EECS 16B Designing Information Devices and Systems II
Spring 2021 UC Berkeley

Homework 10

This homework is due on Friday, April 2, 2021, at 11:00PM. Self-grades and HW Resubmission are due on Tuesday, April 6, 2021, at 11:00PM.

1. Reading Lecture Notes

Staying up to date with lectures is an important part of the learning process in this course. Here are links to the notes that you need to read for this week: Note 10A

(a) Consider $A \in \mathbb{R}^{n\times n}$ where the columns of A (denoted by $\vec{a_k}, 1 \leq k \leq n$) are orthonormal. What does the least squares solution $(A^TA)^{-1}A^T\vec{y}$ simplify to?

Solution: Since the columns of A are orthonormal, then $(A^TA)^{-1} = I$. If we write $A = \begin{bmatrix} \vec{a_1} & \dots & \vec{a_n} \end{bmatrix}$ then $A^T = \begin{bmatrix} \vec{a_1}^T \\ \vdots \\ \vec{a_n}^T \end{bmatrix}$ and finally $A^T\vec{y} = \begin{bmatrix} \vec{a_1}^T\vec{y} \\ \vdots \\ \vec{a_n}^T\vec{y} \end{bmatrix}$. So, our final expression is $(A^TA)^{-1}A^T\vec{y} = \begin{bmatrix} \vec{a_1}^T\vec{y} \\ \vdots \\ \vec{a_n}^T\vec{y} \end{bmatrix}$.

(b) Suppose we have two vectors $\vec{v_1} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\vec{v_2} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \in \mathbb{R}^2$. Then, we use Gram-Schmidt Orthonormalization to construct $\vec{q_1}$ and $\vec{q_2}$. Are $\vec{q_1}$ and $\vec{q_2}$ the only vectors that form an orthogonal basis for the span$\{\vec{v_1}, \vec{v_2}\}$?

Solution: No, in fact there are infinitely many vectors that make an orthogonal basis for span$\{\vec{v_1}, \vec{v_2}\}$! We can take $\vec{q_1}$ and $\vec{q_2}$ and rotate them by the same angle. That is, define $\vec{u_1}(\theta) = R_\theta\vec{q_1}$ and $\vec{u_2}(\theta) = R_\theta\vec{q_2}$ where R_θ is a matrix that rotates a 2-D vector by an angle θ. So, $\vec{u_1}(\theta)$ and $\vec{u_2}(\theta)$ form an orthongonal basis for the span$\{\vec{v_1}, \vec{v_2}\}$ for all values of θ.

2. Gram-Schmidt Basic

(a) **Use Gram-Schmidt to find a matrix U whose columns form an orthonormal basis for the column space of V.**

$$V = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Solution:

We start with the columns of V as our basis for the column space of V, and we want to find an orthonormal basis for this same space using Gram-Schmidt. For notational convenience, define

$$V = \begin{bmatrix} | & | & | \\ \vec{v}_1 & \vec{v}_2 & \vec{v}_3 \\ | & | & | \end{bmatrix}$$

We summarize the first few steps of the Gram-Schmidt algorithm as follows:

i. $\vec{u}_1' = \vec{v}_1$; $\vec{u}_1 = \frac{\vec{u}_1'}{\|\vec{u}_1'\|}$.

ii. $\vec{u}_2' = \vec{v}_2 - \langle \vec{v}_2, \vec{u}_1 \rangle \vec{u}_1$; $\vec{u}_2 = \frac{\vec{u}_2'}{\|\vec{u}_2'\|}$.

iii. $\vec{u}_3' = \vec{v}_3 - \langle \vec{v}_3, \vec{u}_1 \rangle \vec{u}_1 - \langle \vec{v}_3, \vec{u}_2 \rangle \vec{u}_2$; $\vec{u}_3 = \frac{\vec{u}_3'}{\|\vec{u}_3'\|}$.

For the column space of V, this is

i. $\vec{u}_1' = \vec{v}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T$. Since $\vec{u}_1'$ is already normalized, we simply set $\vec{u}_1 = \vec{u}_1'$.

ii.

$$\vec{u}_2' = \vec{v}_2 - \langle \vec{v}_2, \vec{u}_1 \rangle \vec{u}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad \vec{u}_2 = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{bmatrix}$$

iii.

$$\vec{u}_3' = \vec{v}_3 - \langle \vec{v}_3, \vec{u}_1 \rangle \vec{u}_1 - \langle \vec{v}_3, \vec{u}_2 \rangle \vec{u}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \frac{2}{\sqrt{2}} \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \qquad \vec{u}_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

Thus, the matrix U is given by

$$U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

(b) Show that you get the same resulting vector when you project $\vec{w} = \begin{bmatrix} 1 \\ -1 \\ 0 \\ -1 \\ 0 \end{bmatrix}$ onto the columns of V as you do when you project onto the columns of U, i.e. **show that**

$$V(V^TV)^{-1}V^T\vec{w} = U(U^TU)^{-1}U^T\vec{w}.$$

Feel free to use numpy. No need to grind this out by hand.

Solution:

Note that whatever basis we use for a subspace, when we project a vector onto that subspace, we get the same vector. For example, when we project the vector $\vec{w} = \begin{bmatrix}1 & -1 & 0 & -1 & 0\end{bmatrix}^T$ onto the subspace using the V basis, we get

$$\begin{aligned}
V(V^TV)^{-1}V^T\vec{w} &= \begin{bmatrix}1 & 1 & 1\\0 & 1 & 1\\0 & 1 & 1\\0 & 0 & 1\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 1\\1 & 3 & 3\\1 & 3 & 5\end{bmatrix}^{-1}\begin{bmatrix}1 & 0 & 0 & 0 & 0\\1 & 1 & 1 & 0 & 0\\1 & 1 & 1 & 1 & 1\end{bmatrix}\begin{bmatrix}1\\-1\\0\\-1\\0\end{bmatrix}\\
&= \begin{bmatrix}1 & 1 & 1\\0 & 1 & 1\\0 & 1 & 1\\0 & 0 & 1\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\frac{3}{2} & -\frac{1}{2} & 0\\-\frac{1}{2} & 1 & -\frac{1}{2}\\0 & -\frac{1}{2} & \frac{1}{2}\end{bmatrix}\begin{bmatrix}1\\0\\-1\end{bmatrix}\\
&= \begin{bmatrix}1 & 1 & 1\\0 & 1 & 1\\0 & 1 & 1\\0 & 0 & 1\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\frac{3}{2}\\0\\-\frac{1}{2}\end{bmatrix} = \begin{bmatrix}1\\-\frac{1}{2}\\-\frac{1}{2}\\-\frac{1}{2}\\-\frac{1}{2}\end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
U(U^TU)^{-1}U^T\vec{w} &= \begin{bmatrix}1 & 0 & 0\\0 & \frac{1}{\sqrt{2}} & 0\\0 & \frac{1}{\sqrt{2}} & 0\\0 & 0 & \frac{1}{\sqrt{2}}\\0 & 0 & \frac{1}{\sqrt{2}}\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}^{-1}\begin{bmatrix}1 & 0 & 0 & 0 & 0\\0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0\\0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix}\begin{bmatrix}1\\-1\\0\\-1\\0\end{bmatrix}\\
&= \begin{bmatrix}1 & 0 & 0\\0 & \frac{1}{\sqrt{2}} & 0\\0 & \frac{1}{\sqrt{2}} & 0\\0 & 0 & \frac{1}{\sqrt{2}}\\0 & 0 & \frac{1}{\sqrt{2}}\end{bmatrix}\begin{bmatrix}1\\-\frac{1}{\sqrt{2}}\\-\frac{1}{\sqrt{2}}\end{bmatrix} = \begin{bmatrix}1\\-\frac{1}{2}\\-\frac{1}{2}\\-\frac{1}{2}\\-\frac{1}{2}\end{bmatrix}
\end{aligned}$$

Note however that the projection using the U basis was much simpler. Since U^TU is the identity, we didn't need to do a matrix inversion.

3. Symmetric Matrices

We want to show that every real symmetric matrix can be diagonalized by a matrix of its orthonormal eigenvectors. In other words, a symmetric matrix has an orthonormal eigenbasis. This is called the spectral theorem for real symmetric matrices.

In discussion section, you have seen a recursive derivation of a related fact. Formally however, such recursive derivations are usually turned into proofs by using induction. This problem serves to both freshen your mind regarding induction as well as to give you a chance to prove for yourself this very important theorem. (This is the same essential proof as that of Schur upper-triangularization. So understanding this problem will help solidify your understanding of that proof as well.)

(a) You will start by proving a basic lemma about real symmetric matrices under an orthonormal change of basis. **Prove that if S is an $m \times m$ symmetric matrix ($S = S^\top$) and U is any $m \times n$ matrix, then $U^T SU$ is also symmetric.**

Solution: We must show $(U^T SU)^T = (U^T SU)$ to show it is symmetric. The transpose of $U^T SU$, $(U^T SU)^T$ is identical to $U^T SU$ as follows:

$$(U^T SU)^T = (SU)^T U = U^T S^T U = U^T SU$$

since $S^T = S$.

(b) Another useful lemma is one about finding orthonormal bases. **Show that given a single nonzero vector $\vec{u}_0$ of dimension n, it is possible to find an orthonormal set of n vectors, $\vec{v}_0, \ldots, \vec{v}_{n-1}$ such that $\vec{v}_0 = \alpha \vec{u}_0$ for some scalar α.**

(Hint: Use the Gram-Schmidt process on the list of $n+1$ vectors obtained by starting with the given vector and appending the standard basis — i.e. the columns of the identity matrix. Follow the procedure as in lecture.)

Solution: Finding $\vec{v}_0, \ldots, \vec{v}_{n-1}$ can be done with two steps: (1) based on $\vec{u}_0$, find $\vec{u}_1, \ldots, \vec{u}_N$, such that span$(\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_N) = \mathbb{R}^n$ with $N \geq n$ and (2) apply the Gram-Schmidt process to orthonormalize that set of vectors, dropping any zero vectors that we encounter along the way.

For (1), we can simply append the standard basis to $\vec{v}_0 = \frac{\vec{v}_0}{\|\vec{v}_0\|}$. This gives us $N = n + 1 \geq n$ vectors that certainly span all of $\mathbb{R}^n$ since the standard basis definitely spans all of $\mathbb{R}^n$ even by itself.

For (2), we can simply apply the Gram-Schmidt process to find an orthonormal set of n vectors. The Gram-Schmidt process takes a finite, linearly independent set $U = \{\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_N\}$ and generates an orthonormal set $W = \{\vec{w}_0, \vec{w}_1, \cdots, \vec{w}_{n-1}\}$ that spans the same space as U. If along the way, it finds a zero vector, it just drops that vector.

For this problem, we first compute the set W based on U: let $\vec{w}_0 = \vec{u}_0$, $\vec{w}_1 = \vec{u}_1 - \frac{\langle \vec{w}_0, \vec{u}_1 \rangle}{\|\vec{w}_0\|} \vec{w}_0$, and keep computing other $\vec{w}_k$ in W by subtracting $\vec{u}_k$ with its projections upon the prior existing $\vec{w}_0, \vec{w}_1, \ldots, \vec{w}_{k-1}$. If the result of the subtraction is zero, we can skip it. After that, we normalize each vector in W to make it an orthonormal basis V, where $\vec{v}_0 = \frac{\vec{w}_0}{\|\vec{w}_0\|} = \frac{\vec{u}_0}{\|\vec{u}_0\|} = \alpha \vec{u}_0$.

(c) For the main proof that every real symmetric matrix is diagonalized by a matrix of its orthonormal eigenvectors, we will proceed by formal induction. Recall that for a proof by induction, we have to start with a base case - this is also the base case in a recursive derivation.

Consider the trivial case of S having dimensions $[1 \times 1]$ ($n = 1$). **Does S have an eigenvector? Does S have an eigenbasis? Can this eigenbasis be made orthonormal? Is the matrix diagonal in this basis? Are the entries real?**

Solution: Yes, it has an eigenvector, because S would be a scalar. Let $S = [s]$ and $\vec{u} = 1$. Then note that $S\vec{u} = s\vec{u}$. This implies that s is an eigenvalue and $\vec{u} = 1$ is an eigenvector. This eigenvector

$\vec{u}$ makes up an eigenbasis in $\mathbb{R}^1$. This eigenbasis is trivially orthonormal, because there is no other eigenvector of S which is not orthogonal to $\vec{u}$. We can think of S as a $[1 \times 1]$ matrix, such that the only entry is real. Also, S is diagonal in that basis because there is only one element.

(d) After the base case, we do an inductive stage of the main proof. The first step in the inductive stage is to write down the induction hypothesis. Assume that the property that every real symmetric matrix is diagonalized by a matrix of its orthonormal eigenvectors holds for all symmetric matrices with size $[(n-1) \times (n-1)]$. **Complete the proof template below by filling in the blanks**. *(Hint: In general for proofs by induction, you want to start with the strongest version of what you want to prove. This gives you the most powerful inductive hypothesis.)*

Goal: we want to show that any $[n \times n]$ real symmetric matrix S can be diagonalized by a matrix of its orthonormal real eigenvectors.
Base case: when $n = 1$, this holds for $[1 \times 1]$ matrix $S = [s]$ as you showed in Part (d).
Inductive hypothesis: Let Q be an $[(n-1) \times (n-1)]$ real symmetric matrix with eigenvectors $\vec{u}_0, \ldots, \vec{u}_{n-2}$, then we can express Q as $U \Lambda_Q U^T$, where Λ_Q is a ________ matrix with ________ eigenvalues of Q along its ________, and U is an $[(n-1) \times (n-1)]$ matrix, where the columns are ________ of Q.

Solution:

Goal: we want to show that any $[n \times n]$ real symmetric matrix S can be diagonalized by a matrix of its orthonormal real eigenvectors.
Base case: when $n = 1$, this holds for $[1 \times 1]$ matrix $S = [s]$ as you showed in Part (d).
Inductive hypothesis: Let Q be an $[(n-1) \times (n-1)]$ real symmetric matrix with eigenvectors $\vec{u}_0, \ldots, \vec{u}_{n-2}$, then we can express Q as $U \Lambda_Q U^T$, where Λ_Q is a diagonal matrix with real eigenvalues of Q along its diagonal, and U is an $[(n-1) \times (n-1)]$ matrix, where the columns are orthonormal real eigenvectors of Q.

(e) Now think about a symmetric matrix S with size $[n \times n]$. Consider a real eigenvalue λ_0 of S and the corresponding eigenvector $\vec{u}_0$ (a column vector with size n). **Use an appropriate orthonormal change of basis V to show that $S = VXV^T$, where X is of the form**

$$X = \begin{bmatrix} \lambda_0 & x_{1,2} & \cdots & x_{1,n} \\ 0 & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & x_{n,2} & \cdots & x_{n,n} \end{bmatrix}.$$

That is, the first column of X is $\begin{bmatrix} \lambda_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$.

(Hint: Follow the proof strategy from lecture. What should the first column of V be? How can you fill out the rest?)

Solution: According to (b), using the Gram-Schmidt process to construct a basis starting with $\vec{u}_0$, we could derive an orthonormal set of n vectors, $\vec{v}_0, \ldots, \vec{v}_{n-1}$. Choose $\vec{v}_0 = \frac{\vec{u}_0}{\|\vec{u}_0\|}$ Let $\vec{v}_0, \ldots, \vec{v}_{n-1}$ be the columns of matrix V. Suppose $X = V^T S V$, such that $S = VXV^T$ (because V is an orthonormal basis, $V^T V = V V^T = I$).

Hence we have

$$X = \begin{bmatrix} \vec{v}_0^T \\ \vec{v}_1^T \\ \vdots \\ \vec{v}_{n-1}^T \end{bmatrix} S \begin{bmatrix} \vec{v}_0 & \vec{v}_1 & \cdots & \vec{v}_{n-1} \end{bmatrix},$$

where $\vec{v}_0$ is the normalized eigenvector corresponding to the eigenvalue λ_0 of S, which means $S\vec{v}_0 = \lambda_0\vec{v}_0$. The first column of X will be

$$\begin{bmatrix} \vec{v}_0^T \\ \vec{v}_1^T \\ \vdots \\ \vec{v}_{n-1}^T \end{bmatrix} S\vec{v}_0 = \begin{bmatrix} \vec{v}_0^T \\ \vec{v}_1^T \\ \vdots \\ \vec{v}_{n-1}^T \end{bmatrix} \lambda_0\vec{v}_0.$$

Since V is an orthonomal matrix, $\vec{v}_i^T\vec{v}_j = 1$ when $i = j$; otherwise it is zero. Therefore, the first column of X is

$$\begin{bmatrix} \lambda_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

which is what we wanted to show.

(f) Continue the previous part **to show that in fact S can be written as**

$$S = V \begin{bmatrix} \lambda_0 & \vec{0}^T \\ \vec{0} & Q \end{bmatrix} V^T$$

where Q is an $[(n-1)\times(n-1)]$ symmetric matrix and V is the matrix that you found above. *Hint: Define $\vec{v}_0 = \frac{\vec{u}_0}{\|\vec{u}_0\|}$, think of V as $V = \begin{bmatrix} \vec{v}_0 & R \end{bmatrix}$, where the dimension of R is $[n\times(n-1)]$. Also recall that S is a symmetric matrix.*

Solution: Recall the lemma we proved in (a), X must be a symmetric matrix, because $X = V^T SV$ where S is a symmetric matrix and V is orthonormal. Since we have proved the first column of this matrix is

$$\begin{bmatrix} \lambda_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

the transpose of this is the first row. Hence X can be written as

$$\begin{bmatrix} \lambda_0 & \vec{0}^T \\ \vec{0} & Q \end{bmatrix},$$

where Q is an $n-1$ by $n-1$ symmetric matrix. This is a short direct argument that gets us what we want.

An alternative approach was shown in discussion where we computed X directly. Let R be a matrix with columns $\vec{v}_1, \ldots, \vec{v}_{n-1}$, which are orthogonal unit vectors. Recall that λ_0 is an eigenvalue of S, with the corresponding eigenvector $\vec{v}_0$. From here, we can compute X step by step :

$$X = V^T SV = \begin{bmatrix} \vec{v}_0 & R \end{bmatrix}^T S \begin{bmatrix} \vec{v}_0 & R \end{bmatrix} = \begin{bmatrix} \vec{v}_0^T \\ R^T \end{bmatrix} S \begin{bmatrix} \vec{v}_0 & R \end{bmatrix} = \begin{bmatrix} \vec{v}_0^T \\ R^T \end{bmatrix} \begin{bmatrix} S\vec{v}_0 & SR \end{bmatrix} = \begin{bmatrix} \vec{v}_0^T \\ R^T \end{bmatrix} \begin{bmatrix} \lambda_0\vec{v}_0 & SR \end{bmatrix}$$

Now we could write X as

$$\begin{bmatrix} \lambda_0\vec{v}_0^T\vec{v}_0 & \vec{v}_0^T SR \\ \lambda_0 R^T\vec{v}_0 & R^T SR \end{bmatrix},$$

Because $\vec{v}_0$ is a unit vector, $\vec{v}_0^T\vec{v}_0 = 1$. Since V is an orthonormal matrix, the inner product of $\vec{v}_0$ with column vectors in R must be 0, so $R^T\vec{v}_0 = \vec{0}$. Also, $\vec{v}_0^T SR = (S\vec{v}_0)^T R = \lambda_0\vec{v}_0^T R = \vec{0}^T$.

Then we get the desired form of X, where $Q = R^T SR$. Recall the lemma we proved in (a), because S is a symmetric matrix, while the columns of R are orthonormal, Q must be symmetric.

Either one of these approaches is fine.

(g) According to our induction hypothesis, we can write Q as $U\Lambda U^T$ where U is an orthonormal $[(n-1) \times (n-1)]$ square matrix and Λ is a diagonal matrix with real entries along the diagonal and 0s everywhere else. **Substitute this in the previous part to show that indeed there must exist an orthonormal $[n \times n]$ square matrix W such that**

$$S = W \begin{bmatrix} \lambda_0 & \vec{0}^T \\ \vec{0} & \Lambda \end{bmatrix} W^T$$

(Hint: If R and U are both orthonormal matrices, is RU also orthonormal? What is $(RU)^T(RU)$?)

Solution:

From the previous part, we know that, S can be written as follows.

$$S = \begin{bmatrix} \vec{v}_0 & R \end{bmatrix} \begin{bmatrix} \lambda_0 & \vec{0}^T \\ \vec{0} & Q \end{bmatrix} \begin{bmatrix} \vec{v}_0^T \\ R^T \end{bmatrix}$$

Thanks to our induction hypothesis, we can compute S as follows:

$$S = \begin{bmatrix} \vec{v}_0 & R \end{bmatrix} \begin{bmatrix} \lambda_0 & \vec{0}^T \\ \vec{0} & U\Lambda U^T \end{bmatrix} \begin{bmatrix} \vec{v}_0^T \\ R^T \end{bmatrix}$$

At this point, what we would like to do is to pull the Us out of the inner matrix and stick them with the Rs in the outer matrices. Then we would be done. This could be done immediately because of the properties of block matrix multiplication, or we could further justify it.

If you wanted to further justify it (this was not required for full credit), we could calculate as follows:

$$\begin{bmatrix} \vec{v}_0 & R \end{bmatrix} \begin{bmatrix} \lambda_0 & \vec{0}^T \\ \vec{0} & U\Lambda U^T \end{bmatrix} \begin{bmatrix} \vec{v}_0^T \\ R^T \end{bmatrix} = \begin{bmatrix} \lambda_0\vec{v}_0 & RU\Lambda U^T \end{bmatrix} \begin{bmatrix} \vec{v}_0^T \\ R^T \end{bmatrix} = (\begin{bmatrix} \lambda_0\vec{v}_0 & O_{n\times(n-1)} \end{bmatrix} + \begin{bmatrix} \vec{0} & RU\Lambda U^T \end{bmatrix}) \begin{bmatrix} \vec{v}_0^T \\ R^T \end{bmatrix},$$

where $O_{n\times(n-1)}$ is a matrix filled with zeros.

Then

$$S = \begin{bmatrix} \lambda_0\vec{v}_0\vec{v}_0^T \end{bmatrix} + \begin{bmatrix} RU\Lambda U^T R^T \end{bmatrix}.$$

Then we can place S into the orthonormal basis $W = \begin{bmatrix} \vec{v}_0 & RU \end{bmatrix}$ to verify:

$$S = \begin{bmatrix} \vec{v}_0 & RU \end{bmatrix} \begin{bmatrix} \lambda_0 & \vec{0}^T \\ \vec{0} & \Lambda \end{bmatrix} \begin{bmatrix} \vec{v}_0^T \\ U^T R^T \end{bmatrix} = \begin{bmatrix} \lambda_0 \vec{v}_0 & RU\Lambda \end{bmatrix} \begin{bmatrix} \vec{v}_0^T \\ U^T R^T \end{bmatrix} = (\begin{bmatrix} \lambda_0 \vec{v}_0 & O_{n\times(n-1)} \end{bmatrix} + \begin{bmatrix} \vec{0} & RU\Lambda \end{bmatrix}) \begin{bmatrix} \vec{v}_0^T \\ U^T R^T \end{bmatrix},$$

which also becomes

$$\begin{bmatrix} \lambda_0 \vec{v}_0 \vec{v}_0^T \end{bmatrix} + \begin{bmatrix} RU\Lambda U^T R^T \end{bmatrix}.$$

Regardless of whether we felt required to justify pulling the Us out, we know that the multiplication of two orthonomal matrices (in this case, R and U) is an orthonormal matrix as well. Why? Because $(RU)^T(RU) = U^T R^T RU = U^T IU = U^T U = I$.

It is easy to further verify that $\vec{v}_0$ is orthonormal to all columns of RU by seeing what happens to $\vec{v}_0^T RU$: $\vec{v}_0^T RU = (\vec{v}_0^T R)U = \vec{0}^T U = \vec{0}^T$, because $\vec{v}_0$ is orthogonal to column vectors of R. Therefore, we can now define the orthonormal basis W as,

$$W = \begin{bmatrix} \vec{v}_0 & RU \end{bmatrix}$$

This shows what we wanted to prove:

$$S = W \begin{bmatrix} \lambda_0 & \vec{0}^T \\ \vec{0} & \Lambda \end{bmatrix} W^T$$

and we are done.

(h) **Finally, prove that the eigenvalues λ of real, symmetric matrix S are real.** *Hint: Suppose that S had a complex eigenvalue λ with eigenvector $\vec{v}$. Because S is a real matrix, what do you know about $S\overline{\vec{v}}$, where $S\overline{\vec{v}}$ is the complex conjugate of $\vec{v}$? What happens when you take a potentially complex number and multiply it by its own complex conjugate? Consequently, what do you know happens if you multiply a complex vector $\vec{v}$ by the conjugate of its transpose: i.e. consider $\overline{\vec{v}}^T \vec{v}$? Since S is symmetric, what do you know about $\overline{\vec{v}}^T S$?*

Solution: Let λ be a possibly complex eigenvalue of a real symmetric matrix A. Thus, there is a nonzero vector $\vec{v}$ such that $A\vec{v} = \lambda\vec{v}$.

By taking the complex conjugates of both sides, and noting that $\overline{A} = A$ since A has real entries, we get

$\overline{A\vec{v}} = \overline{\lambda\vec{v}} \implies A\overline{\vec{v}} = \overline{\lambda\vec{v}}$

Now, using $A^T = A$

$$\overline{\vec{v}}^T A\vec{v} = \overline{\vec{v}}^T (A\vec{v}) = \overline{\vec{v}}^T (\lambda\vec{v}) = \lambda(\overline{\vec{v}}^T \vec{v})$$

$$\overline{\vec{v}}^T A\vec{v} = \left(A\overline{\vec{v}}\right)^T \vec{v} = \left(\overline{\lambda\vec{v}}\right)^T \vec{v} = \overline{\lambda}(\overline{\vec{v}}^T \vec{v})$$

Since $v \neq 0$, we have that $\overline{v}^T v > 0$. Thus, $\lambda = \overline{\lambda}$ and consequently $\lambda \in \mathbb{R}$.

By induction, we are now done since we have proved that having the desired property for $n-1$ implies that we have the property for n and we also have a valid base case at $n = 1$.

According to the base case and inductive steps we just proved, the statement, “every real symmetric matrix is diagonalized by a matrix of its real orthonormal eigenvectors” is proved by induction.

Solution: The proof here also has this nice recursive character to it that goes along well with what you have learned in our sister courses 61ab.

You will get more practice doing inductive proofs in our successor course 70.

4. Classification of Sinusoids

This HW problem can be viewed as a warm-up for the next topic in the course: which is going to be motivated by figuring out how to process signals recorded from the brain to decipher what a person wants to do in terms of a specific command to their robot arm. These kinds of problems are called "classification" problems. In this exercise, you will be using jupyter to classify sinusoids.

The iPython notebook Sinusoidal_Projection_fa19_prob.ipynb will guide you through the process of performing sinusoidal projections.

Suppose you already know the true potential frequencies f_i and potential phases ϕ_i of a set of sinusoidal signals

$$S := \left\{ \sin(2\pi f_i k + \phi_i), i = 1, 2, \ldots, n \right\}, \tag{1}$$

and you have some noisy samples of these true sinusoidal signals. You want to determine the true sinusoidal signal for each of these noisy samples—How would you approach the problem?

We will show in this problem that we can project noisy sinusoidal signals onto noiseless sinusoids to achieve good classification.

In the realistic world, one often doesn't have the complete waveform of a continuous function, instead oftentimes one works with *samples* of the continous function.

In our case, we generate noisy samples of the true sinusoidal signals in the following way. For each of the $num\_sinusoids$ true frequencies, each noisy sample y_i consists of N sample points sampled with a sampling rate of F_s sample rate, and corrupted by noise scaled by σ.

$$y_i(k) = \sin\left(2\pi f \cdot k / F_s\right) + \sigma \cdot Noise. \quad k = 1, 2, \ldots, N.$$

In our example, we will work with $num\_sinusoids = 3$.

A higher σ corresponds to more noise in our measurements.

Please complete the notebook by following the instructions given.

(a) Run the first part of the jupyter notebook to generate our noisy data points. **Use** $\sigma = 0.1, 1.0, 10.0, 100.0$ **and comment on what you observe in the plots.**

Solution: The solutions notebook includes the plots. As you increase the scaling of the noise, you should observe that the averages and observations become noisier. When the noise is very small, we don't need to do much to classify the noisy data.

(b) **Complete part (b) of the notebook** to project noisy sinusoids onto potential true sinusoids.
Sketch the resulting 3D plot of projections qualitatively. Comment on what happens when you try the noise scalings $\sigma = 0.1, 1.0, 10.0, 100.0$**.**

Solution: The solutions notebook includes the plots. As you increase the scaling of the noise, you should observe that the clustering becomes more undistinguishable.

(c) **Complete part (c) of the notebook** to classify the data points and calculate the number of misclassified points.
Report the number of misclassifications for $\sigma = 0.1, 1.0, 10.0, 100.0$**. Explain what happens when there is a high level of noise.** Recall that our noisy process is random so that there can be cases where there are misclassifications even in low noise.

Solution: At values less than 2.0, the number of misclassified points should be close to 0. At 5.0, the number of misclassified points should be very high. This is natural because our sinusoids only have magnitude 1. With very high noise, it becomes harder to distinguish true signal from noise.

(d) **For what qualitative regions of the noise level is it very beneficial for us to use projections?** For very low values of noise, do you have to do projections to successfully classify? What else could you have done? This question is asking you to reflect on what you have observed.

Solution: When noise makes the signals indistinguishable by eye, it is helpful for us to use projections.

5. Using upper-triangularization to solve differential equations

You know that for any square matrix A with real eigenvalues, there exists a real matrix V with orthonormal columns and a real upper triangular matrix R so that $A = VRV^T$. In particular, to set notation explicitly:

$$V = \begin{bmatrix} \vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n \end{bmatrix}$$
$$R = \begin{bmatrix} \vec{r}_1^T \\ \vec{r}_2^T \\ \vdots \\ \vec{r}_n^T \end{bmatrix}$$

where the rows of the upper-triangular R look like

$$\begin{aligned} \vec{r}_1^T &= [\lambda_1, r_{1,2}, r_{1,3}, \ldots, r_{1,n}] \\ \vec{r}_2^T &= [0, \lambda_2, r_{2,3}, r_{2,4}, \ldots, r_{2,n}] \\ \vec{r}_i^T &= [\underbrace{0, \ldots, 0}_{i-1 \text{ times}}, \lambda_i, r_{i,i+1}, r_{i,i+2}, \ldots, r_{i,n}] \\ \vec{r}_n^T &= [\underbrace{0, \ldots, 0}_{n-1 \text{ times}}, \lambda_n] \end{aligned}$$

where the λ_i are the eigenvalues of A.

Suppose our goal is to solve the n-dimensional system of differential equations written out in vector/matrix form as:

$$\begin{aligned} \frac{d}{dt}\vec{x}(t) &= A\vec{x}(t) + \vec{u}(t), \\ \vec{x}(0) &= \vec{x}_0, \end{aligned}$$

where $\vec{x}_0$ is a specified initial condition and $\vec{u}(t)$ is a given vector of functions of time.

Assume that the V and R have already been computed and are accessible to you using the notation above.

Assume that you have access to a function $ScalarSolve(\lambda, y_0, \check{u})$ that takes a real number λ, a real number y_0, and a real-valued function of time $\check{u}$ as inputs and returns a real-valued function of time that is the solution to the scalar differential equation

$$\frac{d}{dt}y(t) = \lambda y(t) + \check{u}(t) \tag{2}$$

with initial condition $y(0) = y_0$.

Also assume that you can do regular arithmetic using real-valued functions and it will do the right thing. So if u is a real-valued function of time, and g is also a real-valued funcion of time, then $5u + 6g$ will be a real valued function of time that evaluates to $5u(t) + 6g(t)$ at time t.

Use V, R to construct a procedure for solving this differential equation

$$\begin{aligned} \frac{d}{dt}\vec{x}(t) &= A\vec{x}(t) + \vec{u}(t), \\ \vec{x}(0) &= \vec{x}_0, \end{aligned}$$

for $\vec{x}(t)$ by filling in the following template in the spots marked $\clubsuit, \diamondsuit, \heartsuit, \spadesuit$.

(Note: It will be useful to upper triangularize A by change of basis to get a differential equation in terms of R instead of A.)
(Note: We use the notation $\vec{v}[i]$ to be the ith component of the vector $\vec{v}$)
(HINT: The process here should be simillar to diagonalization with some modifications. Start from the last row of the system and work your way up to understand the algorithm.)

1: $\widetilde{\vec{x}}_0 = V^T \vec{x}_0$ ▷ Change the initial condition to be in V-coordinates
2: $\widetilde{\vec{u}} = V^T \vec{u}$ ▷ Change the external input functions to be in V-coordinates
3: **for** $i = n$ down to 1 **do** ▷ Iterate up from the bottom row
4: $\quad \breve{u}_i = \clubsuit + \sum_{j=i+1}^{n} \spadesuit$ ▷ Make the effective input for this level
5: $\quad \widetilde{x}_i = ScalarSolve(\diamondsuit, \widetilde{x}_0[i], \breve{u}_i)$ ▷ Solve this level's scalar differential equation
6: **end for**
7: $\vec{x}(t) = \heartsuit \begin{bmatrix} \widetilde{x}_1 \\ \widetilde{x}_2 \\ \vdots \\ \widetilde{x}_n \end{bmatrix} (t)$ ▷ Change back into original coordinates

(a) Give the expression for $\heartsuit$ on line 7 of the algorithm above. (i.e. How do you get from $\widetilde{\vec{x}}(t)$ to $\vec{x}(t)$?)

Solution: Since $\widetilde{\vec{x}}_0 = V^T \vec{x}_0$ we know we are changing to V-basis. So, the implicit change of variable that we are doing is $\widetilde{\vec{x}} = V^T \vec{x}$, this means that to come back, $\vec{x} = V\widetilde{\vec{x}}$.
Thus, $\heartsuit = V$.

(b) Give the expression for $\diamondsuit$ on line 5 of the algorithm above. (i.e. What are the λ arguments to $ScalarSolve$, equation (2), for the i^{th} iteration of the for-loop?)
(HINT: Convert the differential equation to be in terms of R instead of A. It may be helpful to start with i = n and develop a general form for the ith row.)

Solution: We begin by taking our vector differential equation and substituting in our upper triangularization:

$$\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + \vec{u}(t)$$
$$\frac{d}{dt}\vec{x}(t) = VRV^T\vec{x}(t) + \vec{u}(t)$$

Multiplying both sides by V^T and using the fact that $V^T V = I$

$$V^T \frac{d}{dt}\vec{x}(t) = RV^T\vec{x}(t) + V^T\vec{u}(t)$$

Now, we perform change of variables, $\widetilde{\vec{x}} = V^T \vec{x}$ and $\widetilde{\vec{u}} = V^T \vec{u}$ so we get,

$$\frac{d}{dt}\widetilde{\vec{x}}(t) = R\widetilde{\vec{x}}(t) + \widetilde{\vec{u}}(t)$$

Thus, the i^{th} equation in this system is,

$$\frac{d}{dt}\widetilde{x}_i(t) = r_i^T \widetilde{\vec{x}}(t) + \widetilde{u}_i(t)$$

Using, $\vec{r}_i^T = [\underbrace{0, \ldots, 0}_{i-1 \text{ times}}, \lambda_i, r_{i,i+1}, r_{i,i+2}, \ldots, r_{i,n}]$ we get,

$$\frac{d}{dt}\widetilde{x}_i(t) = \lambda_i \widetilde{x}_i(t) + r_{i,i+1}\widetilde{x}_{i+1}(t) + r_{i,i+2}\widetilde{x}_{i+2}(t) + \cdots + r_{i,n}\widetilde{x}_n(t) + \widetilde{u}_i(t)$$

Thus, $\frac{d}{dt}\widetilde{x}_i(t) = \lambda_i \widetilde{x}_i(t) + \widetilde{u}_i(t) + \sum_{j=i+1}^{n} r_{i,j}\widetilde{x}_j(t)$
Here, we can see that when solving the scalar differntial equation for the ith row, the scaling term is λ_i: $\diamondsuit = \lambda_i$.

(c) Give the expression for ♣ on line 4 of the algorithm above.

Solution:
Since, from above, $\frac{d}{dt}\widetilde{x}_i(t) = \lambda_i \widetilde{x}_i(t) + \widetilde{u}_i(t) + \sum_{j=i+1}^{n} r_{i,j}\widetilde{x}_j(t)$ we can see that the $\widetilde{u}_i$ is the input term that does not depend on the inner sum. From this we conclude that
$\clubsuit = \widetilde{u}_i$.

(d) Give the expression for ♠ on line 4 of the algorithm above.

Solution:
Since, from above, $\frac{d}{dt}\widetilde{x}_i(t) = \lambda_i \widetilde{x}_i(t) + \widetilde{u}_i(t) + \sum_{j=i+1}^{n} r_{i,j}\widetilde{x}_j(t)$ and so we know what is inside the inner sum:
$\spadesuit = r_{i,j}\widetilde{x}_j$.
Congratulations! You now know how to systematically solve any system of differential equations with constant coefficients, as long as you know how to solve the scalar case with inputs. The same argument style applies for recurrence relations. The only gap that remains is the assumption that all the eigenvalues are real, but now that you understand orthogonality for complex vectors, you can also update your understanding of upper-triangularization to allow for complex matrices as well.

(e) (**OPTIONAL**) Let us complete the algorithm by investigating how $ScalarSolve(\lambda, y_0, \breve{u})$ works. Consider an input that is a weighted polynomial times and exponential.

$$\breve{u}(t) = \alpha t^{\beta} e^{\gamma t}$$

Here, α is a real constant, β is a non-negative integer, and γ is a real exponent. In addition, we will assume that $\gamma = \lambda$ for simplicity. We encourage you to attempt solving this system if $\gamma \neq \lambda$ if you are curious.

What function should $ScalarSolve(\lambda, y_0, \breve{u})$ return for the above $\breve{u}$? Remember, we are only considering the case where $\gamma = \lambda$. Express the answer in terms of α, β, γ.

Solution: $ScalarSolve(\lambda, y_0, \breve{u})$ should return the solution to the differential equation

$$\frac{d}{dt}y(t) = \lambda y(t) + \breve{u}(t) \tag{3}$$

with initial condition $y(0) = y_0$.
Recall from HW 2 the following general integral solution to such a differential equation

$$y(t) = y_0 e^{\lambda t} + \int_0^t \breve{u}(\tau) e^{\lambda(t-\tau)} d\tau \tag{4}$$

Plugging $\breve{u}(t)$ into (4) yields

$$y(t) = y_0 e^{\lambda t} + \int_0^t \alpha \tau^{\beta} e^{\gamma \tau} e^{\lambda(t-\tau)} d\tau \tag{5}$$

$$= y_0 e^{\lambda t} + \alpha e^{\lambda t} \int_0^t \tau^{\beta} e^{\tau(\gamma - \lambda)} d\tau \tag{6}$$

There are now two cases: if $\gamma = \lambda$ and if $\gamma \neq \lambda$. Again, the problem only requires us to solve for the case where $\gamma = \lambda$, but we will show both.

Case 1: $\gamma = \lambda$

In this case, we can simplify (6) to

$$y(t) = y_0 e^{\lambda t} + \alpha e^{\lambda t} \int_0^t \tau^\beta d\tau \tag{7}$$

$$= y_0 e^{\lambda t} + \alpha e^{\lambda t} \left[\frac{1}{\beta+1} \tau^{\beta+1} \right] |_{\tau=0}^{\tau=t} \tag{8}$$

$$= y_0 e^{\lambda t} + \frac{\alpha}{\beta+1} t^{\beta+1} e^{\lambda t} \tag{9}$$

Notice that in this case, we increment the power of t — that is what happens when we enounter this exact λ that matches the input exponential. **This is the solution to the question. Everything below is for your own curiousity**

Case 2: $\gamma \neq \lambda$

In this case, we must use integration by parts to solve for the integral in (6). It would have been fine if you had just looked up the formula. But for completeness, we show the procedure from integral calculus.

Using the tabular integration method, one can solve for $\int F(t)G(t)dt$ by creating a table where the function $F(t)$ is successively differentiated on the left column, and the function $G(t)$ is successively integrated in the right column. Every other entry is negated in the first column, and finally, take the sum of the entries from the first column times the entries in the second column that are one row below.

Considering the integral from (6): $\mathcal{I} = \int_0^t \tau^\beta e^{\tau(\gamma-\lambda)} d\tau$

F	G
$(+)\,\tau^\beta$	$e^{\tau(\gamma-\lambda)}$
$(-)\,\beta\tau^{\beta-1}$	$\frac{1}{(\gamma-\lambda)^1} e^{\tau(\gamma-\lambda)}$
$(+)\,\beta(\beta-1)\tau^{\beta-2}$	$\frac{1}{(\gamma-\lambda)^2} e^{\tau(\gamma-\lambda)}$
$\vdots$	$\vdots$
$(-1)^{\beta-1}\,\frac{\beta!}{1!}\tau^1$	$\frac{1}{(\gamma-\lambda)^{\beta-1}} e^{\tau(\gamma-\lambda)}$
$(-1)^\beta\,\beta!$	$\frac{1}{(\gamma-\lambda)^\beta} e^{\tau(\gamma-\lambda)}$
0	$\frac{1}{(\gamma-\lambda)^{\beta+1}} e^{\tau(\gamma-\lambda)}$

Writing the summation yields

$$\mathcal{I} = \sum_{m=0}^{\beta} (-1)^m \frac{\beta!}{(\beta-m)!} \frac{1}{(\gamma-\lambda)^{(1+m)}} \tau^{\beta-m} e^{\tau(\gamma-\lambda)} \tag{10}$$

We must evaluate $\mathcal{I}$ at the integration bounds $\tau = 0$ and $\tau = t$, which yields

$$\mathcal{I}_{def} = \sum_{m=0}^{\beta} (-1)^m \frac{\beta!}{(\beta-m)!} \frac{1}{(\gamma-\lambda)^{(1+m)}} t^{\beta-m} e^{t(\gamma-\lambda)} \tag{11}$$

Notice that this case doesn't spawn any power of t that is higher than the original β that we started with, although lower powers of t can be spawned in this process. The only way to get higher powers is to encounter the exact same $\lambda = \gamma$.

We use the integral we found $\mathcal{I}_{def}$ to find the solution for (6) as:

$$y(t) = y_0 e^{\lambda t} + \alpha e^{\lambda t}\left(\sum_{m=0}^{\beta}(-1)^m \frac{\beta!}{(\beta-m)!}\frac{1}{(\gamma-\lambda)^{(1+m)}}t^{\beta-m}e^{t(\gamma-\lambda)}\right) \tag{12}$$

$$= y_0 e^{\lambda t} + \alpha\left(\sum_{m=0}^{\beta}(-1)^m \frac{\beta!}{(\beta-m)!}\frac{1}{(\gamma-\lambda)^{(1+m)}}t^{\beta-m}e^{\gamma t}\right) \tag{13}$$

Conclusion of the entire problem: The approach here is completely algorithmic and leans on the linear-algebra of upper-triangularization. In later courses (like 120), you will learn other techniques to get the same solutions that rely on complex analysis based approaches called Laplace Transforms. The overall work is the same in both cases. The advantage to the given approach is just that the proof/derivation is entirely elementary. Meanwhile, the Laplace Transform approach needs to rely on the uniqueness of Laplace Transforms which requires the techniques of Math 185 and beyond to establish.

6. Homework Process and Study Group

Citing sources and collaborators are an important part of life, including being a student!
We also want to understand what resources you find helpful and how much time homework is taking, so we can change things in the future if possible.

(a) **What sources (if any) did you use as you worked through the homework?**

(b) **If you worked with someone on this homework, who did you work with?**
List names and student ID's. (In case of homework party, you can also just describe the group.)

Contributors:

- Siddharth Iyer.
- Yu-Yun Dai.
- Sanjit Batra.
- Anant Sahai.
- Sidney Buchbinder.
- Gaoyue Zhou.
- Kuan-Yun Lee.
- Daniel Abraham.