CS 515, Assignment # 5

Prof. Ron

Due March 9, 2005

Prepared by Youngmi Hur

Goals. In this assignment, you will learn how to manipulate the wavelet coefficients in order to capture the information you need. You will make use to this end `The IDR FrameNet Portal` . First go to `The IDR FrameNet Portal` and login using your username and password.

Problem 1. In this problem, we will study linear processing of wavelet coefficients. We will use 7 different algorithms (=templates) to this end. These algorithms are based on one single method, but tune, each, differently the parameters of that method.

For $j = 1, \cdots, 7$, our template j will set all the wavelet coefficients at scales $\leq j$ to be zero. (Note that our index j is opposite to the one used in class: here lower j corresponds to *higher* frequency). This process is LINEAR since it multiplies each coefficient by a fixed scalar (0 or 1 in our case), with the scalar depending on the location of the coefficient, but not on the actual value of the coefficient. The process depends on neither the 'System' nor the 'Data'.

Step A: Go to `1 Dimension`>`Data Options`>`Manage Collections`. Check the checkbox corresponding to `my_data`. Then the checkbox corresponding to `hur::An_internet_signal`. Then click `save selection` on the top row.

Step B: Go to `1 Dimension`>`Transform Options`>`Manage Collections`. Check the checkbox corresponding to `hur::FORlinear` and click `save selection`.

Step C: Go to `1 Dimension`>`Template Options`>`Manage Collections`. Check the checkbox corresponding to `hur::FORlinear` and click `save selection`.

Step D: (**View the signal**) Go to `1 Dimension`>`Analysis` and choose all the above collections as your 'Groups' for Step 1 of the Analysis page. Click the link saying `There is 1 signal in this group`. Inside Data Index, click `View`. This is the signal we will use for Problem 1.

Download this signal into your computer. (If you use Unix or Linux, to extract `tar.gz` file, use the unix command `tar -xvvzf`).

Using matlab, read the file `job-9.1` into a variable called `X`. Matlab command `X=dlmread('job-9.1', ' ')` will do the work. Plot `X` and check if it looks the same as the graph in DEVise 9 window. Turn in the graph of `X` with a proper title. Also turn in the matlab codes that you used to draw the graph.

Do not close this DEVise 9 window. We will need it later.

Step E: (**Experiment with RS2**) Go back to the Analysis page in the main window and click `Start Visualization`. You will see DEVise 1 window launching. The graphs you see in the bottom frame are the wavelet coefficients of the data with the system `RS2`. This is due to the fact that Template is chosen `framenet::default/identity` by default. On the top frame of DEVise window, change the Template to be `hur::FORlinear/2` and see what happens in the bottom frame of DEVise window. The template `hur::FORlinear/2` sets the wavelet coefficients from frequency level 1 and 2 to be zero. Try other templates and observe the changes on the graphs.

Now with the template `hur::FORlinear/7`, click `Reconstruct` on the first row of the top frame of DEVise 1 window. You will see DEVise 2 window launching. Compare this graph with the one in Step D.

Download the signal in DEVise 2 window to your computer.

Using matlab, read the file `job-2.1` into a variable called `reX_RS2`. Plot `reX_RS2` and the difference `X` $-$ `reX_RS2` with proper titles. Turn in both the graphs and the matlab codes that you used.

Step F: (**Experiment with PS4-1typeII**) Now on the top frame of DEVise 1 window, change the System to be `PS4-1typeII`. Do the same analysis as in Step E and download the reconstructed signal in DEVise 2 window into your computer.

Using matlab, read the file `job-2.1` into a variable called `reX_PS`. Plot `reX_PS` and the difference `X` $-$ `reX_PS` with proper titles. Turn in both the graphs and the matlab codes that you used.

Step G: (**Experiment on Batch Reconstruction**) Close all the DEVise windows. In the main window, go to `1 Dimension`>`Reconstruction`. In Step 2 of the Reconstruction page, change the Views to be `Transform` × `Templates` and then click `Start Reconstruction`. Now compare the graphs on the left column with the graphs on the right column. You can zoom in to see the details.

Two graphs on the first row are the same as the original signal since `RS2` and `PS4-1typeII` give 'Perfect Reconstruction' if we do not modify the wavelet coefficients (which corresponds to `framenet::default/identity` .)

Except that first row, on any fixed row, decide which reconstruction is visually 'better'. Explain why do you think that reconstruction is 'better'.

Now download the graphs into your computer. For $n = 2, 3, 4, 5, 6, 7, 8$, read the downloaded file `job-2.n` into a variable called `reX_j_RS2` with $j = n - 1$. For $n = 10, 11, 12, 13, 14, 15, 16$, read the downloaded file `job-2.n` into a variable called `reX_j_PS` with $j = n - 9$. For each $j = 1, 2, 3, 4, 5, 6, 7$, compute the difference `errX_j_RS2` = `X` − `reX_j_RS2` and the difference `errX_j_PS` = `X` − `reX_j_PS`. For each j, compute ‖`errX_j_RS2`‖/‖`X`‖ and ‖`errX_j_PS`‖/‖`X`‖. Figure out which one is smaller, for each j. Compare the results here with the observations you made visually in the above. Also save all the signals `errX_j_RS2` and `errX_j_PS` as column vectors. For example, for `errX_1_RS2`, use the matlab command `save errX_1_RS2.txt errX_1_RS2 -ascii -double` to save. You will need these files in the next step. Turn in the results you have obtained as well as the matlab codes you have used.

Step H: (**Upload Data**) Go to `1 Dimension>Data Options>Upload Data`. There, upload each `errX_j_RS2` and `errX_j_PS`, for $j = 1, 2, 3, 4, 5, 6, 7$ using the form. In the Install Data page, select `my_data` as your collection name and then click `Save to Collection`. After uploading all 10 files, go to `1 Dimension>Data Options>Share Data`. Add the username `hur` to share your collection `my_data`. Close all DEVise windows.

Problem 2. In this problem, we will experiment an example of nonlinear processing of wavelet coefficients. We will use the so-called 'Hard thresholding' algorithm. The **Hard thresholding** η_λ **at threshold level** $\lambda > 0$ is defined to be

$$\eta_\lambda(t) := \begin{cases} t, & \text{if } |t| > \lambda, \\ 0, & \text{if } |t| \le \lambda. \end{cases}$$

We apply this Hard thresholding operator on the wavelet coefficients. Of course, different λ will give different result. To select a 'near-best' λ for each different case, we need to look at the individual wavelet coefficient and that makes this process NONLINEAR.

Step A: Go to `1 Dimension>Data Options>Manage Collections`. Check the checkbox corresponding to `hur::A_noisy_signal` and then click `save selection`.

Step B: Go to `1 Dimension>Transform Options>Manage Collections`. Check the checkbox corresponding to `hur::FORnonlinear` and click `save selection`.

Step C: Go to `1 Dimension>Template Options>Manage Collections`. Check the checkbox corresponding to `hur::FORnonlinear` and click `save selection`.

Step D: (**View the signal**) Go to `1 Dimension>Analysis` and choose all the above collections as your 'Groups' for Step 1 of the Analysis page. Click the link saying `There is 1 signal in this group`. Inside Data Index, click `View`. This is the signal we will use for Problem 2.

Download this signal into your computer.

Using matlab, read the file `job-9.1` into a variable called `Y`. Plot `Y` and check if it looks the same as the graph in DEVise 9 window. Turn in the graph of `Y` with a proper title. Also turn in the matlab codes that you used to draw the graph.

Do not close this DEVise 9 window. We will need it later.

Step E: (**Experiment with RS4**) Go back to the Analysis page and click `Start Visualization`. You will see DEVise 1 window launching. The graphs you see in the bottom frame are the wavelet coefficients of the data with the system `RS4`. Now on the top frame of DEVise window, change the Scale of Y Axis to be `Local` and then click `Submit display options`. Look at the graphs carefully by paying attention to the size of them. On the top frame, change the Template to be `hur::FORnonlinear/Hard_Thresh/For A_noisy_signal/For RS4` and see what happens in the bottom frame of DEVise window. The template `hur::FORnonlinear/Hard_Thresh/For A_noisy_signal/For RS4` apply Hard thresholding with some number λ that I have chosen for this specific data and system. Click `Reconstruct` on the first row of the top frame of DEVise 1 window. You will see DEVise 2 window launching. Compare this graph with the one in Step D.

Download the signal in DEVise 2 window into your computer.

Using matlab, read the file `job-2.1` into a variable called `reY_RS4`. Plot `reY_RS4` and the difference `Y − reY_RS4` with proper titles. Also compute $\|$`Y − reY_RS4`$\|/\|$`Y`$\|$. Turn in the graphs, the results and the matlab codes that you used.

Step F: (**Experiment with DAUB8**) On the top frame of the DEVise 1 window, change the System to be `DAUB8`. Observe that the λ I have chosen for `RS4` is not doing a good job for these wavelet coefficients anymore. Now to see the untouched wavelet coefficients, change the Template to be `framenet::default/identity` . After that, change the template to be `FORnonlinear/Hard_Thresh/For A_noisy_signal/For DAUB8`. The template `FORnonlinear/Hard_Thresh/For A_noisy_signal/For DAUB8` apply Hard thresholding with some number (other) λ that I have chosen for this specific data and system. Now click `Reconstruct`. Compare the new graph in DEVise 2 window with the one in Step D.

Download the signal in DEVise 2 window into your computer.

Using matlab, read the file `job-2.1` into a variable called `reY_DAUB`. Plot `reY_DAUB` and the difference `Y − reY_DAUB` with proper titles.

Also compute $\|$`Y − reY_DAUB`$\|/\|$`Y`$\|$. Turn in the graphs, the results and the matlab codes that you used.

Step G: (**Experiment with** λ) Choose the System to be `RS4` and choose the Template to be `FORnonlinear/Hard_Thresh/With_lambda`. This time, you get to choose the Thresholding level λ. Click the radio button corresponding `Enter a number` and then enter a number to Threshold, then click `Submit parameters`. Click `Reconstruct` to see the reconstructed signal after the Thresholding. Try different λ until you find a good one. When you find a good λ, write down your choice λ and download the reconstructed signal into your computer. Using matlab, read the file `job-2.1` into a variable called `reY_RS4StepG`. Compute the ratio $\|$`Y − reY_RS4StepG`$\|/\|$`Y`$\|$ and compare this with the ratio in Step E.