ECE 532: Homework 1

Due on Tuesday, Sept. 9

Robert Nowak 11:00 am

Elijah Bernstein-Cooper

September 18, 2014

Problem 1

1a

Given $\boldsymbol{X} = [\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_n] \in \mathbb{R}^p$, we can express the matrix $\boldsymbol{C}$ where

$$\boldsymbol{C} = XX^T \tag{1}$$

as the following sum of rank-1 matrices

$$\boldsymbol{C} = \sum_{i=1}^{n} \frac{\boldsymbol{x}_i \boldsymbol{x}_i^T}{n} \tag{2}$$

1b

The rank of $\boldsymbol{C}$ will be n.

Problem 2

2a

To determine if $\Phi(\boldsymbol{x})$ is a norm where

$$\Phi(\boldsymbol{x}) = \sum_{j=1}^{m} \left(\sum_{i \in G_j} x_i^2 \right)^{1/2} \tag{3}$$

we first recognize that $\Phi(\boldsymbol{x})$ is simply a sum over an instance of the p-norm where $p = 2$ because $i \in G_j$ will include all elements in the sent $\{1, 2, \ldots, n\}$. The sum over the p-norm is also a 1-norm. The norm of a norm, is in fact a norm, thus $\Phi(\boldsymbol{x})$ is a norm.

2b

When $m = 1$, $\Phi(\boldsymbol{x})$ is the Euclidean norm. When $m = n$, $\Phi(\boldsymbol{x})$ is the 1-norm.

Problem 3

Given

$$\cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^T \boldsymbol{y}}{\|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2} \tag{4}$$

and that $|\cos(\boldsymbol{x}, \boldsymbol{y})| \leq 1$, the absolute value of the numerator cannot be larger than the denominator, thus $|\boldsymbol{x}^T \boldsymbol{y}| \leq \|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2$.

Problem 4

4a

Given $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ we can write $\boldsymbol{x}$ as

$$\boldsymbol{x} = \boldsymbol{A}^{-1} \boldsymbol{y} \tag{5}$$

4b

To bound the 2-norm of $\boldsymbol{x}$ with a function of $\boldsymbol{A}$ and $\boldsymbol{y}$ we first take $\|\boldsymbol{x}\| = \|\boldsymbol{A}^{-1}\boldsymbol{y}\|$ which can be expressed as

$$\frac{\|\boldsymbol{A}^{-1}\boldsymbol{y}\|}{\|\boldsymbol{y}\|} \|\boldsymbol{y}\| \tag{6}$$

where

$$\frac{\|\boldsymbol{A}^{-1}\boldsymbol{y}\|}{\|\boldsymbol{y}\|} \tag{7}$$

is the matrix norm. Eq. 7 will always be less than $\|\boldsymbol{A}^{-1}\|$, thus

$$\|\boldsymbol{x}\| \leq \|\boldsymbol{A}^{-1}\| \|\boldsymbol{y}\| \tag{8}$$

Problem 5

5a

The rank of $\boldsymbol{A}$ is 3.

5b

$\boldsymbol{x}$ can be expressed as

$$\boldsymbol{x} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \end{pmatrix} \boldsymbol{y} \tag{9}$$

Problem 6

6a

The rank of $\boldsymbol{X}$ is 3.

6b

The rank of $\frac{\boldsymbol{X}\boldsymbol{X}^T}{n}$ is 3.

6c

A set of linearly independent columns of $\boldsymbol{X}$ are

$$\begin{pmatrix}1\\0\\0\\0\end{pmatrix} \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

Problem 7

We have adopted a non-local method of denoising an image based on matching intensities throughout the image. Our algorithm cycles through each pixel in the noisey image and chooses 25% of the total pixels which are closest in intensity to the pixel in the cycle. See Figure 1 for an example of the local vs. non-local denoising algorithms. We can see that the non-local intensity averaging does not retain the original contrast levels in the image, but does successfully decrease the noise.

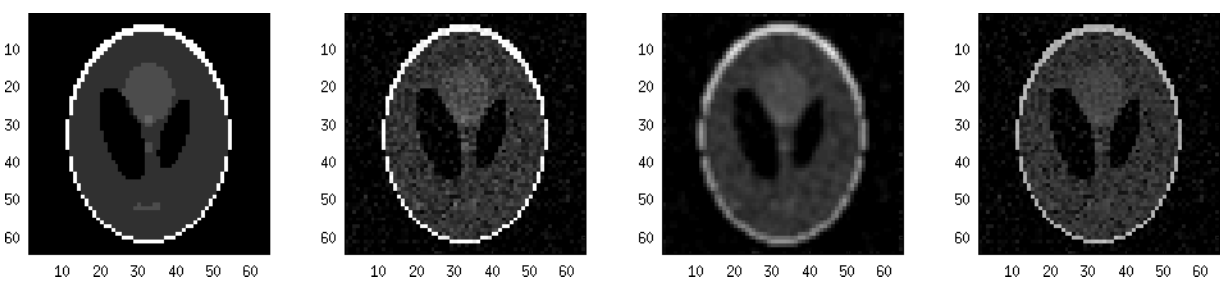

Figure 1: *Left:* Original image. *Left middle:* Original image with added Gaussian noise. *Right middle:* Denoised image with a local kernel filter. *Right:* Denoised image with intensity matching.

Problem 7 continued on next page...

Below is the code used to perform the intensity matching denoising.

```
clear
close all
n = 64;

% noise free image
original= double(phantom(n)) * 256;

% add noise
noisey = original+ randn(size(original)) * 15;

% denoise by distance-based averaging
w = [1/16 1/8 1/16;1/8 1/4 1/8; 1/16 1/8 1/16];

for i=1:n
    for j=1:n
        if (i==1)||(i==n)||(j==1)||(j==n)
            xavg(i,j) = noisey(i,j);
        else
            b = [noisey(i-1, j-1) noisey(i, j-1) noisey(i+1, j-1);
                 noisey(i-1, j) noisey(i, j) noisey(i+1, j);
                 noisey(i-1, j+1) noisey(i, j+1) noisey(i+1, j+1)];
            xavg(i, j) = sum(sum(b.*w));
        end
    end
end

% Denoise by intensity similarity
intensity_list = reshape(noisey, numel(noisey), 1);
[intensity_list, sort_indices] = sort(intensity_list);
denoised_image = zeros(numel(intensity_list), 1);

for i=1:numel(intensity_list)
    int_val = intensity_list(i);

    diff = intensity_list - int_val;
    [diff, diff_indices] = sort(abs(diff));
    int_val_near = mean(intensity_list(diff_indices(2:1000)));

    int_val_avg = (int_val + int_val_near) / 2.0;

    denoised_image(sort_indices(i)) = int_val_avg;
end

shape = size(noisey);
denoised_image = reshape(denoised_image, shape(1), shape(2));

fig = figure(1);clf;
subplot(141);imagesc(original, [0,256]);axis image; colormap gray
subplot(142);imagesc(noisey, [0,256]);axis image; colormap gray
subplot(143);imagesc(xavg, [0,256]);axis image; colormap gray
subplot(144);imagesc(denoised_image, [0,256]);axis image; colormap gray
linkaxes

saveas(fig, 'noisey_images', 'png')
```