CS-736 Assignment:
Statistical Image Modeling, Image Denoising

Instructor: Suyash P. Awate

1. **(25 points) Bayesian Denoising of a Phantom Magnetic Resonance Image.**

   Get the 2D noiseless image and the 2D noisy image in the "data" folder.

   Implement a **maximum-a-posteriori Bayesian image-denoising** algorithm that uses a suitable noise model (i.i.d. Gaussian or independent Rician) coupled with a MRF prior that uses a 4-neighbor neighborhood system (each pixel has 4 neighbors: left, right, up, down; the neighborhood wraps around at image boundaries) that has cliques of size *no* more than 2.

   Use gradient ascent optimization with dynamic step size; implementing a fixed step size isn't acceptable. Ensure that the value of the objective function, i.e., the log posterior, at each iteration increases. Use the noisy image as the initial estimate.

   Use 3 different MRF priors where the potential functions $V(x_i, x_j) := g(x_i - x_j)$ underlying the MRF penalize the difference between the neighboring voxel values x_i, x_j as follows (see class notes for details). You may rely on the "circshift()" function in Matlab when computing differences between every pixel in the image and its neighbors.

   Introduce a parameter $\alpha \in [0, 1]$ to control the weighting between the prior (weight α) and the likelihood (weight $1 - \alpha$).

   Specifically, implement the following functionality as part of the denoising algorithm:

   (a) (2 points)
   A suitable noise model. You don't need the noise level because that parameter can be absorbed in $1 - \alpha$ that you'll tune manually (Tuning α essentially manipulates the noise level, in case of the likelihood. So we can ignore the noise level σ when tuning α manually. Use $\sigma = 1$).

   (b) (2 points)
   MRF prior: Quadratic function: $g_1(u) := |u|^2$.

   (c) (2 points)
   MRF prior: Discontinuity-adaptive Huber function: $g_2(u) := 0.5|u|^2$, when $|u| \leq \gamma$ and $g(u) := \gamma|u| - 0.5\gamma^2$, when $|u| > \gamma$, where $0 < \gamma < \infty$ is a constant.

   (d) (2 points)
   MRF prior: Discontinuity-adaptive function: $g_3(u) := \gamma|u| - \gamma^2 \log(1 + |u|/\gamma)$, where $0 < \gamma < \infty$ is a constant.

   For each MRF prior, manually tune the parameters α and γ (where applicable) to denoising the noisy image in order to achieve the least possible relative root-mean-squared error (RRMSE). The RRMSE for 2 images A and B is defined as :
   RRMSE$(A, B) = \sqrt{\sum_p (|A(p)| - |B(p)|)^2} / \sqrt{\sum_p |A(p)|^2}$, where the summation is over all pixels p; always use the noiseless image as A.

   Report the following:

   (a) (0 point)
   Report the RRMSE between the noisy and noiseless images.

(b) (8 points)

Report the optimal values of the parameters and the corresponding RRMSEs for each of the 3 denoising algorithms. For each optimal parameter value reported (for each of the 3 denoising algorithms), give evidence of the optimality of the reported values by reporting the RRMSE values for two nearby parameter values (around the optimal) at plus/minus 20% of the optimal value. That is, if a^*, b^* are the optimal parameter values, then report:
a^*, b^*, RRMSE(a^*, b^*),
RRMSE$(1.2a^*, b^*)$, RRMSE$(0.8a^*, b^*)$,
RRMSE$(a^*, 1.2b^*)$, RRMSE$(a^*, 0.8b^*)$.
(Tip: the optimal values for α might be very close to extreme limits of the allowed range. Be aware of that possibility.) .

(c) (6 points)

Show the following 5 images (at each pixel, show the magnitude of the pixel value; use a jet colormap) in the report using exactly the same colormap (i) Noiseless image, (ii) Noisy image,

(iii) Image denoised using quadratic prior $g_1(\cdot)$ and optimal parameter tuning;

(iv) Image denoised using Huber prior $g_1(\cdot)$ and optimal parameter tuning;

(v) Image denoised using discontinuity-adaptive prior $g_3(\cdot)$ and optimal parameter tuning.

(d) (3 points)

Show the plots of the objective-function values (vertical axis) versus iteration (horizontal axis) corresponding to each of the 3 denoised results in (iii), (iv), and (v) above.

2. **(9=6+3 points) Bayesian Denoising of a Brain Magnetic Resonance Image.**

   Repeat the same task for the brain MRI image in the “data” folder.

   Show the images and the plots.

3. **(9=6+3 points) Bayesian Denoising of a RGB Microscopy Image.**

   Repeat the same task for the microscopy image in the "data" folder (assuming a Gaussian noise model as an approximation), but with the following MRF prior functions:

   (A) where the penalty between two vector-valued neighboring pixels is the squared-L2-norm of their vector difference;

   (B) where the penalty between two vector-valued neighboring pixels is the L2 norm of their vector difference;

   (C) where the penalty between two vector-valued neighboring pixels is the (Huber-regularized) L1 norm of their vector difference.

   (a) (6 points) Show the following 5 images (at each pixel, show the magnitude of the pixel value; use a jet colormap) in the report using exactly the same colormap (i) Noiseless image, (ii) Noisy image,

   (iii) Image denoised using prior-A and optimal parameter tuning,

   (iv) Image denoised using prior-B and optimal parameter tuning,

   (v) Image denoised using prior-C and optimal parameter tuning.

   In which of the three cases (i.e., A,B,C) do you observe edge preservation ? and why ?

   (b) (3 points) Show the plots of the objective-function values (vertical axis) versus iteration (horizontal axis) corresponding to each of the 3 denoised results above.

4. **(25 points) Dictionary Learning on Image Patches, Followed by Image Denoising.**

Get the 2D chest-CT image in the "data" folder.

Consider a dictionary matrix D with K columns as its atoms, where the k-th column d_k, for any k, has $\|d_k\|_2 \leq 1$ (hard constraint).

Consider the following dictionary learning formulation for some set of image patches $\{x_i\}_{i=1}^{I}$, under the hard constraint stated above:

$$\arg\min_D \min_r \sum_{i=1}^{I} \|x_i - Dr_i\|_2^2 + \lambda \|r_i\|_p^p,$$

where each r_i is a column vector (of length K) containing the coefficients used in the representation of image patch x_i, and $p > 0$ parameterizes the norm/quasi-norm of the vector r_i.

(a) (9 points)

Implement a function to learn the dictionary D for image patches of size 8×8 pixels. The dictionary D should have the number of atoms $K = 64$. Your optimization algorithm should be the same for any suitable choice of p.

For computational efficiency, the learning must only use a subset of image patches that have a sufficiently large variance of intensities withing the patch (e.g., avoiding constant-intensity patches). Nevertheless, the final dictionary must indeed be able to model and fit all kinds of patches. With these design motivations, engineer a strategy to create the subset of patches $\{x_i\}_{i=1}^{I}$ (within the given image) that will be used to learn D, and justify your strategy.

(b) (6 points)

Use your function to learn four different dictionaries for values of $p = 2$, $p = 1.6$, $p = 1.2$, $p = 0.8$.

- (2 points)

In each of the 4 cases, during the optimization process, show the graph of the objective function versus iterations.

- (2 points)

In each of the 4 cases, before and after the optimization process, show the atoms used as the initial estimate (must be same for all cases) and the optimal estimates. Describe any trend that you observe.

- (2 points)

In each of the 4 cases, after the optimization process, show the histogram of coefficients within each r_i, pooled across all r_i. Describe and justify the trend that you (expect to) observe.

(c) (10 points)

- (2 points)

Given the 2D chest-CT image, simulate a noisy (or noisier) version by introducing i.i.d. Gaussian zero-mean noise of standard deviation equaling 10% of the intensity range in the given image.

- (4 points)

You want to use the learned dictionary for $p = 0.8$ to denoise the simulated noisy image (not just a subset of its patches). Propose an optimization problem for the same, and implement a function to compute an optimal estimate of the denoised image.

- (2 points)

During the optimization process, show the graph of the objective function versus iterations.

- (2 points)

Show the simulated noisy image, the denoised image, and the original/given image.