Homework 1

Due on September 20 (11:59 p.m)

Problem 1: Linear Algebra

Let's warm up with a simple algebra problem.

1a. Rotational Matrices

I) A rotation in 4-D (Whose coordinates we will call X, Y, Z and W) is characterized by three angles. We will characterize them as a rotation along the $X - Y$ plane, a rotation along the $Y - Z$ plane and rotation along the $Z - W$ plane. Derive the rotation matrix R that transforms a vector $[x, y, z, w]^T$ to new vector $[\hat{x}, \hat{y}, \hat{z}, \widehat{w}]^T$ by rotating it counter-clockwise by an angle θ along the $X - Y$ plane, an angle δ along the $Y - Z$ plane and an angle ϕ along the $Z - W$ plane.

II) Confirm that $RR^T = I$.

1b. Lengths of Vectors

I) The following matrix transforms 5-dimensional vectors into 4-dimensional ones:

$$A = \begin{bmatrix} 1 & 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 & 7 \\ 2 & 1 & 5 & 6 & 11 \\ 4 & 7 & 9 & 8 & 15 \end{bmatrix}$$

1. A 5×1 vector v of length 1 is transformed by A as $u = Av$. What is the longest length that u can be?
2. What is the shortest non-zero length of u?
3. What is the rank of A? The null space of A is the space of all vectors v such that $Av = 0$.
4. What is the dimensionality of the null space of A?

II) Answer the above four questions for the following matrix. Here B transform 4-D vectors to 5-D. Note that $B = A^T$.

$$B = \begin{bmatrix} 1 & 2 & 2 & 4 \\ 1 & 3 & 1 & 7 \\ 2 & 4 & 5 & 9 \\ 3 & 5 & 6 & 8 \\ 4 & 7 & 11 & 15 \end{bmatrix}$$

III) The "Restricted Isometry Property" (RIP) constant of a matrix characterizes the change in length of vectors transformed by sub-matrices of the matrix. For our matrix A, let A_s be a matrix formed of any s columns of A. If A is $M \times N$, A_s will be $M \times s$. We can form A_s in ${}^{N}C_s$ ways from the N columns of A (we assume that the order of vectors in A_s is immaterial). Let w be an $s \times 1$ vector of length 1. Let l_{max} be the longest vector that one can obtain by transforming w by any A_s. Let l_{min} be the shortest vector obtained by transforming w by any A_s. The RIP-s constant δ_s of the matrix A is defined as:

$$\delta_s = \frac{l_{max} - l_{min}}{l_{max} + l_{min}}$$

1. What is δ_2 (i.e., δ_s for $s = 2$) for the matrix A given above? Hint you must consider all ${}^{5}C_2$ possibilities for A_s .
2. What is δ_3 ?

Problem 2: Projections

Please note all the files needed for this exercise can be found in the homework template.

The recording segment Polyushka.wav, played on harmonica, has been downloaded from YouTube with permission from the Artist.

Note.tar is a tar file, which contains a set of notes from a harmonica. You are required to transcribe the music. For transcription you must determine how each of the notes is played to compose the music.

"matlab_instructions" file inside the homework package contains information how to convert each note into a spectral vector, and the entire music to a spectrogram matrix.

2a. Analysis By Individual Note:

I) For each note individually, compute the contribution of that note to the entire music. Mathematically, if N_i is the vector representing the i^{th} note, and M the music matrix, find the row vector W_i such that $N_i W_i \approx M$. Return the transcription of each note.

II) Recompose the music by "playing" each note according to the transcription you just found. Mathematically, compute $\widehat{M} = \sum_i N_i W_i$ and invert the result to produce music. To invert it to a music signal, follow the instructions given in the MATLAB notes. Return the recomposed music. Comment about how the recomposed music compares to the original signal.

2b. Joint Analysis Using All Notes:

I) Compute the contribution of all notes to the entire music jointly. Mathematically, if $N = [N_1, N_2, \dots]$ is the notes matrix where the individual columns are the notes, find the matrix W such that $NW \approx M$. The i^{th} row of W is the transcription of the i^{th} note.

II) Recompose the music by "playing" each note according to the transcription you just found in problem 2b-I. Mathematically, compute $\widehat{M} = NW$, and invert the result to produce the music. Return the recomposed music. Comment about how the recomposed music compares to the original signal. Is the recomposed music identical to the music you constructed in problem 2a-II? Explain your finding.

Problem 3: Optimization and Non-Negative Decomposition

Simple projection of music magnitude spectrograms (which are non-negative) onto a set of notes will result in *negative* weights for some notes. To explain, let M be the (magnitude) spectrogram of the music. It is a matrix of size $D \times T$, where D is the size of the Fourier transform and T is the number of spectral vectors in the signal. Let N be a matrix of notes. Each column of N is the magnitude spectral vector for one note. N has size $D \times K$, where K is the number of notes.

Conventional projection of M onto the notes N computes the approximation

$$\widehat{M} = NW$$

Such that $\left\|M - \widehat{M}\right\|_F^2 = \sum_{i,j}\left(M_{i,j} - \widehat{M}_{i,j}\right)^2$ is minimized. Here $\left\|M - \widehat{M}\right\|_F$ is known as the Frobenius norm of $M - \widehat{M}$. $M_{i,j}$ is the $(i,j)^{th}$ entry of M, and $\widehat{M}_{i,j}$ is similarly the $(i,j)^{th}$ entry of $\widehat{M}$. Please note the definition of the Frobenius norm; we will use it later.

$\widehat{M}$ is the projection of M onto N. W, of course, is given by $W = pinv(N)M$. W can be viewed as the transcription of M in terms of the notes in N. So, the j^{th} column of M, which we represent as M_j and is the spectrum in the j^{th} frame of the music, is approximated by the notes in N as

$$M_j = \sum_i N_i W_{i,j}$$

where N_i is the i^{th} column of N and represents the i^{th} note, and $W_{i,j}$ is the weight assigned to the i^{th} note in composing the j^{th} frame of the music.

The problem is that in this computation, we will frequently find $W_{i,j}$ values to be *negative*. In other words, this model requires you to subtract some notes- since $W_{i,j}N_i$ will have negative entries if $W_{i,j}$ is negative, this is equivalent to subtracting the weighted note $\left|W_{i,j}\right|N_i$ in the j^{th} frame. Clearly, this is an unreasonable operation intuitively; when we actually play music, we never unplay a note (which is what playing a negative note would be).

Also, $\widehat{M}$ may have negative entries. In other words, our projection of M onto the notes in N can result in *negative* spectral magnitudes in some frequencies at certain times. Again, this is meaningless physically- spectral magnitudes cannot, by definition, be negative.

In this homework problem we will try to fix this anomaly.

We will do this by computing the approximation $\widehat{M} = NW$ with the constraint that the entries of must W always be greater than or equal to 0, i.e., they must be non-negative. To do so we will use a simple gradient descent algorithm, which minimizes the error $\|M - NW\|_F^2$ subject to the constraint that all entries in W are non-negative.

3a. Computing a derivative

We define the following error function

$$E = \frac{1}{DT} \|M - NW\|_F^2$$

where D is the number of dimensions (rows) in M, and T is the number of vectors (frames) in M. Derive the formula for $\frac{\partial E}{\partial W}$.

3b. A Non-Negative Projection

We define the following gradient descent rule to estimate W. It is an iterative estimate. Let W^0 be the initial estimate of W and W^n the estimate after n iterations. We use the following projected gradient update rule

$$\widehat{W}^{n+1} = W^n - \eta \frac{\partial E}{\partial W} |W^n$$
$$W^{n+1} = \max\left(\widehat{W}^{n+1}, 0\right)$$

where $\frac{\partial E}{\partial W} |W^n$ is the derivative of E with respect to W computed at $W = W^n$, and $\max\left(\widehat{W}^{n+1}, 0\right)$ is a *component-wise* flooring operation that sets all negative entries in $\widehat{W}^{n+1}$ to 0.

In effect, our *feasible set* for values of W are $W \geq 0$, where the symbol '$\geq$' indicates that *every* element of W must be greater than or equal to 0. The algorithm performs a conventional gradient descent update, and projects any solutions that fall outside the feasible set back onto the feasible set through the *max* operation.

Implement the above algorithm. Initialize W to a matrix of all 1s. Run the algorithm for η values $(0.0001, 0.001, 0.01, 0.1)$. Run 250 iterations in each case. Plot E as a function of iteration number n. Return this plot and the final matrix W. Also show a plot of best error E as a function of η.

3c. Recreating the music (No points for this one)

For the best η (which resulted in the lowest error) recreate the music using this transcription as $\widehat{M} = NW$. Resynthesize the music from $\widehat{M}$. What does it sound like? You may return the resynthesized music to impress us (although we won't score you on it).

How To Submit

You have to submit everything through canvas. Inside "Hw1_template.zip", the "data" folder contains the notes and music required for Problem 2 and 3. The folders named Problem1, 2, and 3 contain necessary instructions (sometimes for both Python and MATLAB) and MATLAB templates. If you use MATLAB, use these templates. If you use Python, use the Jupyter notebook template for all problems in google collab in the following link-

https://colab.research.google.com/drive/1N28A0DrqwOO1A8wO7xfFWmB6NPit592Q?usp=sharing

Instructions for MATLAB:

You will submit a zip file containing all the required files specified below. Title your zip file as "HW1_yourJHID.zip", where 'yourJHID' is your JHED ID.

<u>For Problem 1:</u> Write your answers for problem 1 in the doc file titled "Problem_1_answers" inside the folder "Problem1".

Deliverables: Problem_1_answers.docx

<u>For Problem 2:</u> From the Problem 2 folder, follow the ".m" MATLAB template files and the additional instructions in "matlab_instructions" file.

Deliverables:

problem2a.dat, problem2a_synthesis.wav, problem2b.dat, problem2b_synthesis.wav

<u>For Problem 3:</u> For 3a, write the answer in "Problem_3a_answer.docx" inside the Problem 3 folder. You can write the answer in the doc file, or you can upload picture of handwritten answer titled "**Problem_3a_answer**". For 3b and 3c, follow the ".m" MATLAB template files in that folder.

Deliverables:

Problem_3a_answer.docx/.jpg/.png, problem3b_eta_xxx.dat, problem3b_eta_xxx_errorplot.png, problem3b_eta_vs_E.png, polyushka_syn.wav (optional)

Instructions for Python:

You will have to use Jupyter Notebook for Python. We provided the template in Google Colab. Instructions are given in the link. You have to submit the specific deliverables mentioned below. Put them inside "results" folder inside the designated problem folder. You have to submit the "ipynb" file and "Hw1_yourJHID.zip" zip file containing the required results or reports. In the text comments on Canvas, put the link to your Google colab notebook with proper permissions.

For Problem 1: Write your answers for problem 1 in the answer cells under "Problem 1: Linear Algebra". Or you can write your answers for problem 1 in the doc file titled "Problem_1_answers" inside the folder "Problem1" in "Hw1_template.zip".

Deliverables: Problem_1_answers.docx (if you choose doc)

For Problem 2: The templates are given in the cells under "Problem 2: Projections". Edit the empty functions and cells. Here all functions and cells require editing except for function "stft". More instructions are given in "notebook_instructions.pdf" inside folder "Problem 2" in "Hw1_template.zip".

Deliverables:

problem2a.dat, problem2a_synthesis.wav, problem2b.dat, problem2b_synthesis.wav

For Problem 3: The templates are given in the cells under "Problem 3: optimization and non-negative decomposition". For 3a, you can write the answer in Notebook or you can upload picture of handwritten answer titled "**Problem_3a_answer**".

Deliverables:

Problem_3a_answer.jpg/.png, problem3b_eta_xxx.dat, problem3b_eta_xxx_errorplot.png, problem3b_eta_vs_E.png, polyushka_syn.wav (optional)