Carnegie Mellon University
15-826 – Multimedia Databases and Data Mining
Spring 2009, C. Faloutsos
Assignment 3, Due Date: **April 21**, in class

Reminders

- Due date: April 21, 3:00pm, hard copy in class and soft copy e-mailed to the TA (*bziebart+826 at cs*) in a single e-mail (along with your source code).
- Please turn in a **typed** report.
- All homeworks including this one are to be done INDVIDUALLY.
- Expected effort for this homework (approximate times):
  - Q1: 2 hours
    - 1 hour to locate and install necessary algorithms
    - 1 hour to run algorithms and write answers for the questions
  - Q2: 3 hours
    - 1 hour to download and run the existing script
    - 1 hour to write and debug your algorithm
    - 1 hour to run your algorithm and answer questions
  - Q3: 3 hours
    - 1 hour to derive the equation
    - 1 hour to write and debug your algorithm
    - 1 hour to run your algorithm and answer questions
  - Q4: 3 hours
    - 2 hours to locate and install necessary algorithms
    - 1 hour run algorithms and answer questions
  - Q5: 9 hours
    - 5 hours to install software and download data
    - 2 hours to write algorithms
    - 2 hour to run algorithms and answer questions

Q1 – Singular Value Decomposition [20pts]

Problem Description: We will now investigate the Singular Value Decomposition of a cloud of points.

Consider the 5-d dataset available at (`http://www.cs.cmu.edu/~bziebart/15826-S09/hw3/svd.dat`). Each point, $\mathbf{p}_i$, in the point cloud lies roughly on a 2-d plan spanned by two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ with

additional noise term ϵ_i that has zero mean and a diagonal co-variance matrix Σ with same value along the diagonal. More formally, $p_i = x_i * \mathbf{v}_1 + y_i * \mathbf{v}_2] + \epsilon_i$, where $\epsilon_i \sim N(0, \Sigma)$ and $\epsilon = \sigma I_{5,5}$. $I_{5,5}$ is the identity matrix.

1. Use SVD to recover $\mathbf{v}_1$ and $\mathbf{v}_2$. Submit the values of these vectors.
2. Plot and submit the pre-image points (x_i, y_i).
3. Estimate and report the value of Σ (i.e., find σ) using the second and third principle components of the point cloud.

Q2 – Discrete Wavelet Transform [20pts]

Problem Description: We will now investigate wavelet decompositions for a number of 1-d datasets.

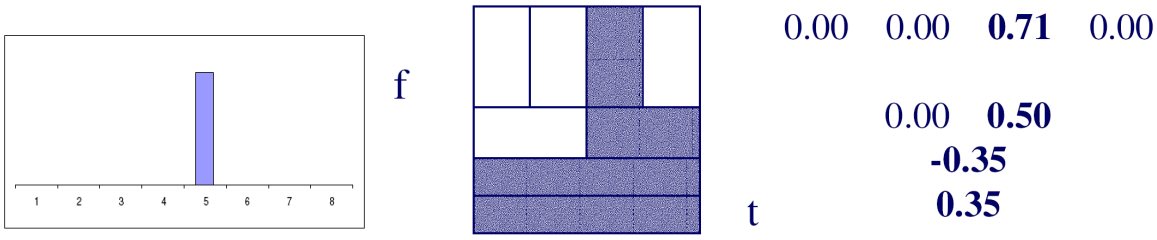

Figure 1: A spike signal (left), the wavelet decomposition coefficient magnitude plot (center), and the wavelet decomposition coefficient numerical plot (right).

Consider the spike shown in Figure 1 (left). The plot of wavelet decomposition coefficients for this signal is shown in Figure 1 (center) and the values in Figure 1 (right). Your submitted wavelet decomposition coefficient plots should be similar in presentation to Figure 1 (center).

1. Implement the inverse DWT for Haar Wavelets. Your algorithm should accept as input the wavelet decomposition coefficients (see Figure 1) and output the recovered signal[1]. Please submit your code.
2. For each of each of these 1-d datasets (`http://www.cs.cmu.edu/~bziebart/15826-S09/hw3/dwt1.dat`, `http://www.cs.cmu.edu/~bziebart/15826-S09/hw3/dwt2.dat`, and `http://www.cs.cmu.edu/~bziebart/15826-S09/hw3/dwt3.dat`) perform the following:
   - Using the wavelet script available at (`http://www.cs.cmu.edu/~bziebart/15826-S09/hw3/wavelet.pl`, obtain and plot the wavelet decomposition plot for all the levels of coefficients.[2]

[1] Assume lengths that are powers of 2.
[2] For example, `./wavelet.pl < dwt1.dat > output.dat`.

- Set half the coefficients with the smallest magnitudes to zero. Using your inverse transform code, obtain a reconstruction of the original dataset from the wavelet decomposition coefficients. Plot the difference between the original dataset and this reconstruction.

3. For the mystery wavelet coefficients (`http://www.cs.cmu.edu/~bziebart/15826-S09/hw3/idwt.dat`), use your inverse wavelet transform algorithm to recover the original 1-d signal and plot that signal.

Q3 – Discrete Fourier Transform with Missing Values[20pts]

We will now use the Discrete Fourier Transform to analyze the spectral frequency of different data and recover missing values.

Recall that the discrete Fourier transform (DFT) is:

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \text{ for } k = 0, ..., N-1. \tag{1}$$

The inverse DFT is:

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} \text{ for } n = 0, ..., N-1. \tag{2}$$

1. Using I_n as an indicator of missing values (0 when x_n is missing and 1 otherwise), we can modify the discrete Fourier transform to deal with missing data:

$$X(k) = \frac{1}{\sqrt{\sum_{n=0}^{N-1} I_n}} \sum_{n=0}^{N-1} I_n x_n e^{-\frac{2\pi i}{N} kn} \text{ for } k = 0, ..., N-1. \tag{3}$$

   Modify the inverse DFT (Equation 2) to deal with datasets that have missing values for some x_n. Write and submit the new formula.

2. Consider the 1-d data stream (`http://www.cs.cmu.edu/~bziebart/15826-S09/hw3/dft-miss.dat`) and data mask (`http://www.cs.cmu.edu/~bziebart/15826-S09/hw3/dft-mask.dat`). For every value of 0 in the data mask, the corresponding value in the data stream is to be considered missing and the value in the data stream should be ignored. Employ your algorithm on this dataset and report the first 10 coefficients.

3. Keep the coefficients recovered by your algorithm and use them to recover the data stream for the entire sequence (including your recovered missing values). Plot and submit this entire sequence.

Q4 – Tensors [20pts]

We will now perform tensor analysis.
(`http://www.cs.cmu.edu/~bziebart/15826-S09/hw3/tensor.dat`)

Consider a 10 x 10 x 5 communication tensor of (person1, person2, word). $X_{i,j,k} = 1$ indicates that sender i messaged receiver j using word k.

The provided dataset (`http://www.cs.cmu.edu/~bziebart/15826-S09/hw3/tensor.dat`) is a list of i, j, k indices that have value 1 in the tensor.

1. Employ PARAFAC to uncover the two major groups (i.e., two factors).
2. List the senders (i values), receivers (j values), and messages (k values) for each group.

Q5 – Hadoop [20pts]

We will now use Hadoop to analyze the Netflix movie rating graph. Note that while the main benefit of Hadoop is the ability to run on multiple machines, it can also be run on a single machine, which we will do to learn how to use it. **Warning:** The Netflix dataset is very large (2.6GB uncompressed) and a large amount of additional software is needed for Hadoop. Please plan accordingly. For Windows machines, Wubi (`http://wubi-installer.org`) is a simple way to install Ubuntu on a Windows machine without any hard drive partitioning, or Cygwin can be employed.

Preliminaries:

- Download and install Hadoop from (`http://hadoop.apache.org/core`).
- Download and install PIG from (`http://hadoop.apache.org/pig`).
  You will also need to install: Perl, SVN, and Ant
- Download the Netflix dataset
  (`http://www.db.cs.cmu.edu/db-site/Datasets/graphData/NETFLIX/`)
  - Each file corresponds to a movie.
  - Entries are "MovieID,UserID,Rating,Date" – we will only need the MovieID and UserID.

1. Using PIG Latin (the language of PIG) and Hadoop, write an algorithm to recover a distribution for the number of ratings per movie. Note that the rating values themselves are not needed for this. Submit this code.
2. Using PIG Latin and Hadoop, write an algorithm to recover the distribution of the number of ratings per user. Again, note that the rating values themselves are not needed. Submit this code.
3. Run your first algorithm and submit a plot of the log-log distribution of number of ratings per movie (i.e., log number of movies with N ratings vs. log N).
4. Run your second algorithm and submit a plot of the log-log distribution of number of ratings per user (i.e., log number of users with N ratings vs. log N).