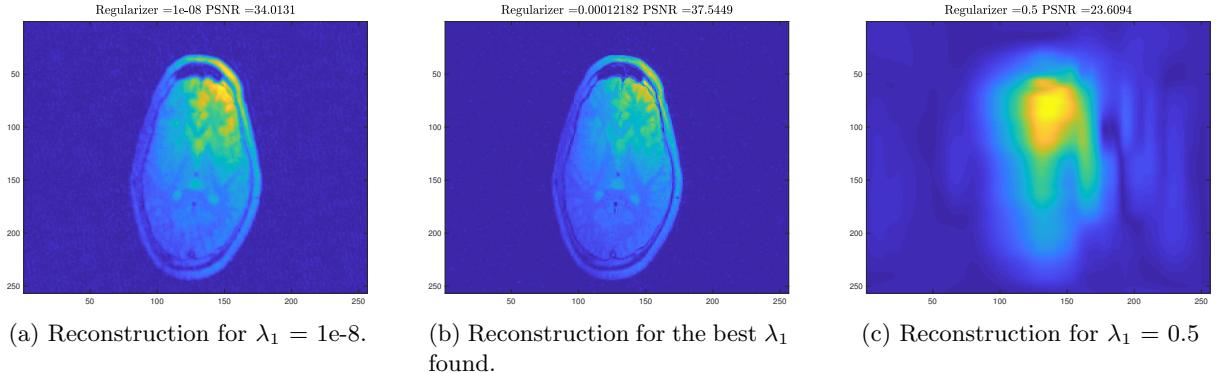


Figure 8: Comparison between reconstructed images using linear decoder.

3.4 Exercise 3.4

Comparison of different estimators.

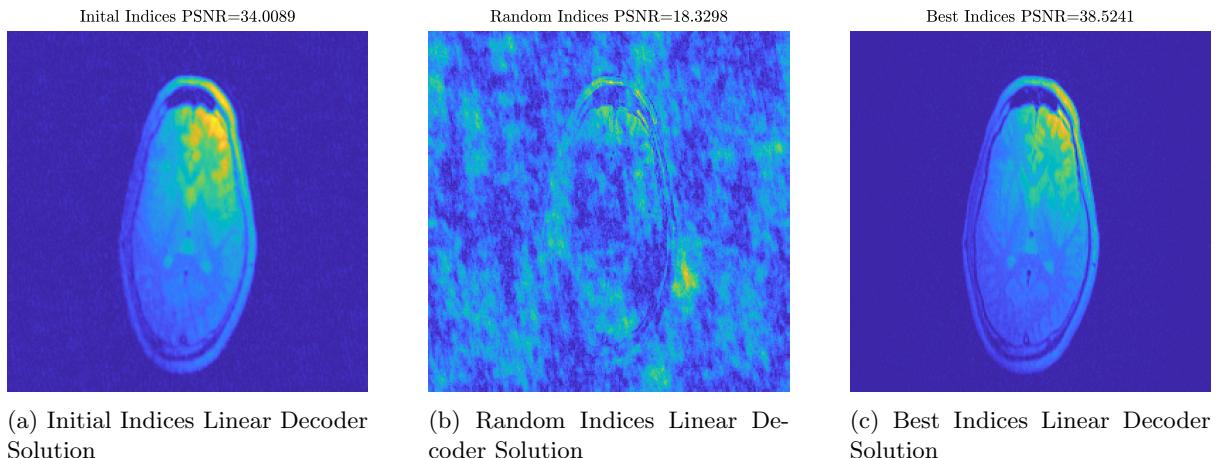


Figure 9: Comparison between reconstructed images using linear decoder.

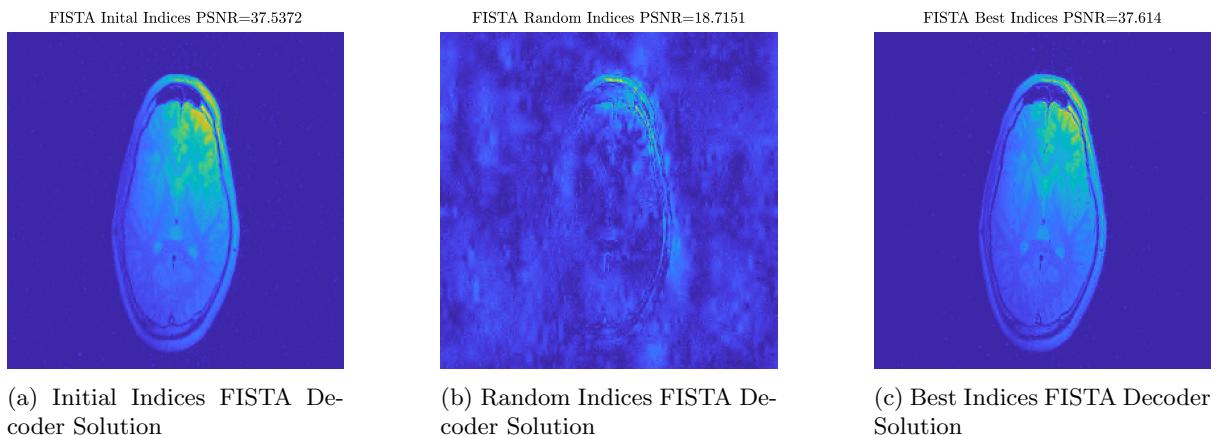


Figure 10: Comparison between reconstructed images using FISTA with restart algorithm.

The reconstructions obtained with the Linear Operator take an average of 0.005s to compute while the reconstructions obtained through the application of FISTA algorithm take an average of 7s.

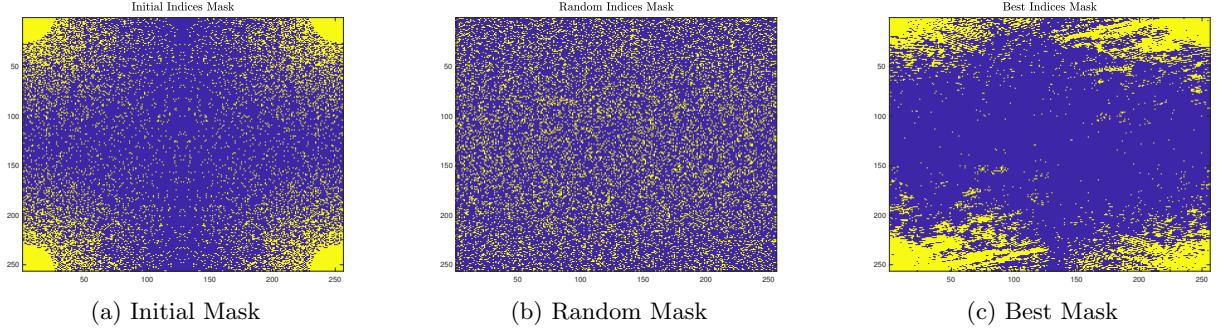


Figure 11: Comparison between masks. Pixels shown in blue corresponds to the pixels considered in the reconstruction while the yellow pixels are set to 0.

In Figures 9 and 10 it can be seen the different reconstructed images. In the case of Initial Indices, FISTA is capable of achieving a better result obtaining a better defined image. On the contrary, the image obtained with the linear decoder looks smoother and part of the details in the image are lost. In this case, if the application requires accurate results, it would be better to use FISTA as reconstruction method although the increase in computation time.

In the case where the "Best Indices" are provided, both methods achieve highly accurate results and the usage of the linear decoder would be preferred in front of FISTA due to the difference in times. Nevertheless, obtaining the best indices can be very costly and it may be better to start from a suboptimal solution and improve it with FISTA rather than computing the best indices to be able to use the linear decoder.

For the case of completely random indices, any of the two methods cannot reconstruct the image and the results obtained are far from reflecting the desired result.

4 Image in-painting

4.1 Exercise 4.1

Adapt the FISTA algorithm to solve problem 40 and problem 41. Perform a parameter sweep over the regularization parameters.

$$\min_{\alpha \in \mathbb{C}^p} \frac{1}{2} \|b - P_\Omega F W^T \alpha\|_2^2 + \lambda_1 \|\alpha\|_1 \quad (40)$$

$$\min_{\alpha \in \mathbb{C}^p} \frac{1}{2} \|b - P_\Omega F x\|_2^2 + \lambda_{TV} \|x\|_{TV} \quad (41)$$

The result of the parameter sweep is shown in Figure 12

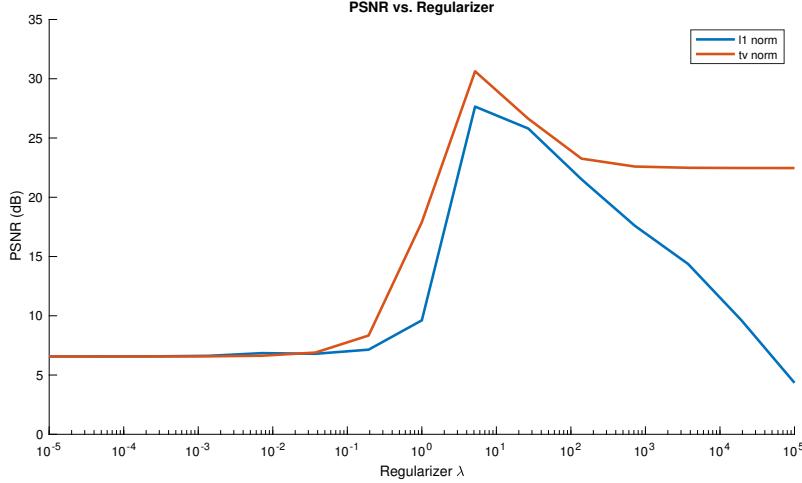


Figure 12: Parameter sweep for λ_1 and λ_{TV} regularization parameters.

The highest PSNR values are obtained for the regularizer values around $\lambda = 10$. For λ values smaller than 0.1, the PSNR remains constant around 8dB. For $\lambda \geq 0.1$ the PSNR begins to increase achieving a maximum around $\lambda = 10$. After the maximum is achieved, the PSNR drops. For l1-norm regularization, the value of PSNR decreases towards 0 while for TV-norm regularization, it decreases until achieving a lower bound on the PSNR of around 25dB.

The reconstructed images for different regularization parameters are shown in the following figures:

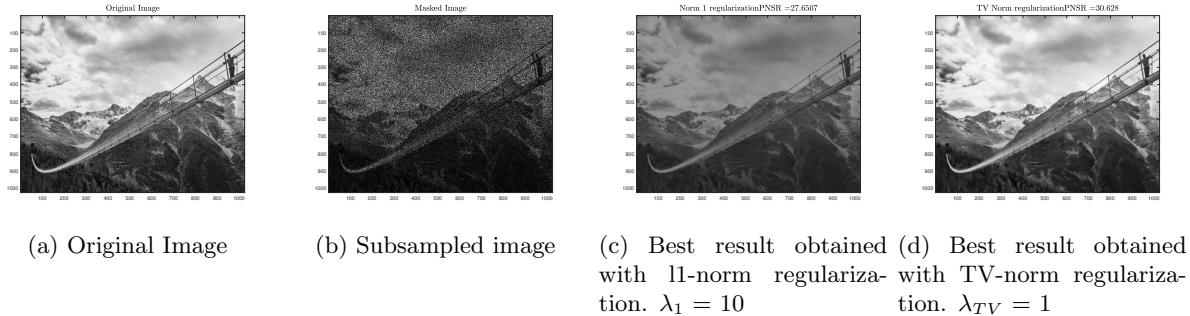


Figure 13: Comparison of original image and best results obtained through the two different methods.

The effect of extreme values in the regularizer parameter can be seen in Figure 14.

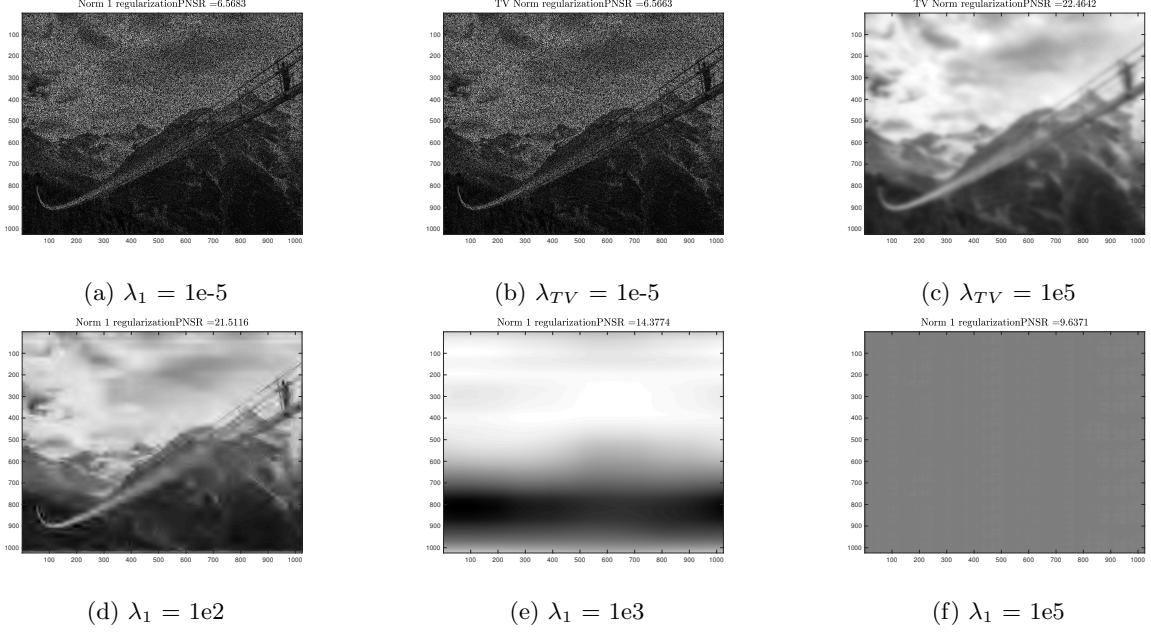


Figure 14: Reconstructed images for different values of regularization parameter. Images generated after 100 iterations for $l1$ -norm regularization and 30 iterations for TV -norm regularization.

Given the images shown in Figure 14, it can be seen that for small values of λ , there is no reconstruction at all and the obtained image is just the masked one. For very large regularization values, the image becomes too smooth in both cases. For TV -norm, the image is distorted but still can be recognized. For $l1$ -norm regularization, the image is distorted making it hard to interpret and recognize the original image. If the value of the regularizer keeps increasing the distortion is complete, ending up by fully destroying the image as it can be seen in Images d) to f) of Figure 14.

4.2 Exercise 4.2

Convergence analysis:

1. Implement the following variations of FISTA
 - a ISTA.
 - b FISTA with no restart.
 - c FISTA with fixed iteration restart.
 - d FISTA with gradient scheme restart.
2. Run FISTA with the non-monotonicity test as restart criterion for 5000 iterations with tolerance 10^{-15} to estimate the optimal solution for problem 41

The optimal value obtained is:

$$F^* = 5.096030358114446 * 10^7 \quad (42)$$

3. Study the convergence of the previously mentioned variations of FISTA. The tolerance criterion for stop iterating is $|F(x^k) - F^*| / F^* < 10^{-15}$

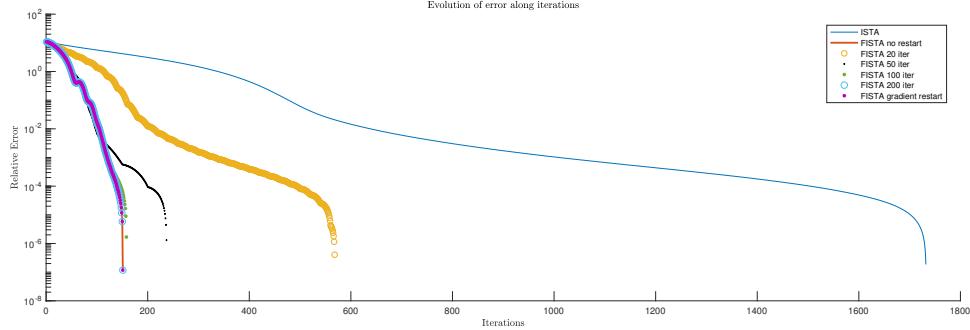


Figure 15: Evolution of the error $|F(x^k) - F^*| / F^* < 10^{-15}$ along 2000 iterations computed with different FISTA variations.

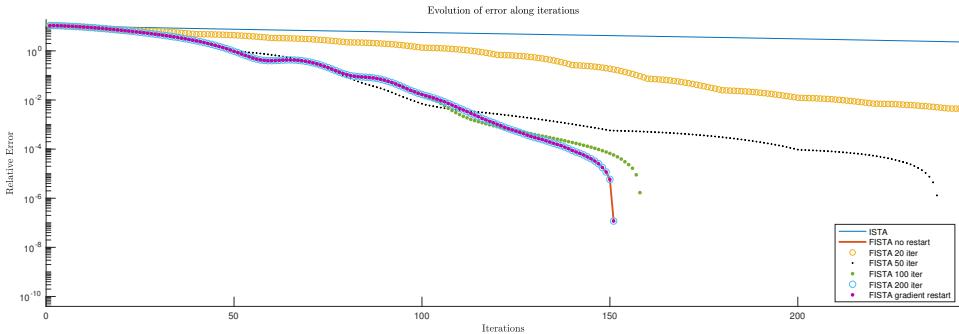


Figure 16: Evolution of the error $|F(x^k) - F^*| / F^* < 10^{-15}$ along the firsts iterations computed with different FISTA variations. Focus on the differences between the fastest algorithms.

Figures 15 and 16 show the evolution of the error for different variations of FISTA taking as optimal value of F the one obtained through 5000 iterations of FISTA with non-monotonicity as restart criterion.

The slowest algorithm to converge is ISTA. This behaviour is expected as ISTA does not implement any momentum term to accelerate the descent. The speed of convergence also increases as the frequency of restart of the momentum decreases. Finally, the performance of restart after every 200 iterations, restart with gradient scheme and no restart are very similar in terms of evolution of the error along iterations. Nevertheless, in terms of time, using the gradient scheme implies realizing a higher number of operations which will cause an increase in the total amount of time required to converge. In that case, given there is no difference in terms of performance that motivates its usage, it will be more efficient choosing one of the algorithms that do not require gradient evaluation. All variants of FISTA achieve a relative error smaller than 10^{-15} in less than 2000 iterations.

If the attention is focus on the performance of the fastest algorithms. The fact that FISTA without restart and FISTA with gradient scheme restart perform equally indicates that the momentum term has not achieved such a value that causes the gradient to be non-monotone and therefore needs to be restarted. Finally, the variation of FISTA with fixed restart every 200 iterations has the same performance as the other two algorithms as it reaches the stopping criterion before the first restart needs to be performed.

4. Replace F^* with $F^\natural(x^\natural)$ and plot a graph $\log(|F(x^k) - F^\natural| / F^\natural)$ against k using the previously mentioned variations of FISTA for 1000 iterations.

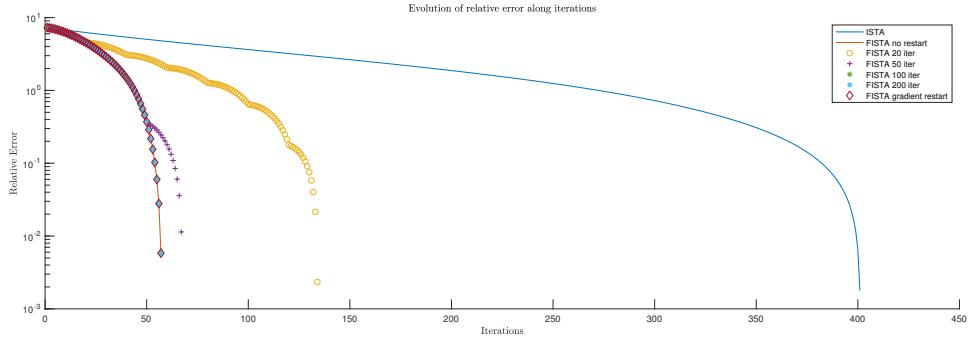


Figure 17: Evolution of the error $|F(x^k) - F^\natural| / F^\natural < 10^{-15}$ along 1000 iterations computed with different FISTA variations.

The value obtained as $F^\natural = 7.263059984119681 * 10^7$ which is higher than F^* . This happens as the original image has a greater l1-norm than the optimal value obtained, causing the final value of F to be higher.

Given a higher value of F as optimal, there is an early achievement of a low relative error. This does not allow to achieve a regime where differences in the performance of FISTA without restart, with restart every 100 iterations and every 200 iterations and with restart based on gradient scheme arise. The 4 algorithms reach the stopping criterion without performing any restart in the momentum term, behaving the same way until reaching the stop criterion.

On the other hand, the variations of FISTA with restart every 20 and every 50 iterations are slowed as they are being restarted while the momentum term was still accelerating the convergence in a proper way. Finally, again ISTA is the variation with lower convergence rate, taking around 8 times more iterations than the faster alternatives to achieve the a relative error $> 10^{-15}$.