Mathematical methods for Image Processing

Hands-on session
1 and 2

Solving a tykhonov model in image denoising :
Comparison of three Iterative algorithms

During this hands-on session, we will compare the performances of different algorithms for smooth optimization. To do this, we will consider a denoising problem: our data v is of the form

$$v = u + b,$$

where $u \in \mathbb{R}^{N^2}$ is the image we are looking for; $v \in \mathbb{R}^{N^2}$ is the data at our disposal and $b \in \mathbb{R}^{N^2}$ is the realization of a Gaussian white noise.

The model we are considering is to minimize the energy

$$E(w) = \sum_{i,j=0}^{N-1} |\nabla w_{i,j}|^2 + \lambda \sum_{i,j=0}^{N-1} (w_{i,j} - v_{i,j})^2,$$

where $w \in \mathbb{R}^{N^2}$ and $\lambda \geq 0$, with

$$\nabla w_{i,j} = \begin{pmatrix} (D_x w)_{i,j} \\ (D_y w)_{i,j} \end{pmatrix} = \begin{pmatrix} w_{i+1,j} - w_{i,j} \\ w_{i,j+1} - w_{i,j} \end{pmatrix}$$

for $(i,j) \in \{0,\dots,N-1\}^2$ (we assume w periodic).

Exercice 1.

(1) Calculate the adjoint operator D_x^* (and D_y^*) of D_x (and D_y).
(2) Calculate a close form formula for: - the linear operator $A : \mathbb{R}^{N^2} \longrightarrow \mathbb{R}^{N^2}$; - the constant image $B \in \mathbb{R}^{N^2}$; - and the constant C (independent of w), such that for all $w \in \mathbb{R}^{N^2}$,

$$E(w) = \langle Aw, w \rangle + \langle B, w \rangle + C,$$

where $\langle .,. \rangle$ stands for the usual inner product in $\mathbb{R}^{N^2}$.

(3) Check that A is self-adjoint. Check that the operator A is a convolution[1] with a kernel $h \in \mathbb{R}^{N^2}$ and give a closed form expression for h.

(4) Check that the discrete Fourier transform[2] of h has the form

$$\hat{h}_{k,l} = \lambda + 4 - 2\left(cos(\frac{2\pi k}{N}) + cos(\frac{2\pi l}{N})\right) \qquad \forall k,l = 0..N-1.$$

NB : We remind that we have for all $k,l = 0..N-1$

$$(\widehat{h * w})_{k,l} = \hat{h}_{k,l}\hat{w}_{k,l}.$$

We also remind that, given the above properties, the eigenvalues of A are $(\hat{h}_{k,l})_{0 \leq k,l < N}$.

(5) Calculate the gradient and the Hessian of E : $\nabla E(w)$ and $\nabla^2 E(w)$.

(6) Deduce from the previous question upper bounds $\alpha > 0$ and $L > 0$ such that for all w and $w' \in \mathbb{R}^{N^2}$

$$\alpha \|w'\|_2^2 \leq \left\langle \nabla^2 E(w) w', w' \right\rangle \leq L \|w'\|_2^2.$$

Make the connection with the hypotheses guarantying the convergence of the gradient algorithm.

(7) Deduce, from the preceding questions, an algorithm based on the Fast Fourier Transform minimizing E.

Exercice 2.

(1)
- Go to the website http://www.math.univ-toulouse.fr/∼fmalgouy/index.html
- Download the packages needed for hands-on session as well as the file *tp1.zip*.
- Uncompress the archives. The zip file tp1.zip contains Scilab files.
- Open the file "toolbox_signal/imageplot.sci".
  - (a) Comment the line : "if MSDOS, xbasr(), end"
  - (b) Comment the line : "if xget('pixmap')==1"
  - (c) Line 29 of the function "imshow", change for strf = '020'
- Launch Scilab, change the current environment (in the "file" menu) and open "script.sce".

[1] We remind that the convolution is defined by :

$$\forall m,n = 0..N-1, (h * w)_{m,n} = \sum_{m',n'=0}^{N-1} h_{m',n'} w_{m-m',n-n'}.$$

[2] We remind that the Discrete Fourier Transform is defined by :

$$\forall k,l = 0..N-1, \hat{h}_{k,l} = \sum_{m,n=0}^{N-1} h_{m,n}\, e^{-2i\pi \frac{km+ln}{N}}.$$

(2) Use the function *rand* to add a Gaussien noise of standard deviation 10 to the image *barb* (scaled such that its gray level are between 0 and 255).

(3) Read the program contained in *quadratique_exacte.sci*. What does it compute? Explain your answer and make connections with Exercise 1.

(4) Use the function *quadratique_exacte* to denoise the noisy image obtained at question 2, for the values $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1$, $\lambda = 10$, $\lambda = 100$. Explain the role of the parameter λ. According to your visual inspection of the results, what value of λ would you recommend?

(5) Read the program *quadratique_approx.sci*. What does it compute? Explain your answer and make connections with Exercise 1 and the algorithms studied during the lectures.

(6) Use the function *quadratique_approx* to denoise the noisy image obtained at question 2, for the values $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1$, $\lambda = 10$, $\lambda = 100$. For each value of λ, how many iterates need the algorithm to converge? Make connections with the exercise 1 and explain this behavior.

(7) Copy *quadratique_approx.sci* in a file *quadratique_approx1.sci* and modify it to use the constant step-size rule and the Armijo step-size rule.

(8) Compare the different step-size rules when denoising the image obtained at the question 2, for $\lambda = 0.1$ and $\lambda = 1$.

Exercice 3.

(1) Copy the file *quadratique_approx1* to a file *quadratique_approx2*. Then modify the program contained in *quadratique_approx2* so that it follows the conjugate gradient algorithm described in Algorithm 1. (The operator A is as in exercice 1.)

NB : For a quadratic function, the conjugate gradient algorithm converges in at most N^2 iterations.

Algorithm 1 Conjugate gradient algorithm

Entry: Entry needed for computing E and ∇E
Output: Approximation of a minimizer : w^*

Initialize w_0
Initialize $d_0 = \nabla E(w_0)$
Initialize $w_1 = w_0 - \frac{\langle d_0, d_0 \rangle}{2\langle A d_0, d_0 \rangle}\, d_0$
While Not converged **Do**
  Compute $d_k = \nabla E(w_k) + \frac{\|\nabla E(w_k)\|^2}{\|\nabla E(w_{k-1})\|^2}\, d_{k-1}$
  Compute the step-size $t_k = \frac{\langle \nabla E(w_k), d_k \rangle}{2\langle A d_k, d_k \rangle}$ (It is the steepest descent step-size d_k)
  Update : $w_{k+1} \leftarrow w_k - t_k\, d_k$
End while

---

(2) Use this algorithm to denoise the image obtained at the question 2, for $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1$, $\lambda = 10$, $\lambda = 100$. For each of these λ values, how many iterations does it take the algorithm to converge? Compare these results with those obtained in the previous exercise and comment on this comparison.

Exercice 4.

(1) Copy the file *quadratique_approx1* to a file *quadratique_approx3*. Then modify the program contained in *quadratique_approx3* so that it follows the accelerated gradient algorithm described in Algorithm 2.

NB : The conjugate gradient algorithm has a convergence rate of $\frac{1}{k^2}$, where k is the iteration number.

Algorithm 2 Accelerated gradient algorithm

Entry: Entry needed for computing E and ∇E
Output: Approximation of a minimizer : w^*

Compute a step-size $t = \frac{1}{L}$ (where L is the Lipschitz constant of the gradient)
Initialize w_0
Initialize $v_1 = w_0$
While Not converged **Do**
  Compute $w_k = v_k - t\nabla E(v_k)$
  Update : $v_{k+1} = w_k - \frac{k-1}{k+2}(w_k - w_{k-1})$
End while

---

(2) Use this algorithm to denoise the image obtained at the question 2, for $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1$, $\lambda = 10$, $\lambda = 100$. For each of these λ values, how many iterations does it take the algorithm to converge? Compare these results with those obtained in the previous exercises and comment on this comparison.