MEDICAL PHYSICS / BIOMEDICAL ENGINEERING 574

Homework Set 3 - SOLUTIONS

Diego Hernando

Due date: Monday, March 11, 2019
(Due by the end of the day, either electronically or on paper. Note that your Matlab implementation of the algorithms needs to be handed in electronically even if the rest of the homework is handed in on paper)

Implement Steepest Descent and Conjugate Gradients for Image-Scale Optimization

All problems in this homework set have a common structure, with increasing complexity. In each case, you should implement two algorithms (SD and CG) to solve an image reconstruction optimization problem, which can be written as follows:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{C}\mathbf{x}\|_2^2 \tag{1}$$

where $\hat{\mathbf{x}}$ is our desired reconstructed image, $\mathbf{b}$ is our measured data, $\mathbf{A}$ is our 'system' matrix, that relates our image to the corresponding measurements, $\mathbf{D}$ is a filtering matrix that calculates (possibly weighted) spatial finite differences on our image, in order to perform smoothness-based regularization. Finally, λ is our regularization parameter that balances data fidelity and smoothness.

For each of the formulations described below, run our two algorithms (SD and CG) for 100 iterations, and report/plot:

- The (logarithm of the) value of the cost function ($\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{C}\mathbf{x}\|_2^2$) as a function of iterations (this should be decreasing for both algorithms, but different between algorithms). Since the value of the cost function will likely change a lot, make sure to plot its logarithm vs the iteration number.
- The final image after the iterations are done for each of the algorithms (display it using a common colormap for both algorithms).
- The relative performance of SD and CG (ie: how rapidly do they converge)

1 Reconstruction from Partial Fourier Samples without Regularization (20 points)

Write a script that loads the data in `hw3_problem1.mat`, and reconstructs an image of size 320×320 pixels from the samples in array `b`. The samples are DFT samples of the image, taken at the locations specified by the mask in array `mask`. In other words, the matrix $\mathbf{A}$ in our general formulation (equation 1 above) is

$$\mathbf{A} = \mathbf{M}\,\mathbf{F}$$

where $\mathbf{F}$ performs a 2D DFT (including `fftshifts` and `ifftshifts`) and the matrix $\mathbf{M}$ is a sampling matrix that selects a subset of samples from the resulting 320×320 2D DFT. The selected subset of samples is given by the binary array `m`.

In this case, there is no regularization term (ie: $\lambda = 0$).

SOLUTION: Please see figure 1 for an illustration of the implementation of this reconstruction. Note that in this case, both algorithms converge in a single iteration. The reason for this single-step convergence is the fact that the gradient is pointing directly towards the least-squares solution. In other words, the gradient is proportional to the zero-filled image reconstruction $\mathbf{F^{-1}M^{H}b}$ (which is also a LS solution to this problem): remember that when we initialize with all zeroes, the gradient at $\mathbf{x}_0$ will be $\mathbf{F^{H}M^{H}b} = N^2\mathbf{F^{-1}M^{H}b}$ where N is our image size (320 pixels). Therefore, both the SD and the CG algorithms jump directly to the LS 'zero-filled' solution.

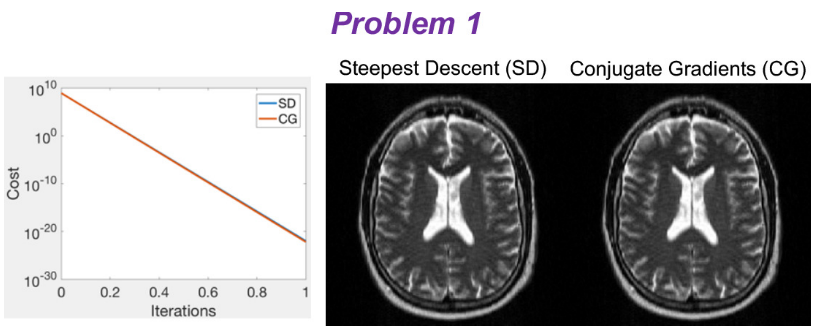

Figure 1: Image reconstruction from undersampled Fourier data using a least-squares formulation without regularization, using both a steepest descent (SD) algorithm as well as a conjugate-gradient (CG) algorithm. Both algorithms were initialized with an all-zeros initial guess, and converge in a single iteration to the least-squares solution.

2 Reconstruction from Partial Fourier Samples with Smoothness Regularization (40 points)

Write a script that loads the data in `hw3_problem2.mat`, and reconstructs an image (320×320 pixels) from the samples in array `b`. Next we describe the data and smoothness terms:

- *Data term:* Similar to the previous case, the samples in `b` are DFT samples of the image, taken at the locations specified by the mask in array `mask`. In other words, the matrix $\mathbf{A}$ in our general formulation (equation 1 above) is

$$\mathbf{A} = \mathbf{M}\mathbf{F}$$

  where $\mathbf{F}$ performs a DFT (including `fftshifts`) and the matrix $\mathbf{M}$ is a sampling matrix that selects a subset of samples from the resulting 320×320 DFT. The selected subset of samples is given by the binary array `m`.

- *Smoothness term:* The smoothness term is given by $\lambda\|\mathbf{C}\mathbf{x}\|_2^2$, where $\mathbf{C} = \mathbf{D}$ calculates finite differences in 2D, as described in lecture 14 of this course (where we denoted it as `D2` in the Matlab code to indicate its 2D nature).

You should repeat this optimization for several different values of λ:

- $\lambda = 10^0$
- $\lambda = 10^2$
- $\lambda = 10^4$
- $\lambda = 10^6$
- $\lambda = 10^8$

and, in each case, report the cost function as a function of iterations (plot), as well as the final estimated image (display). This should be done for both the SD and the CG algorithms.

Question: How does the final value of the cost function change as λ increases? How does the visual appearance of the image change as λ increases?

SOLUTION: Please see figure 2 for an illustration of the implementation of this reconstruction. Note that in this case, the algorithms require more iterations to converge (ie: the gradient does not point directly to the solution). Also, the value of the final cost function goes up with increasing λ, as does the smoothness of the reconstructed image. This implies that the cost function for problem 2 (upon convergence) should generally be larger than that for problem 1, since problem 1 simply corresponds to problem 2 with $\lambda = 0$. For large values of λ, SD becomes quite slow and would require many iterations to converge, whereas CG converges in fewer iterations. Further, SD requires a relatively time comsuming line search at each iteration, so each iteration of SD is substantially slower than each iteration of CG. Therefore, CG is a better choice for this problem, particularly as λ increases.

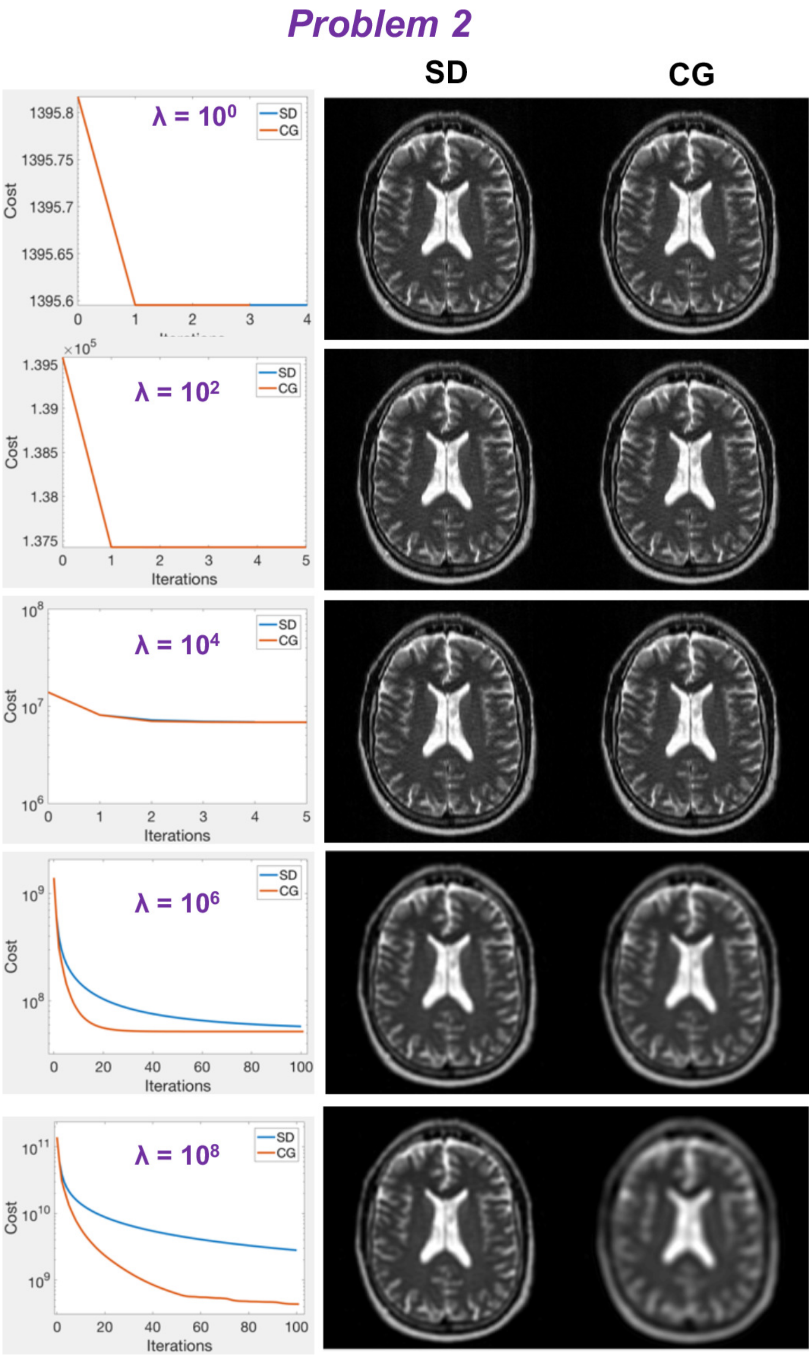

Figure 2: Image reconstruction from undersampled Fourier data using a least-squares formulation with unweighted smoothness regularization, using both a steepest descent (SD) algorithm as well as a conjugate-gradient (CG) algorithm. Both algorithms were initialized with the zero-filled solution as initial guess.

3 Reconstruction from Partial Fourier Samples with Anatomically-Informed Smoothness Regularization (40 points)

Write a script that loads the data in `hw3_problem3.mat`, and reconstructs an image of size 320×320 pixels from the samples in array `b`. Next we describe the data and smoothness terms:

- *Data term:* Similar to the previous case, the samples in `b` are DFT samples of the image, taken at the locations specified by the mask in array `mask`. In other words, the matrix $\mathbf{A}$ in our general formulation (equation 1 above) is

  $$\mathbf{A} = \mathbf{M}\mathbf{F}$$

  where $\mathbf{F}$ performs a DFT (including `fftshifts`) and the matrix $\mathbf{M}$ is a sampling matrix that selects a subset of samples from the resulting 320×320 DFT. The selected subset of samples is given by the binary array `m`.

- *Smoothness term:* The smoothness term is given by $\lambda\|\mathbf{C}\mathbf{x}\|_2^2$, where $\mathbf{C} = \mathbf{W}\mathbf{D}$ calculates weighted finite differences in 2D. The matrix $\mathbf{D}$, similarly to the previous problem, is as described in lecture 14 of this course (where we denoted it as `D2` in the Matlab code to indicate its 2D nature). The weighting matrix $\mathbf{W}$ is a diagonal matrix with weights derived from an image of the same anatomy as our desired image: it specifies in which locations we will penalize roughness, and in which locations (those with expected edges) we will not penalize roughness. The diagonal elements of this matrix can be found in array `w`.

You should repeat this optimization for several different values of λ:

- $\lambda = 10^0$
- $\lambda = 10^2$
- $\lambda = 10^4$
- $\lambda = 10^6$
- $\lambda = 10^8$

and, in each case, report the cost function as a function of iterations (plot), as well as the final estimated image (display). This should be done for both the SD and the CG algorithms.

Question: How does the final value of the cost function change as λ increases? How does the visual appearance of the image change as λ increases? How do the cost function and visual appearance of the image in this problem compare to those in the previous problem?

SOLUTION: Please see figure 3 for an illustration of the implementation of this reconstruction. Note that in this case, the algorithms require more iterations than in the unregularized case to converge (ie: the gradient does not point directly to the solution). Also, the value of the final cost function goes up with increasing λ, as does the smoothness of the reconstructed image. However, because of the weighting in the regularization term, the locations with predicted edges in the image do not become smoothed even for large λ, so this weighted regularization results in substantially better image quality than the unweighted regularization from problem 2. In general, this formulation results (upon convergence) in cost function values larger than in the unregularized case (problem 1). However, the cost function values are lower than in the unweighted smoothness case (problem 2) - the reason for this being that our weights are all ≤ 1. For most values of λ, SD becomes quite slow and would require many iterations to converge, whereas CG converges in fewer iterations. Further, SD requires a relatively time comsuming line search at each iteration, so each iteration of SD is substantially slower than each iteration of CG. Therefore, CG is a better choice for this problem, particularly as λ increases.

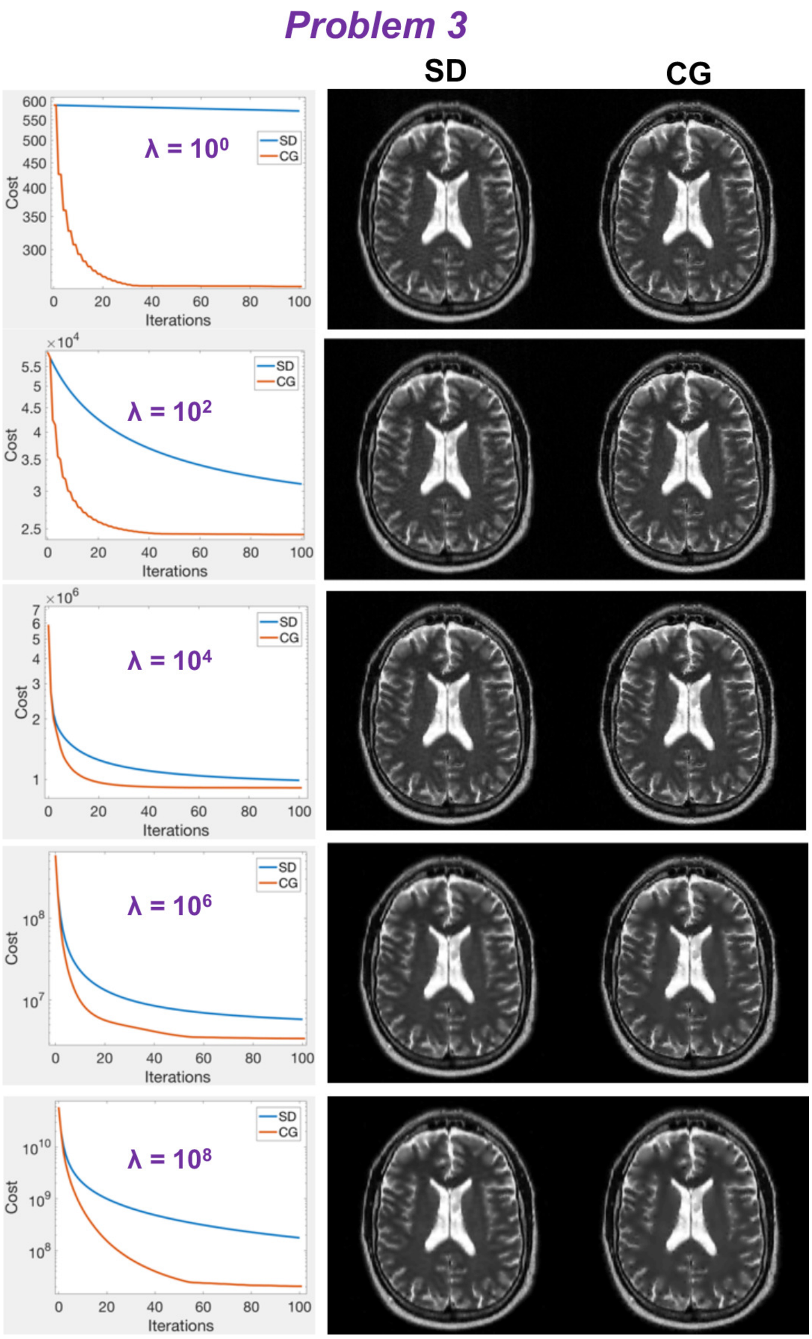

Figure 3: Image reconstruction from undersampled Fourier data using a least-squares formulation with weighted smoothness regularization, using both a steepest descent (SD) algorithm as well as a conjugate-gradient (CG) algorithm. Both algorithms were initialized with the zero-filled solution as initial guess.