HW5: Signal Separation and Image Recovery

ECE 5950 – Sparse Signal Processing (SSP)

- **Due date:** Tuesday, 11/11/14 at 11:59pm
- **Submission instructions:** Upload your solutions (including Matlab code) to Cornell Blackboard. Upload only a single file (e.g., one pdf including figures and source code, or a single archive file).
- **Total points:** 100pts

Problem 1: Audio signal separation (40pts)

The goal of this problem is to use ℓ_1-norm based signal separation to remove clicks and pops from an old phonograph recording, without knowing the locations of the clicks and pops.

Part 1 Load the audio file "mussorgsky.wav" in Matlab and play it through your computer's speakers or via headphones. Identify a 512-dimensional segment (or patch) of the audio file where a strong click or pop is clearly visible. Remember the location of this segment and plot the amplitudes over this segment.

Part 2 Take the segment $\mathbf{y}$ extracted in Part 1 and separate it into one part that is sparse in the discrete cosine transform (DCT) domain and one part that is sparse in the identity. In particular, solve the following (blind) signal-separation problem:

$$\text{(SEP)} \qquad \underset{\mathbf{x},\mathbf{e}}{\text{minimize}} \; \|\mathbf{x}\|_1 + \|\mathbf{e}\|_1 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{e}\|_2 \leq \varepsilon.$$

Here $\mathbf{A}$ is the *inverse* DCT matrix and $\mathbf{B}$ the identity matrix; the vector $\mathbf{x}$ is the sparse representation of the music signal and $\mathbf{e}$ the sparse representation of the clicks/pops. Try to find a good (small) parameter ε so that the clicks/pops are well separated from the audio signal. Compute an estimate of the audio signal without clicks/pops via $\hat{\mathbf{y}} = \mathbf{A}\hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ is the sparse representation obtained by solving the (SEP) problem. Plot the original corrupted segment $\mathbf{y}$ and the clean version $\hat{\mathbf{y}}$. Also plot the sparse representations $\mathbf{x}$ and $\mathbf{e}$. *Hint: Use SPGL1* `https://www.math.ucdavis.edu/~mpf/spgl1/` *to solve the (SEP) problem. The SPGL1 solver allows function handles and is way faster than CVX!*

Part 3 Now chop the entire audio signal into adjacent segments of 512 samples and perform (SEP) on each segment. Then resynthesize a clean version of the music signal as discussed in Part 2 for each block. Save the resulting audio file and also listen to it. *Hint: Boundary artifacts between adjacent segments will be clearly audible. If you perform recovery with overlapping blocks and proper resynthesizing (windowing and overlap and add), the result will sound much better!*

Problem 2: Compressive sensing and images (40pts)

There are many signal models that sparsify images. Examples of these include: (a) patches being sparse in a 2-dimensional DCT basis (the idea behind JPEG compression); (b) patches being sparse in some learned dictionary; and (c) the entire image sparse in the wavelet domain (the idea behind JPEG2000 compression).

These models can be used to recover an image from compressive sensing measurements. The goal is to recover an image from a random subset of all pixels using basis pursuit (BP). More specifically, given an $N \times N = 256 \times 256$ dimensional image, you have access to the intensity values at only M randomly selected pixels. Try to recover the original image under each of the following two sparsity models for images:

- *Image is sparse in the 2-dimensional DCT:* Assume that the entire image (if you would observe all its entries) is sparse in the 2D DCT transform domain
- *Image is sparse in a wavelet basis:* Assume that the entire image (if you would observe all its entries) is sparse in the 'db4' wavelet domain.

Part 1 Perform image recovery for three values of $M = 0.125N^2, 0.25N^2, 0.5N^2, 0.75N^2$ of the total number of pixels N^2 in the image. For each value of M, extract a random subset of pixels (chosen uniformly at random). Show the reconstructed images for each value of M and for both sparsifying bases; also save the resulting images. Compute the mean square error (MSE) and the reconstruction SNR of the recovered images.

Part 2 Which sparsifying basis works better for this task? Explain the phenomenon using some of the theoretical results learned in this class?

Hints:

- *Matlab has a powerful wavelet toolbox. Use the* `dwt` *and* `idwt` *commands. The Matlab help will explain you how to use them.*
- *Forget about using CVX for this task—use a fast solver for BP instead. I highly recommend SPGL1 mentioned in Problem 1.*
- *Use the test image provided with this assignment (boat.tiff). But you can also use any other grayscale test image for this task.*

Problem 3: Briefly describe your final project I (20pts)

Write the background and problem statement of your final project of this class. In particular, briefly describe what the main goals of the project are, then use math to describe the problem in a concise way. Finally, outline ways (as a bullet list) how you are going to solve the problem; of course, you do not know the final answer yet, but I want to see what methods you are going to try out next. Remember that someone who reads your description should be able to follow your ideas and potential solution approaches.

Hints:

- *I suggest the use of LaTeXto write your brief report. Note that you will have to hand in a final report anyway, so if you are doing it right now, it will save you work at the end of the class.*
- *Do not write too much. Just try to explain the key concepts that are needed to understand your problem and the ways you are going to solve it. In case you have no idea how to solve your problem, talk to the professor.*
- *For those who work in groups, both need to hand in the description (but it can be the same).*