Home-Work #2: Image Inverse Problems with Iterative-Shrinkage Algorithms

In this home-work assignment we will use several iterative shrinkage algorithms for recovering a synthetic image from its measurements. The background material for this homework can be found in chapter 6 in the book.

Please email a report for the required steps described below to the lecturer, along with your Matlab code and data. Please create one script called 'HW2-all.m' that goes through the various stages of the tasks and presents the results. This script may call other functions in the same directory if you prefer to work this way.

Please refrain from using existing code, as the purpose of this HW is to expose you to the details of the studied algorithms.

Note that there are some similarities between this and Homework #1, and you are welcome to use your own written code when necessary. However, be careful – few key parameters have changed and you should be wise and revise your code, so as to obtain efficient run-time.

Part A - Data Construction

In order to handle the reconstruction task in this assignment, throughout this work we will be minimizing the function

\[ f(x) = \frac{1}{2} \|y - CDx\|_2^2 + \lambda \|x\|_1, \]

using iterative shrinkage algorithms. Here are the ingredients involved:

- The operator \( D \) is the “dictionary” (of size \( 100^2 \)-by-\( L \)) under which the ideal image is known to be sparse. You can construct this matrix in one of two ways:
  - As in Homework #1, create its atoms one by one – start with a zero image of size 100-by-100 pixels, insert to it a unit-value rectangle of random size (in the range 5-20 pixels), position (uniformly spread in the area of the image), and sign. Normalize each atom to have a unit norm. Gather this way \( L = 5 \cdot 100^2 \) such atoms, and verify for each that it is not empty or nearly so (with less than 40 non-zeros).
  - Create all the above atoms by systematically sweeping through all possible positions (with jumps of 3 pixels), and sizes in the range 5-15 (with jumps of 2), and then discarding of atoms that are too empty. Normalization is mandatory here as well. In this case, \( L \) will be smaller than 50,000.

- **Advice 1**: Here and throughout the HW assignment, you MUST use sparse matrices and vectors rather than full ones.
• The operator $C$ will have three different forms, and all your experiments should be repeated for each of these options:

  o **Case 1: Inpainting.** In this case $C$ is of size $(p \cdot 100^2)$-by-$100^2$ ($p<<1$) that stands for random choice of pixels in the image. $C$ is created just like in Home-Work #1. The value of $p$ should be 0.1 (for 10% of the image pixels).

  o **Case 2: Deblurring.** In this case this matrix is square of size $100^2$-by-$100^2$ that stands for a circulant and separable blur of the image. We start with a 1d blur of length 11-taps of the form

    $$h[k] = c \cdot \exp \left\{ \frac{-k^2}{2} \right\} \text{ where } -5 \leq k \leq 5,$$

  where $c$ is set such that the sum of this vector is 1. $C$ should be created by first generating a 100-by-100 circulant (sparse) matrix $C_1$ that represents this 1d blur, and then creating $C$ by a Kronecker product of this matrix by itself, $C = C_1 \otimes C_1$.

  o **Case 3: Compressed-Sensing.** In this case this matrix is of size $(p \cdot 100^2)$-by-$100^2$ ($p<<1$) that stands for random projections of the image. $C$ is created by drawing a random matrix of the desired size, with entries that are i.i.d. zero mean Gaussian (and normalized rows). The value of $p$ should be 0.1 (10% of the image pixels).

• Generate a Mondrian-like synthetic image $y_0$ of size 100-by-100 by randomly generating a sparse vector $x_0$ of length $5 \cdot 100^2$ with 30 non-zeros (random locations and random values), and then multiply by $D$. This will serve as our original image we aim to recover.

• Create the measured data vector $y$ by the equation $y = CX_0 + n$, where $n$ is a white zero-mean Gaussian additive noise with STD=$\sigma$ (make it 0.02 of the dynamic range of the measurements vector $CDx_0$). An example is given below for the ideal image and its measurements in the case of blur.

![The original image $y_0$ and its blurred and noisy version.](image)

• **Advice 2:** When computing $CDx_0$, it would be wise to compute it by $C(Dx_0)$ so that the matrix $CD$ is not computed, nor stored. As opposed to Homework #1, this time this trick is critical.

• **Advice 3:** The multiplication $CD$ may create zero columns in the inpainting case. Avoid this situation by randomly drawing $C$ and checking that this does not happen.
Part B – Initial Results

- "Oracle" reconstruction: Compute the resulting image that would minimize the $L_2$ term $\|y - CDx\|_2^2$, assuming that the TRUE support of $x_0$ is known. Present the resulting image and compute its PSNR quality by (dynamic range is the difference between the maximum and the minimum in the original image):

$$PSNR = 10\log_{10} \left( \frac{\text{dynamic range}^2}{\frac{1}{100^2} \|y - y_0\|_2^2} \right)$$

- $L_2$-based reconstruction: Find the $L_2$-shortest vector $\hat{x}$ that minimizes the term $\|y - CDx\|_2^2$. This is similar to the oracle experiment but without forcing $\hat{x}$ to be sparse. Note that this stands for a possible solution corresponding to $\lambda = 0$ in the function

$$f(x) = \frac{1}{2} \|y - CDx\|_2^2 + \lambda \|x\|_1.$$ 

This should be done by programming the Conjugate-Gradient algorithm (Wikipedia will help you) and initializing this iterative solver with zero. Run it for 500 iterations. Present the resulting image and compute its PSNR quality.

Part C – Iterative Shrinkage

- As said before, we will solve the three inverse problems by minimizing the function described above,

$$f(x) = \frac{1}{2} \|y - CDx\|_2^2 + \lambda \|x\|_1$$

where this will be done using the following algorithms:

  o SSF (also known as ISTA) algorithm. You will need to find the parameter $c = \lambda_{\text{max}}(D^TC^TCD)$ and this can be done using the Power-Method (Wikipedia).

  o SSF coupled with a line-search. Implement the Armijo rule (yes, Wikipedia once again).

  o FISTA – see http://iew3.technion.ac.il/~becka/papers/71654.pdf. Do not be alarmed – the algorithm is relatively simple to implement, and it works fast.

- Initialization: you should initialize all the above runs with $\hat{x}_0 = 0$.

- Choice of $\lambda$: Start by seeking a good value $\lambda$ for the problem. Use the SSF algorithm for a fixed and very large number of iterations (e.g. 2000), and seek a value of $\lambda$ that leads to sufficiently sparse (e.g. having 200-300 non-zeros) estimate $\hat{x}_k$ while also providing good PSNR (better than the LS one).

- Run all the above algorithms for the value of $\lambda$ found and show comparative graphs for the three algorithms showing

  (i) the function value as it varies through the iterations; and
(ii) the PSNR as a function of the iteration.

- At any stage of the iterative process, the recovered image is $D\hat{x}_k$. We are interested in the best PSNR result, compared to the ideal image we started from. Show the final images obtained from the above runs.

- Apply the debiasing idea as described in the end of chapter 6. This is very closely related to the oracle test you did above, but instead of taking the true support, you use the one found by the iterative solution above. Note that this idea is relevant only if the number of non-zeros in $\hat{x}_k$ is relatively small, so make sure that this is the case.

- **Advice 5:** As the iterations proceed, the PSNR does not have to consistently increase. In this homework assignment we will concentrate on the value obtained near convergence of the process. You should be aware of the fact that there are methods to automatically detect the iteration in which the PSNR is at its peak and stop the algorithm – we will not deal with this here.

- **Do not forget** – Repeat all the above for each of the three problems – inpainting, deblurring, and compressed-sensing, and organize the results in a clear and elegant way for each task, so as to see the comparative results for each of the experiments.