Assignment 3

Digital Signal Analysis & Applications

February 28, 2019

- **Deadline is 10th March 2019, 11:55 PM**
- All questions are compulsory. Follow the instructions carefully.
- All coding questions have to be done in MATLAB only.
- **Make a detailed report for all the questions**
- **Ensure that submitted assignment is your original work. Please do not copy any part from any source including your friends, seniors and/or the internet. If any such attempt is caught then serious action will be taken.**
- The submission - Roll number.zip - should contain the given files: *q1.m*, *q2.m*, *q4.m*, *q5.m*, *q6.m*, *q7.m*, *Report.pdf* as well as the auxiliary functions.
- Report should contain details of algorithm implementation, results, observations & answers to the subjective questions (if any).
- **You are expected to use vector operations in all your Matlab codes. Non vectorized codes will be penalized**

Problem 1.
Perform linear regression on the dataset given in the file *houses.csv* using methods taught in the class. The columns in the data file represent size of house (Square feet), no. of bedrooms & price of house respectively. For this question, split the dataset into two parts: 90% for training and remaining 10% to test your results. (To estimate the correctness of your solution, generate the L2 norm of predicted and ground truth house prices in the test set)

1. What are the coefficients corresponding to each feature of the house? Predict the price of a house with size 1400 square metres and 4 bedrooms
2. Does normalizing the feature values help in any ways? Compare the results that you get with & without normalizing the data
3. Compute the mean of all the features. Does this pass through the regression line / plane that you generated?

4. Can the same method be used to solve this problem if the no. of rows in the data is 1 million? Discuss the computational complexity of the current method.

Problem 2. Pick two images f, h of same dimension (256 X 256). Compute their respective Fourier transforms F, H.

1. Verify that iDFT[FH] does NOT correspond to the center portion of f * h (convolution). (For 256 X 256 images, iDFT[FH] would be 256 X 256 but f * h would be 511 X 511)
2. Compute the average of squared difference between pixel values in iDFT[FH] and the central 256 X 256 portion of f * h.
3. What changes do you observe when you zeropad the original images to dimension (511 X 511) and now calculate iDFT[FH] and report the new error.

Problem 3.
Choose any 64x64 image. Now, add 64 columns and rows of zeros to the right and bottom side of the original image. Repeat this process two more times each time doubling the image size and padding the pixels on the right and bottom by zeroes. You will therefore have 4 images first one 64 x 64 with no zero padding and then 128 x 128, 256 x 256 and 512 x 512 after padding. Find the Fourier transform of all these images. Display the results and explain and justify the relationship between the four outputs you get.

Problem 4.
To help visually impaired people dial numbers, standard two tone telephones play sinusoids at two different frequencies every time a key is pressed.

1. In this problem, you are given a file **Q4.mat** with signal X and the sampling frequency, Fs. The signal consists of the sound played by a particular key of a two tone telephone.
2. Your task is to find out the frequencies of this particular key, and clean the noisy signal as best as you can.

Please write a detailed report for this question, complete with the methods, choices (if any) and calculations you made for this task.

Problem 5.
A UDP Protocol is one in which data is sent in packets from a source to the client choosing the best (fastest) possible route. There is no guarantee that the arriving datagrams will be in the correct sequence. Hence, explicit reordering of chunks of data at the client side is required in the application layer. You are building an online suite for a streaming service. You want to build your application in such a manner, that the data that arrives at the client side, is reordered (before the information is presented to the user, in the application layer) and outputted to the user.

1. Design a function reorderDatagram that takes in the downloaded data chunks, and returns a single data piece (output).

2. Also, for logging purposes, the correct order needs to be displayed in STDOUT. The sound samples need to be denoised too. (Hint: The chunks that are created have an overlap of 3-5 seconds at the ends)

Problem 6.
The implementation of linear spatial filters requires moving a mask centered at each pixel of the image, computing sum of products of mask coefficients and corresponding pixels.

1. Implement an algorithm for low-pass filtering a grayscale image by moving a $k \quad k$ averaging filter of the form ones(k)/(kˆ2).

2. As the filter is moved from one spatial location to the next one, the filter window shares many common pixels in adjacent neighborhoods. Exploit this observation and implement a more efficient version of averaging filter. To appreciate the benefits of doing so, generate a plot of k vs run-time for various sized images. The plot diagram should contain a line plot for each image size you pick. Use different marker types to distinguish the default implementation and improved implementation. Just to give you a rough idea, look at the sample-plot .

Problem 7.
Perform linear regression on *Altitude.ods*. In q1, you would have used the closed form solution for regression. Now do not use the closed form solution, but arrive at the solution using gradient descent. There are several approaches to implementing gradient descent as well which would provide different performences as well. So explore different approaches and find the best fitting line. Make a detailed report on all your observations. For this question as well, split the dataset into two parts: 90% for training and remaining 10% to test your results. You have to implement it from scratch.