ECE 6250, Fall 2019

Homework #5

Due Wednesday October 2, at the beginning of class

As stated in the syllabus, unauthorized use of previous semester course materials is strictly prohibited in this course.

1. Using your class notes, prepare a 1-2 paragraph summary of what we talked about in class in the last week. I do not want just a bulleted list of topics, I want you to use complete sentences and establish context (Why is what we have learned relevant? How does it connect with other things you have learned here or in other classes?). The more insight you give, the better.

2. Let $\psi(t)$ be the "hat" function centered at $t = 0$:

$$\psi(t) = \begin{cases} t+1, & -1 \leq t \leq 0 \\ 1-t, & 0 \leq t \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

   Sketch $\psi(t)$. Then sketch $\psi(at-b)$ for some $a, b > 0$. Label your plot carefully; critical points on the t axis should be labeled in terms of a, b.

3. Write a MATLAB function `gramschmidt.m` that takes a $N \times K$ matrix $\boldsymbol{A}$ with $N \geq K$ and returns an $N \times K$ matrix $\boldsymbol{Q}$ such that 1) the span of the columns of $\boldsymbol{Q}$ is the same as that for $\boldsymbol{A}$, and 2) the columns of $\boldsymbol{Q}$ are orthogonal and have unit ℓ_2 norm.

   Try your code out on the $N \times K = 1000 \times 50$ matrix `A` in `hw05problem3.mat`. Verify that the two conditions hold by checking that $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = \mathbf{I}$ ($\mathbf{I}$ is the $K \times K$ identity matrix) and that the column rank of the $N \times 2K$ matrix $\begin{bmatrix}\boldsymbol{A} & \boldsymbol{Q}\end{bmatrix}$ is equal to K. That is, show that running the command

```
>> rank([A Q])
```

   gives 50 back, and that

```
>> max(max(abs(eye(50)-Q'*Q)))
```

   is suitably small (less than 10^{-10}, say).

4. Write two MATLAB functions, called `mydct.m` and `myidct.m`, that implement the discrete cosine transform (DCT) and its inverse for a vector of length N. You code should be short (4 lines or less per function, no loops), efficient (it should make use of the `fft` command), and match the output of MATLAB's `dct` and `idct` commands. You can verify the latter with

```
x = randn(1000000,1);
d1 = mydct(x);
d2 = dct(x);
norm(d1-d2)

y = randn(1000000,1);
w1 = myidct(y);
w2 = idct(y);
norm(w1-w2)
```

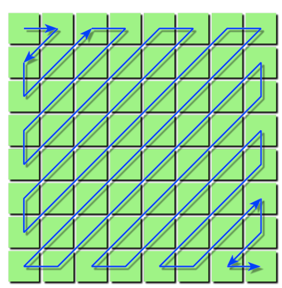
The norms of the differences should be small in both cases. (The one thing you really have to handle here is the fact that the DCT uses cosines that are shifted by half a sample.)

5. In this problem, we will explore the main concepts behind the JPEG compression standard. Start by looking over the Wikipedia page on JPEG:
`http://en.wikipedia.org/wiki/JPEG`
Turn in your code for all parts, plots and reconstructed images for a few values of M for part c, and your calculations and reconstructed image for part d.

   (a) Write a MATLAB function `block_dct2.m` which takes an $N \times N$ pixel image, divides it into 8×8 pixel blocks (you may assume that N is divisible by 8), and returns the discrete cosine transform coefficients for each block. You will find the MATLAB command `dct2.m` helpful.

   Write another MATLAB function `iblock_dct2.m` which is the inverse of the above: that takes the blocked DCT coefficients and returns the image.

   (b) The provided function `jpgzzind.m` orders the indexes of a block from low frequencies to high frequencies, as shown below:



   If `xb` is an 8×8 block, then `xb(jpgzzind(8,8))` is a 64×1 vector containing the same elements, just in the "correct" order.

   Using this utility, write a MATLAB function `block_dct2_approx.m` that takes an $N \times N$ image and a number M, and returns an $N \times N$ approximation $\tilde{\boldsymbol{x}}_M$ formed by keeping the first M DCT coefficients in each block.

   (c) Download the file `bb.tiff`. Read it into MATLAB using `x=double(imread('bb.tiff'))`.

For this image, plot[1]

$$\log_{10}\left(\frac{\|\boldsymbol{x}-\tilde{\boldsymbol{x}}_M\|_2^2}{\|\boldsymbol{x}\|_2^2}\right)$$

versus M (choose a range and number of values of M that make this plot meaningful). Using the `imagesc.m` command, show the approximation for $M = 1, 3, 8$.

(d) Using the quantization table in `jpeg_Qtable.mat`, quantize your transform coefficients using

$$\tilde{\alpha}_{k,\ell} = Q_{k,\ell} \cdot \text{round}\left(\frac{\alpha_{k,\ell}}{Q_{k,\ell}}\right).$$

Calculate how many of the resulting coefficients are non-zero, and compute

$$\log_{10}\left(\frac{\|\boldsymbol{x}-\tilde{\boldsymbol{x}}_q\|_2^2}{\|\boldsymbol{x}\|_2^2}\right)$$

where $\tilde{\boldsymbol{x}}_q$ is the image reconstructed from the quantized coefficients. Verify Parseval by checking that

$$\|\tilde{\boldsymbol{\alpha}}-\boldsymbol{\alpha}\|_2 = \|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|_2.$$

Using the `imagesc` command, show the reconstructed image.

[1] If x is an image, then `norm(x,'fro')` returns the standard sqrt-sum-of-squares of the entries norms. This is the Frobenius norm. If you don't specify the `'fro'`, MATLAB will return the operator norm (largest singular value) of x.