EECS 16B Designing Information Devices and Systems II
Spring 2016 Anant Sahai and Michel Maharbiz

Midterm 1

Exam location: The Faery Land Of Unicorns and Rainbows

PRINT your student ID: ____

PRINT AND SIGN your name: ________________, ________________ ____________________
(last) (first) (signature)

PRINT your Unix account login: ee16b-_____________

PRINT your discussion section and GSI (the one you attend): ____

Name of the person to your left: ____

Name of the person to your right: ____

Name of the person in front of you: ____

Name of the person behind you: ____

Section 0: Pre-exam questions (3 points)

1. **What other courses are you taking this term? (1 pt)**

2. **Think back to remember a moment where you triumphed. Describe it. How did you feel? (2 pts)**

Do not turn this page until the proctor tells you to do so. You can work on Section 0 above before time starts.

PRINT your name and student ID: ____

[Extra page. If you want the work on this page to be graded, make sure you tell us on the problem's main page.]

Section 1: Straightforward questions (36 points)

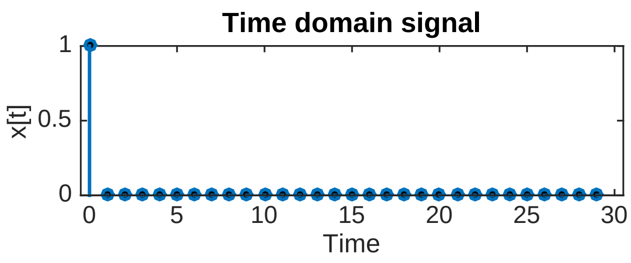
Unless told otherwise, you must show work to get credit. There will be very little partial credit given in this section. **You get one drop: do 4 out of the following 5 questions. (We will grade all and keep the best 4 scores.)** *Each problem is worth 9 points. No additional bonus for getting them all right.*

3. DFT Matching

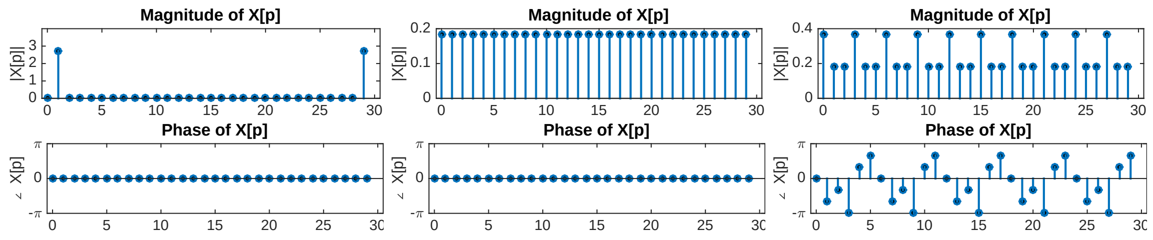
You have to match the n-long time domain signal $\vec{x}$ with its frequency domain $\vec{X}$ coordinates. The DFT coefficients (coordinates in the DFT basis) are represented with their magnitudes $|X[p]|$ and phases $\angle X[p]$, i.e., $X[p] = |X[p]|e^{i\angle X[p]}$ as p goes from 0 (constant) up through $n-1$. Throughout this problem $n = 30$. (For your information, $\frac{1}{\sqrt{30}} \approx 0.18$ and $\sqrt{30} \approx 5.5$.)

Circle your answer. There is no need to give any justification.

(a) Given the time domain signal below,



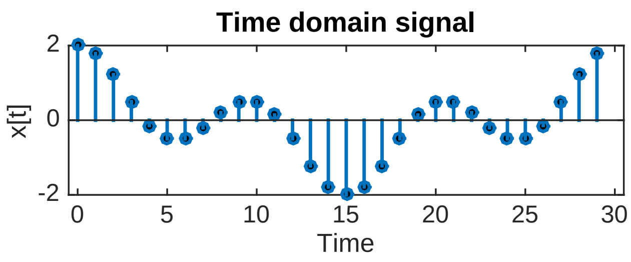
which one is the corresponding DFT coefficients?



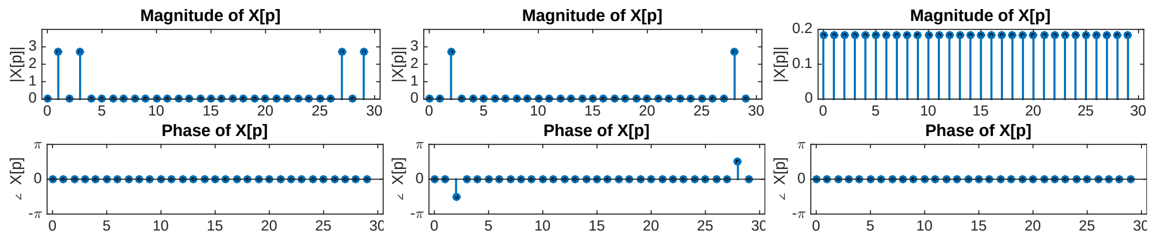
(A) (B) (C)

Solutions: By definition, $X[p] = \sum_{t=0}^{n-1} x[t]u_p[t]^* = u_p[0]^* = \frac{1}{\sqrt{n}}$, so B is the right answer since it is clearly a constant magnitude with zero phase. The zero-phase is because $\frac{1}{\sqrt{n}}$ is real and positive.

(b) Given the time domain signal below,



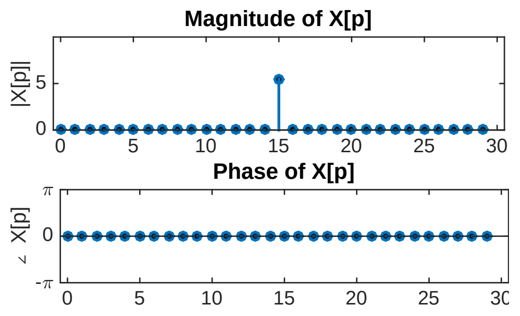
which one is the corresponding DFT coefficients?



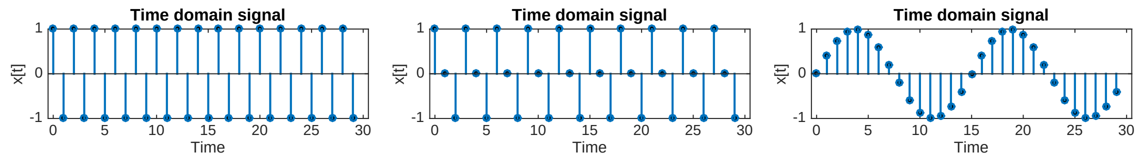
(A) (B) (C)

Solutions: The signal is clearly an addition of two cosines, so A is the right answer. B is just a single phase-shifted cosine since there are only two complex frequency components that respect conjugate symmetry. C is just the answer to the previous part and so can't be the answer to this one.

(c) Given the **DFT domain representation** below,



which one is the corresponding **time domain** signal?



(A) (B) (C)

Solutions: By definition, $x[t] = \sum_{p=0}^{n-1} X[p]u_p[t] = \sqrt{n}u_{15}[t] = e^{i\frac{2\pi}{30}15t} = e^{i\pi t} = (-1)^t$, so A is the right answer.

Many students were tempted by choice (C) because they thought that this has a single complex frequency in it and so should look sinusoidal. But when we look at (C) we see that the frequency is clearly 2 — there are two cycles in the block. 2 is not 15. In (A), we see that there are 15 cycles present.

4. U square

Consider the 4×4 matrix U that consists of the DFT basis for four-dimensional space. $(n = 4)$

$$U = \begin{bmatrix} \vec{u}_0 & \vec{u}_1 & \vec{u}_2 & \vec{u}_3 \end{bmatrix}$$

(a) **Compute both $\vec{u}_1^{\,T}\vec{u}_2$ and $\vec{u}_1^{\,T}\vec{u}_3$.**

Solutions:

For $\vec{u}_1^{\,T}\vec{u}_2$, there are three ways of solving it.

Arguably, the easiest and most elegant is to observe that $\vec{u}_1^T = \vec{u}_{4-3}^T = \vec{u}_{-3}^T = \vec{u}_3^*$. Then we have

$$\vec{u}_1^{\,T}\vec{u}_2 = \vec{u}_3^*\vec{u}_2 = 0$$

by the orthonormality of the DFT basis and the definition of the complex inner product.

Second, we could follow the derivation of orthonormality itself and evaluate the sum using the geometric series formula:

$$\vec{u}_1^{\,T}\vec{u}_2 = \frac{1}{4}\sum_{t=0}^{3} e^{i\frac{2\pi}{4}t}e^{i\frac{2\pi}{4}2t} \tag{1}$$

$$= \frac{1}{4}\sum_{t=0}^{3} e^{i\frac{2\pi}{4}3t} \tag{2}$$

$$= \frac{1}{4}\frac{e^{(i\frac{2\pi}{4}3)4} - 1}{e^{i\frac{2\pi}{4}3} - 1} \tag{3}$$

$$= 0 \tag{4}$$

Third, there is the simple way of just writing out the vectors doing the arithmetic directly:

$$\vec{u}_1^{\,T}\vec{u}_2 = \frac{1}{2}[1, i, -1, -i]\frac{1}{2}\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \tag{5}$$

$$= \frac{1}{4}(1 - i - 1 + i) \tag{6}$$

$$= 0 \tag{7}$$

Similarly, for $\vec{u}_1^{\,T}\vec{u}_3$, there are three ways of solving it.

First and most elegantly, observe that $\vec{u}_1^T = \vec{u}_{4-3}^T = \vec{u}_{-3}^T = \vec{u}_3^*$. Then we have

$$\vec{u}_1^{\,T}\vec{u}_3 = \vec{u}_3^*\vec{u}_3 = 1$$

by the orthonormality of the DFT basis and the nature of complex inner products.

Second using geometric sums:

$$\begin{aligned}
\vec{u_1}^T\vec{u_3} &= \frac{1}{4}\sum_{t=0}^{3} e^{i\frac{2\pi}{4}t}e^{i\frac{2\pi}{4}3t} && (8)\\
&= \frac{1}{4}\sum_{t=0}^{3} e^{i\frac{2\pi}{4}4t} && (9)\\
&= \frac{1}{4}(4) && (10)\\
&= 1 && (11)
\end{aligned}$$

And third, the simple way of just writing out the vectors doing the arithmetic directly:

$$\begin{aligned}
\vec{u_1}^T\vec{u_3} &= \frac{1}{2}[1,i,-1,-i]\frac{1}{2}\begin{bmatrix}1\\-i\\-1\\i\end{bmatrix} && (12)\\
&= \frac{1}{4}(1+1+1+1) && (13)\\
&= 1 && (14)
\end{aligned}$$

(b) **Compute $U^2 = UU$.**

(HINT: Think about the previous part.)

Solutions:

Seeing the pattern above, we can notice that

$$\vec{u_p}^T\vec{u_q} = \begin{cases}1, & \text{if } p+q \equiv 0 \pmod 4.\\ 0, & \text{otherwise}\end{cases} \qquad (15)$$

Then we have

$$U^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{16}$$

This can also be seen by explicit calculation

$$U^2 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} \tag{17}$$

$$= \frac{1}{4}\begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 0 & 4 & 0 \\ 0 & 4 & 0 & 0 \end{bmatrix} \tag{18}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{19}$$

For the exam, the above is fine. But in the real world, the above fact is pretty important.

Notice that U^2 is almost the identity. In fact $U^2 \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} a \\ d \\ c \\ b \end{bmatrix}$ is just a time-reversal of the initial vector except holding the zero-th position constant.

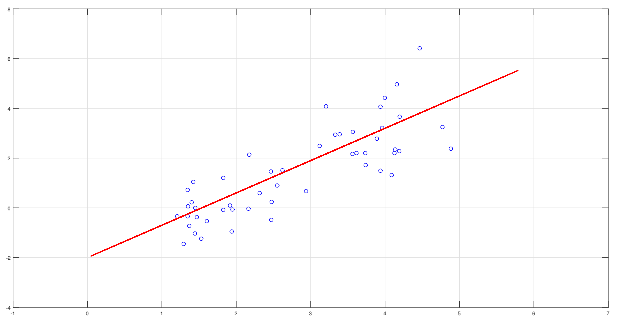
This means that if we can find a fast way to multiply by one of U^* or U, we essentially also have a fast way of multiplying by the other as well. The same applies for a circuit implementation. Whatever clever implication we use for the forward DFT can be tweaked to give the inverse as well.

PRINT your name and student ID: ____

5. PCA Recipe

Let $\vec{x}_i \in \mathbb{R}^d$, $i = 0, \ldots, n-1$ be n data points (column vectors) each having dimension d. Suppose we want to analyze the low dimensional (say k dimensional where $k < d$) structure of this data set with PCA.

The goal is to have just k-coordinates for each data point that represent where they are relative to each other on the best k-dimensional structure that fits all the data points. The following figure provides an illustration with $k = 1$ and $d = 2$.



In the figure, the line is the essential 1 dimensional structure and the dots are the n data points. For each data point, we would want to know the one coordinate describing how far along the line that data point's closest representative was.

Your task is to arrange *in order* an appropriate subset of the steps below to make that happen. Note that not all steps listed will be used.

Just write out a string (e.g. LEBIGMA) as your answer.

A. Assemble the vectors $\{\vec{y}_i\}$ into the columns of the matrix Y.

B. Run SVD on the matrix Y to get $Y = U\Sigma V^T$.

C. Copy the data points into vectors $\vec{y}_i = \vec{x}_i$.

D. Subtract the average from each dimension of the data: $y_i[j] = x_i[j] - \frac{1}{n}\sum_{\ell=0}^{n-1} x_\ell[j]$.

E. Subtract the average from each data point: $y_i[j] = x_i[j] - \frac{1}{d}\sum_{t=0}^{d-1} x_i[t]$.

F. Multiply the matrix Y by V'^T on the right. i.e. Compute YV'^T to get the desired coordinates.

G. Multiply the matrix Y by V'^T on the left. i.e. Compute $V'^T Y$ to get the desired coordinates.

H. Multiply the matrix Y by U'^T on the right. i.e. Compute YU'^T to get the desired coordinates.

I. Multiply the matrix Y by U'^T on the left. i.e. Compute $U'^T Y$ to get the desired coordinates.

J. Select the first k columns from the matrix V to get V'

K. Select the last k columns from the matrix V to get V'

L. Select the first k columns from the matrix U to get U'

M. Select the last k columns from the matrix U to get U'

Solutions: DABLI is the desired answer. Some students added C to the beginning and that was harmless. The important thing to notice was that if the data points are columns, that we need to remove the average of each row from each entry to get rid of the means and recenter the data. Then, having recentered, the first columns of U tell the directions on which the data points should be projected to get the coordinates of their closest representative in the best k-dimensional structure approximating the data.

PRINT your name and student ID: ____

6. DFT Meets k-means

There are n discrete-time signals, $\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_{n-1}$, where the length of each signal is d. We want to use the k-means algorithm to classify them into k clusters, where $k < n$.

Suppose that we run k-means on the time-domain signals $\vec{x}_0, \ldots, \vec{x}_{n-1}$ to find k clusters using the first k signals to initialize the cluster centers.

Alternatively, suppose we run k-means on the same vectors except in frequency domain (i.e. we take the DFT of each signal and then run k-means on that collection of DFT-ed signals) using the frequency-domain representation of the first k signals to initialize the cluster centers.

Is the resulting classification of the n signals by running k-means in the frequency domain going to be the same or different than the classification that results from running k-means in the time domain? Briefly justify your answer.

Solutions: The results are the same.

The DFT basis is an orthonormal basis and hence changing coordinates to it preserves distances. Moreover, the matrix U^* transforming to DFT coordinates is a linear operator and so, by linearity it commutes with taking, averages of vectors. In other words, the DFT of the average of a set of vectors is also the average of the DFTs of those vectors. The operations of the k-means algorithm only consider distances and averages of data points and so the same clusters will be found if we start from the same starting points.

To expand out this reasoning: In the beginning, the two k-meams runs choose the same set of cluster centers. Then, each signal (data point) is clustered into the closest center. Because DFT basis is orthonormal, the distances certainly are preserved and so the classifications are preserved. The initial groupings are the same.

Then the k-means algorithm computes the average of each cluster. Because change of basis is a linear transformation, the DFT coefficients of an averaged signal are the same as the average of DFT coefficients of the original signals.

After computing the centroids (DFT coefficients of the average signals), each data point is clustered to the nearest centroid. The distance between a data signal to each centroid is preserved by the DFT basis. Hence the grouping result is the same as that of the original signals. The same steps repeat until no cluster get updated.

Therefore, considering the DFT coefficients of signals, the whole k-means flow is exactly the same as that on the original signals.

7. SVD

Let $A = \begin{bmatrix} 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 3\sqrt{2} & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{2} & \frac{1}{2} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}^{T}$.

Find $\vec{x}$ such that

$$A\vec{x} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

and $||\vec{x}||$ is as small as possible.

Solutions:

This question is asking you to find and apply the Moore-Penrose Pseudoinverse as discussed in Homework 4 Question 3. Recall from the homework that it is defined as,

$$A^{\dagger} = V\widetilde{\Sigma}U^{T}$$

where,

$$\widetilde{\Sigma} = \begin{bmatrix} \frac{1}{3\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

First, note that,

$$\widetilde{\Sigma}U^{T} = \frac{1}{2} \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ -1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Then,

$$A^{\dagger} = \frac{1}{2}V \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ -1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & -1 & \sqrt{2} & 0 \\ 1 & -1 & -\sqrt{2} & 0 \\ 1 & 1 & 0 & \sqrt{2} \\ 1 & 1 & 0 & -\sqrt{2} \end{bmatrix} \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ -1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 2 & -1 \\ 2 & -1 \\ -1 & 2 \\ -1 & 2 \end{bmatrix}$$

Finally,

$$\vec{x} = A^{\dagger} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 \\ 2 \\ -1 \\ -1 \end{bmatrix}$$

If you got this far in the exam, you should get full credit. From now on, this solution is to aid in understanding. To see why this is the minimum norm solution, observe that,

$$\vec{x} = \left(\frac{1}{3}\right)\vec{v}_0 - (1)\vec{v}_1 + (0)\vec{v}_2 + (0)\vec{v}_3$$

That is to say, $\vec{x}$ is in the span of $\vec{v}_0$ and $\vec{v}_1$. Since there are only two non-zero singular values, we can conclude that $\vec{v}_2$ and $\vec{v}_3$ lie in the nullspace of A. Since V forms a basis and $\vec{v}_2$ and $\vec{v}_3$ are in the null space, $A\vec{x} = \vec{y}$ has many solutions in the form of,

$$\vec{z} = \frac{1}{3}\vec{v}_0 - \vec{v}_1 + \alpha\vec{v}_2 + \beta\vec{v}_3$$

Since V is an orthonormal transformation, it preserves norms. Note that,

$$||\vec{x}||_2^2 = 1 + \frac{1}{9} \text{ and } ||\vec{z}||_2^2 = 1 + \frac{1}{9} + \alpha^2 + \beta^2$$

Since the norm in minimized when $\alpha = \beta = 0$, $\vec{x}$ is the minimum norm solution.

PRINT your name and student ID: ____

[Extra page. If you want the work on this page to be graded, make sure you tell us on the problem's main page.]

PRINT your name and student ID: ______________________________

Section 2: Free-form Problems (72 + 10 points)

8. ROC K-Means (15 pts)

This question is about running k-means to group people into two rock bands based on musical taste. We start with two people as "band centers" to initialize the algorithm. People regroup into bands based on which band has an average musical taste closer to them.

This process of determining the musical taste for each band and re-choosing the groups is called re-grouping. As in k-means, re-grouping will continue until nobody wants to leave their band and join the other one. At that point, we will have bands that are ready to actually play gigs.

We decide to monitor the grouping process as k-means evolves. To do this, we define the state to be the way people are partitioned into two groups. It turns out that with this grouping process, it is impossible to have an empty band. Each band will always have at least one person in it.

In this problem, we consider four people, Ambika, Bao, Carol and Diego, the number of possible states is 7. We label them in the table below.

State	Combination
S_0	$\{[A],[B,C,D]\}$
S_1	$\{[B],[A,C,D]\}$
S_2	$\{[C],[A,B,D]\}$
S_3	$\{[D],[A,B,C]\}$
S_4	$\{[A,B],[C,D]\}$
S_5	$\{[A,C],[B,D]\}$
S_6	$\{[A,D],[B,C]\}$

Notice that $\{[A],[B,C,D]\}$ and $\{[D,C,B],[A]\}$ are the same state because a band is defined by its members.

For simplicity, suppose that musical taste is unidimensional — all that matters is how much you like cowbells. The cowbell scores are listed below:

Name	Ambika	Bao	Carol	Diego
Score	90	85	70	40

(a) (5 pts) **Based on the cowbell scores above, can some initial choice of two people lead to k-means starting in the state S_0? How about the state S_1? How about the state S_3?** Just tell us which of these are possible starting states and which are impossible.

Solutions:

The state S_1 is impossible. There are no initial band centers resulting in the grouping $\{[B],[A,C,D]\}$. Intuitively, it is impossible to group A,C,D together, but put B in another group, because A and B are closest to each other. So, the only way we can start with a cluster of just B is to make B an initial center. At that point, unless A was a band center on her own, she is going to join the B cluster. But if A is also a band center along with B, both C and D would join B's band and not A's.

Meanwhile S_0 and S_3 are possible. If we choose Ambika and Bao as the initial band centers, the grouping is S_0. If we choose Ambika and Diego as the initial band centers, the grouping is S_3.

(b) (5 pts) To track the evolution of the band-formation process, we decide to use a *state transition matrix* T. In this matrix, if $T_{ij} = 1$, it means the group state will change from S_j to S_i if one step of re-grouping happens. In other words, if we compute $T\vec{e}_j$, where $\vec{e}_j$ is the standard j-th basis vector, we will get $\vec{e}_i$. Consider a hypothetical three-state example:

$$T = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

In this example, S_0 will become S_1, S_1 will become S_2, and S_2 will stay forever.

Please fill in the missing entries in the transition matrix below for the Ambika, Bao, Carol and Diego example according to the cowbell scores given previously and the seven states as ordered earlier.

$$T = \begin{bmatrix} 0 & 0 & 0 & \square & 0 & 0 & 0 \\ 0 & 0 & 0 & \square & 0 & 0 & 0 \\ 0 & 0 & 0 & \square & 0 & 0 & 0 \\ 0 & 0 & 0 & \square & 0 & 0 & 0 \\ 1 & 1 & 1 & \square & 1 & 1 & 1 \\ 0 & 0 & 0 & \square & 0 & 0 & 0 \\ 0 & 0 & 0 & \square & 0 & 0 & 0 \end{bmatrix}$$

Solutions: The missing column indicates the state transition from S_3. From S_3, the two centroids are $\frac{(90+85+70)}{3} = 81.67$ for $[A,B,C]$, and 40 for $[D]$. D will clearly stay in its solo band. Noticing that C is 30 away from D but less than 12 away from the other centroid, C will also stay in the big band. So no one will move out from the current band, and the next state is S_3.

Notice that there should be only one non-zero entry in each column (there is only one possible next state for each state).

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

(c) (5 pts) **According to the matrix you just created, what are all the possible final states for the bands?**

Solutions: S_3 and S_4. The two states are local optima. For a state transition matrix, 1s on the diagonal indicate local optima, because these represents states that come back to themselves because no further improvements can be made.

9. Echos revisited (27 pts)

Recall that in the real-world, the radio waves emitted by a wireless transmitter don't only go directly to the wireless receiver that wants to listen to them. Instead, they propagate physically, much the same way that sound propagates. If there is a clear line-of-sight path between the transmitter and the receiver, then certainly the radio waves do reach the destination directly. But they also find other paths, bouncing off of obstacles like walls, the ground, the ceiling, buildings, hills, etc. Because the speed of light is finite, all of these echoes of the transmitted signal reach the destination with different delays.

In this problem, the wireless channel has a direct path with gain 1 and single echo t_d later with gain 0.8.

As we have in the 16 series so far, we will work in finite time. The total length of the signal is denoted by n since as in the locationing labs, the transmissions are assumed to be periodic with period n. So the time-delay $t_d \in \{0, 1, 2, \dots, n-1\}$ and the wireless channel's impulse response $\vec{h}$ has $h[0] = 1$ and $h[t_d] = 0.8$ with all other $h[t] = 0$.

Let the signal $\vec{x}$ be some signal sent by the transmitter, and the signal $\vec{y}$ be the one observed at the receiver. The two signals are connected based on the equation $\vec{y} = C_{\vec{h}}\vec{x}$, where $C_{\vec{h}}$ is the circulant matrix that has $\vec{h}$ as its first column.

Use $n = 5$ for this entire problem.

(a) (6 pts) Let the frequency-domain representations (coordinates in DFT basis) of $\vec{x}$, $\vec{y}$, $\vec{h}$ be $\vec{X}$, $\vec{Y}$, $\vec{H}$, respectively.

Express $\vec{Y}$ in terms of $\vec{H}$ and $\vec{X}$?

Solutions:

$$\begin{aligned}
\vec{Y} &= U^*\vec{y} && (20)\\
&= U^*C_{\vec{h}}\vec{x} && (21)\\
&= U^*C_{\vec{h}}(UU^*)\vec{x} && (22)\\
&= (U^*C_{\vec{h}}U)(U^*\vec{x}) && (23)\\
&= \text{diag}(\sqrt{5}\vec{H})\vec{X} && (24)\\
&= \sqrt{5}(\vec{H} \odot \vec{X}) && (25)
\end{aligned}$$

(b) (5 pts) Let the DFT coefficients of $\vec{h}$ be $\vec{H}$. **Write down the p-th element of $\vec{H}$ in terms of t_d, p, and $n = 5$.**

Solutions: Let $\vec{e}_k$ be the standard k-th basis vector.

$$\vec{H} = U^*\vec{h} \tag{26}$$
$$= U^*(\vec{e}_0 + 0.8\vec{e}_{t_d}) \tag{27}$$
$$= (\vec{u}_0{}^*)^T + 0.8(\vec{u}_{t_d}{}^*)^T \tag{28}$$

So, for the p-th element of $\vec{H}$, we have

$$H[p] = \frac{1}{\sqrt{5}}(1 + 0.8e^{-i\frac{2\pi}{5}t_d p}) \tag{29}$$

(c) (5 pts) To figure out what the impulse response ($\vec{h}$) is, we decide to send a cosine signal $\vec{x}$ at the transmitter and observe the signal $\vec{y}$ at the receiver. Let the t-th element of $\vec{x}$, $x[t]$, be $\cos\left(\frac{2\pi}{5}t\right)$. That is,

$$\vec{x} = \begin{bmatrix}\cos\left(\frac{2\pi}{5}(0)\right) & \cos\left(\frac{2\pi}{5}(1)\right) & \cos\left(\frac{2\pi}{5}(2)\right) & \cos\left(\frac{2\pi}{5}(3)\right) & \cos\left(\frac{2\pi}{5}(4)\right)\end{bmatrix}^T. \tag{30}$$

What is the frequency domain representation (coordinates in the DFT basis) of $\vec{x}$?

Solutions:

$$\vec{X} = U^*\vec{x} \tag{31}$$
$$= U^*\frac{\sqrt{5}}{2}(\vec{u}_1 + \vec{u}_4) \tag{32}$$
$$= \frac{\sqrt{5}}{2}(\vec{e}_1 + \vec{e}_4) \tag{33}$$
$$= \begin{bmatrix} 0 \\ \frac{\sqrt{5}}{2} \\ 0 \\ 0 \\ \frac{\sqrt{5}}{2} \end{bmatrix} \tag{34}$$

(d) (5 pts) After sending the above signal $\vec{x}$, we observe a signal $\vec{y}$ at the receiver. The DFT coefficients of $\vec{y}$ are given by

$$\vec{Y} = \frac{\sqrt{5}}{2}\begin{bmatrix} 0 \\ 1+0.8e^{-i\frac{4\pi}{5}} \\ 0 \\ 0 \\ 1+0.8e^{i\frac{4\pi}{5}} \end{bmatrix} \tag{35}$$

What is the echo delay, t_d, of $\vec{h}$?

Solutions:

$$\begin{aligned} Y[1] &= \sqrt{5}(H[1]X[1]) && (36)\\ &= \sqrt{5}\frac{1}{\sqrt{5}}(1+0.8e^{-i\frac{2\pi}{5}t_d})\frac{\sqrt{5}}{2} && (37)\\ &= \frac{\sqrt{5}}{2}(1+0.8e^{-i\frac{4\pi}{5}}) && (38)\end{aligned}$$

Therefore,

$$t_d = 2 \tag{39}$$

(e) (6 pts) **Please write the t-th element of the time-domain $\vec{y}$ from above in the form of $y[t] = \alpha\cos\left(\frac{2\pi}{5}t+\theta\right)$.**

What is θ? What is α? (You are permitted to use the Angle function to convert a complex number into polar coordinates and you don't have to simplify your answers.)

Solutions:

Let $\gamma = 1+0.8e^{-i\frac{4\pi}{5}}$.

$$\vec{y} = U\vec{Y} \tag{40}$$

$$= U\frac{\sqrt{5}}{2}(\gamma\vec{e}_1 + \bar{\gamma}\vec{e}_4) \tag{41}$$

$$= \frac{\sqrt{5}}{2}(\gamma\vec{u}_1 + \bar{\gamma}\vec{u}_4) \tag{42}$$

Then consider the t-th element of $\vec{y}$.

$$y[t] = \frac{1}{2}(\gamma e^{i\frac{2\pi}{5}t} + \bar{\gamma}e^{-i\frac{2\pi}{5}t}) \tag{43}$$

$$= \frac{1}{2}(|\gamma|e^{i(\frac{2\pi}{5}t+\angle\gamma)} + |\gamma|e^{-i(\frac{2\pi}{5}t+\angle\gamma)}) \tag{44}$$

$$= |\gamma|\cos\left(\frac{2\pi}{5}t + \angle\gamma\right) \tag{45}$$

Therefore, $\alpha = |1 + 0.8e^{-i\frac{4\pi}{5}}|$ and $\theta = \angle(1 + 0.8e^{-i\frac{4\pi}{5}})$.

PRINT your name and student ID: ____

[Extra page. If you want the work on this page to be graded, make sure you tell us on the problem's main page.]

10. Reverse-engineering (30 + 10pts)

One day, a small black box catches your eye. You find a label on the box that reads "Acme SportsRank Proprietary Repeated Multiplier" and learn online that the black box contains some **square** $n \times n$ **symmetric matrix** A with strictly positive eigenvalues $\lambda_0 > \lambda_1 > \ldots > \lambda_{n-1} > 0$. You can't look at the entries in the matrix, but you can compute $A^k\vec{x}$ for any $\vec{x}$ that you choose and any $k > 20$. You know nothing else about A due to its proprietary nature. In this problem, you will find the eigenvectors of A so that you can reverse-engineer the matrix.

(a) (15 pts) Our first task is to find the eigenvector $\vec{u}_0$ corresponding with the largest eigenvalue λ_0. To find this eigenvector we start with some randomly generated vector $\vec{x}$ and use the box to repeatedly multiply $\vec{x}$ by the matrix A. You can assume that anytime you randomly chose a vector $\vec{x}$, it is guaranteed to be able to be written as $\vec{x} = \sum_{p=0}^{n-1} \beta_p \vec{u}_p$ where $\vec{u}_p$ is the normalized p-th eigenvector of A with all the $|\beta_p| > 0$ for all p.
Show that

$$\lim_{k\to\infty} \frac{A^k\vec{x}}{||A^k\vec{x}||} = \vec{u}_0.$$

(HINT: Use the fact that $\left(\frac{x}{y}\right)^k \to 0$ *when* $x < y$ *and* k *is large.)*

Solutions: We use the notation provided in the problem description to write the vector $\vec{x}$ as a sum of A's eigenvectors:

$$\vec{x} = \sum_{i=0}^{n-1} \beta_i \vec{u}_i \tag{46}$$

We know that this is true because A is symmetric, and therefore it has a full complement of eigenvectors. Using this representation, we can write $A\vec{x}$ as:

$$A\vec{x} = \sum_{i=0}^{n-1} A\beta_i\vec{u}_i = \sum_{i=0}^{n-1} \beta_i\lambda_i\vec{u}_i \tag{47}$$

which uses the fact that $A\beta_i\vec{u}_i = \beta_i A\vec{u}_i = \beta_i\lambda_i\vec{u}_i$
Thus, when we multiply by A k times, we get

$$A^k\vec{x} = \sum_{i=0}^{n-1} \beta_i\lambda_i^k\vec{u}_i \tag{48}$$

From this expression, we can factor out $\beta_0\lambda_0^k$ from each term in the sum to get

$$A^k\vec{x} = \beta_0\lambda_0^k \left(\vec{u}_0 + \sum_{i=1}^{n-1} \frac{\beta_i}{\beta_0}\left(\frac{\lambda_i}{\lambda_0}\right)^k \vec{u}_i\right) \tag{49}$$

At this point, it is just a few steps of algebra:

$$\lim_{k\to\infty} \frac{A^k\vec{x}}{\|A^k\vec{x}\|} = \lim_{k\to\infty} \frac{\beta_0\lambda_0^k\left(\vec{u}_0 + \sum_{i=1}^{n-1}\frac{\beta_i}{\beta_0}\left(\frac{\lambda_i}{\lambda_0}\right)^k\vec{u}_i\right)}{\|\beta_0\lambda_0^k\left(\vec{u}_0 + \sum_{i=1}^{n-1}\frac{\beta_i}{\beta_0}\left(\frac{\lambda_i}{\lambda_0}\right)^k\vec{u}_i\right)\|} \tag{50}$$

$$= \lim_{k\to\infty} \frac{\text{sign}(\beta_0)(\vec{u}_0 + \sum_{i=1}^{n-1}\frac{\beta_i}{\beta_0}\left(\frac{\lambda_i}{\lambda_0}\right)^k\vec{u}_i)}{\|\vec{u}_0 + \sum_{i=1}^{n-1}\frac{\beta_i}{\beta_0}\left(\frac{\lambda_i}{\lambda_0}\right)^k\vec{u}_i\|} \tag{51}$$

Notice that the sign of the β_0 stays because the denominator effectively has absolute values in it. At this point, both the top and the bottom have well-defined limits and we can continue

$$\lim_{k\to\infty} \frac{A^k\vec{x}}{\|A^k\vec{x}\|} = \lim_{k\to\infty} \frac{\text{sign}(\beta_0)(\vec{u}_0 + \sum_{i=1}^{n-1}\frac{\beta_i}{\beta_0}\left(\frac{\lambda_i}{\lambda_0}\right)^k\vec{u}_i)}{\|\vec{u}_0 + \sum_{i=1}^{n-1}\frac{\beta_i}{\beta_0}\left(\frac{\lambda_i}{\lambda_0}\right)^k\vec{u}_i\|} \tag{52}$$

$$= \frac{\lim_{k\to\infty}\text{sign}(\beta_0)(\vec{u}_0 + \sum_{i=1}^{n-1}\frac{\beta_i}{\beta_0}\left(\frac{\lambda_i}{\lambda_0}\right)^k\vec{u}_i)}{\lim_{k\to\infty}\|\vec{u}_0 + \sum_{i=1}^{n-1}\frac{\beta_i}{\beta_0}\left(\frac{\lambda_i}{\lambda_0}\right)^k\vec{u}_i\|} \tag{53}$$

$$= \frac{\text{sign}(\beta_0)\vec{u}_0}{\|\vec{u}_0\|} \tag{54}$$

$$= \text{sign}(\beta_0)\vec{u}_0 \tag{55}$$

This is effectively the dominant eigenvector we started with, except with possibly a sign change. This sign change makes no difference since the first eigenspace has been successfully recovered.

This method for finding the largest eigenvector of a matrix is called the power iteration method. It is commonly used when the matrix A is too large to store on a computer, but the product $A\vec{x}$ is fast to compute because A has some known structure such as having 0 in most of its entries. For example, Google uses the power iteration method to actually compute their PageRank scores since the entire matrix representing the web graph is enormous.

(b) (15 pts) The argument above tells us that we can essentially compute a normalized eigenvector as $\vec{u}_0 = \frac{A^k\vec{x}}{||A^k\vec{x}||}$ by choosing a sufficiently large k. Assume that this works perfectly (neglect the effect of k being finite). Now that we have found $\vec{u}_0$, we want to find the eigenvector corresponding with the second largest eigenvalue λ_1. **Use what you know about the eigenvectors for a symmetric matrix as well as the method from the previous part to describe how we could find $\vec{u}_1$ using only black-box access to multiplication by A.** (You cannot change the matrix A — all you can do is pick vectors to multiply by it k times. You are allowed to get a new random vector if you'd like, as well as do any operations on it that you want involving $\vec{u}_0$.) Argue briefly why your method should work.

Solutions: When we write a vector as a weighted sum of the eigenvectors of A (i.e. $\vec{x} = \sum_{i=0}^{n-1} = \beta_i\vec{u}_i$), we know that the terms β_i can be computed using $\vec{u}_i^T\vec{x}$ because of the orthonormality of the $\vec{u}_i$ vectors. Using this and the fact that we have found the first eigenvector $\vec{u}_0$, we can start with a new random vector $\vec{y}$.

We know that $\vec{y} = \sum_{i=0}^{n-1}\tilde{\beta}_i\vec{u}_i$ with $|\tilde{\beta}_i| > 0$ by the assertion made earlier about randomly chosen vectors. We then project out the component of $\vec{u}_0$ in $\vec{y}$ by

$$\vec{y}' = \vec{y} - \text{proj}_{\vec{u}_0}\vec{x} \tag{56}$$
$$= \vec{y} - (\vec{u}_0^T\vec{y})\vec{u} \tag{57}$$

This ensures that when we write $\vec{y}'$ as a sum of the eigenvectors of A, the coefficient $\tilde{\beta}_0' = 0$ because $\vec{u}_0^T\vec{y}' = 0$ by the way we constructed $\vec{y}'$. We then use the method from part (a) to find $\frac{A^k\vec{y}'}{||A^k\vec{y}'||}$ and by the same argument used in part (a), this will give us $\pm\vec{u}_1$. This is because now, λ_1 is the largest eigenvalue that shows up since $A^k\vec{y}' = \tilde{\beta}_1\lambda_1^k\left(\vec{u}_1 + \sum_{i=2}^{n-1}\frac{\tilde{\beta}_i}{\tilde{\beta}_1}\left(\frac{\lambda_i}{\lambda_1}\right)^k\vec{u}_i\right)$. All the other terms go to zero as k gets large.

(c) (Bonus 10pts) Extend the previous part to create a way to find all of the eigenvectors of A. Show how to do the key inductive step. **Assuming we have found the first i eigenvectors, describe a method we can use to find the next eigenvector $\vec{u}_{i+1}$.**

Solutions: This is done in a very similar way to part (b), we start with a fresh random vector $\vec{x}$, and we can assume that $|\beta_i| > 0$. We know the eigenvectors $\vec{u}_0, \ldots, u_i$, and because A is symmetric, these eigenvectors are orthogonal. So we can subtract the projections of $\vec{x}$ onto these eigenvectors to obtain a new vector $\vec{x}' = \vec{x} - \sum_{k=0}^{i} \vec{u}_k \vec{u}_k^T \vec{x}$.

Consequently, we have $\vec{u}_0^T \vec{x}' = \vec{u}_1^T \vec{x}' = \cdots = \vec{u}_i^T \vec{x}' = 0$ while we know that $|\vec{u}_{i+1}^T \vec{x}'| > 0$.

We then use the method from part (a) to give us $A^k \vec{x}' \approx \beta_{i+1} \lambda_{i+1}^k \vec{u}_{i+1}$. We can then normalize this vector to obtain the unit eigenvector $\vec{u}_{i+1}$.

Although this method for finding eigenvectors past the first dominant eigenvector works out mathematically, it is not exactly how this is implemented in numerical software. The challenge is caused by rounding errors in floating point arithmetic and because $A^k \vec{x}$ approaches $\vec{u}_0$ but is never exactly equal to the eigenvector. When the estimate for $\vec{u}_0$ is projected out of some vector $\vec{x}$, there is still a small component in the direction of the true eigenvector $\vec{u}_0$ because of the approximation. As a result, this eigenvector will still dominate the computation $A^k \vec{x}$.

PRINT your name and student ID: ____

[Doodle page! Draw us something if you want or give us suggestions or complaints. You can also use this page to report anything suspicious that you might have noticed.]