MATH 150-03 / COMP 150-07 Homework #3

November 12, 2016

1 Overview

The homework is due by 11:59 PM on Tuesday, November 15th. All code is to be uploaded to the Tufts cluster. Find your directory in /cluster/tufts/train/math150. Make a directory called submit. Inside the submit directory make a directory called hw3. Please put all submission files there by the due date. Please scan your responses to the non-programming parts (or type them up), and put in the folder with the name hw3_written.pdf. Please supply clear instructions on how to compile your codes and any necessary compile scripts and please include sample input files with which your code works. For some parts, you will be using the Intel MKL library and the Intel C compiler, which are available via the module intel/2013_sp1 on the cluster. Note: for the C code parts involving wavelets, if you have difficulty, limit yourself to one level of the WT transform. Please specify clearly in your submission that this is the case.

2 Assignment details

A. (3 pts) Suppose W is an orthogonal transform matrix. What is W^{-T} in terms of W? Prove that $\|Wx\|_2 = \|x\|_2$. Show that $W^T W = W^{-T} W^{-1}$.

B. (7 pts) Suppose the following signal is specified: $[13, 2, 13, 2, 0, 8, 5, 10, 6, 12, 2, 1]$. Compute the one level and two level Haar transforms of this signal. Then show how you would recover the original signal back from the transform. Please show your work.

C. (10 pts) Wavelet image denoising using Haar and CDF 9/7 wavelets using Matlab. Pick two images of your choice. Resize them (or crop) to be 600×600 pixels and include in your submission. Use ImageMagick tools to accomplish this. See for details: http://www.imagemagick.org/script/index.php. Use module load ImageMagick to use these programs. Using the provided example codes in, /cluster/tufts/train/math150/svoron01/Unit3/week3/image_denoising_examples, add Gaussian noise to the image with different signal to noise ratio values: $30, 20, 10$. Apply the (2d) transform to the image, threshold small coefficients, and apply the inverse transform to reconstruct the image. The parameters you use (e.g. thresholding level, number of levels of WT) are up to you. You can threshold using soft thresholding, as discussed in the class or your own custom scheme. Show the results you obtain (the original image, the noisy image, and the reconstructed image). Supply the Matlab code that you used. Finally, combine the parts which convert your images to 600×600, the code to make them noisy, and the code to denoise (the whole experiment) into one batch script which you can run with sbatch. See my sample codes in: /cluster/tufts/train/math150/svoron01/Unit3/week3/image_denoising_examples for an example of this. Supply the batch script and the output from running it.

D. (15 pts) Complete the pthreads mat-vec library we develop with a function for matrix-matrix multiplication. The code for matrix-vector multiplication with pthreads will be provided to you. Explain how your matrix-matrix routine works and write a driver program which measures the runtimes for 50 multiplications using matrices of size 500×500 up to 2500×2500 versus different numbers of threads used. For this, use the timing routines which we will go over in class. Supply a plot comparing the runtimes of your pthreads mat-mat mult code to the mat-mat mult routine of the OpenMP code we develop. You may choose the type of plot you wish to use to compare the timings. Please supply the code you used to get the timings. Notice, that the mat-vec openMP vs mat-vec p-threads timing program is implemented in: /cluster/tufts/train/-math150/svoron01/Unit3/week3/intermediate_mat_vec_lib3/main.c. You will also find the mat-mat mult code in that directory, as well as a function to initialize a matrix with random entries. Be careful with this last part, if you do this in parallel the regular rand() function is not thread-safe.

E. (25 pts) Write a program in C to implement the Haar wavelet transforms (forward and inverse). Make the code compile to binary *haarwt*. The program will read the following multi-line text file:
12, forward, 2
13, 2, 13, 2, 0, 8, 5, 10, 6, 12, 2, 1

The first line specifies the number of vector elements, the transform type (forward or inverse) and the number of levels (either 1 or 2). The second line lists the elements of the vector in one row, comma separated (one or more space in between). Have the program print out the result of the transform in suitable format to stdout. Use OpenMP pragmas to parallelize your code for multi-core systems. Please do not use Intel MKL or other libraries for this part. Notice that the Matlab code for the transforms is available in e.g.
/cluster/tufts/train/math150/svoron01/Unit3/week3/sparsity_constrained_regularization2/wavelets and in previous week folders.

F. (40 pts) Using the Intel MKL library for BLAS operations and OpenMP pragmas for other functions, make a program to run a (possibly Haar wavelet constrained) regularization scheme with the FISTA algorithm. That is, produce a continuation scheme for the optimization problem:

$$\bar{w} = \arg\min_{w} \left\{ \|A(W^{-1}w) - b\|_2^2 + 2\tau\|w\|_1 \right\} \quad ; \quad \bar{x} = W^{-1}w$$

over different values τ.
Your code will be similar to the Matlab code provided in:
/cluster/tufts/train/math150/svoron01/Unit3/week2/sparsity_constrained_regularization1. Except, do not explicitly form the matrices W and W^{-1} and use instead the routines you coded in part (E). We will discuss the details in class. The code should compile to an executable called *runfista*. Notice that as part of your code, you will need to implement a routine which estimates the spectral norm of your matrix $M = AW^{-1}$ and scales the matrix-vector operations accordingly (by one over this quantity). See the script run_sparse_basis_continuation1.m and the estimate spectral norm script, which implements the power iteration method. See also the directory
/cluster/tufts/train/math150/svoron01/Unit3/week3/sparsity_constrained_regularization2/ for a more explicit example of not forming M.

The program will take as input a multi-line text file with the following values:
0 or 1 (to use wavelets or not, int)
1 or 2 (number of Haar WT levels to use, int)
max tau (max regularization parameter, float)
min tau (min regularization parameter, float)
num taus (number of subdivisions, int)
wav_tol (wavelet threshold tolerance, float)
data/matA.bin (matrix location binary file, string)
data/matb.bin (vector rhs in binary file, string)
data/vecx_out.bin (vector output in binary file, string)

A sample file and inputs will be provided to you. Write the resulting solution vector to the specified output file, in the usual binary format (to be read in by readVecBinary matlab script from last unit). Test your program for some sample inputs (which you can generate yourself) based on the scripts I provide and provide some plots in your pdf: (L-curve, residual as function of parameter versus noise norm, original signal, recovered signal). You can simply use my Matlab script to do this, but you should test your C code to make sure it provides plausible results. You can also use the CDF WT transform from the Matlab code. In addition, describe in your pdf the optimization problem you are solving and how the spectral norm estimation algorithm works. Describe the steps your code takes to go from the matrix and right hand side to the final solution.